Documentation 8.0

ZABBIX

02.04.2026

Contents

Руководство пользователя **6**

Добро пожаловать в документацию Zabbix **6**

Начало работы с Zabbix . . . 6
Zabbix Cloud . . . 6
Центр разработчика . . . 6
Сообщество и другие ресурсы . . . 6
1 Установка и первые шаги . . . 6
1 Установка веб-интерфейса . . . 7
2 Краткие руководства по началу работы . . . 13
3 Установка: дополнительные руководства . . . 33
4 Обновление . . . 74
5 Примечания по обновлению для Zabbix 8.0 . . . 89
Известные проблемы . . . 90
Требования . . . 99
2 Что нового в Zabbix 8.0 . . . 108
Шаблоны . . . 109
Элементы данных . . . 110
Плагины . . . 110
Обнаружение низкого уровня . . . 111
Процессы . . . 111
Аутентификация . . . 112
Виджеты . . . 112
Веб-интерфейс . . . 113
Документация . . . 114
Что нового в Zabbix 8.0.x . . . 114
3 Zabbix appliance . . . 114
4 Процессы Zabbix . . . 118
1. Сервер . . . 118
2. Агент . . . 127
3. Агент 2 . . . 131
4. Прокси . . . 140
5. Java gateway . . . 145
6. Sender . . . 149
7. Get . . . 150
8. JS . . . 151
9. Веб-сервис . . . 152
5 Конфигурация . . . 152
1. Узлы сети и группы узлов сети . . . 162
2. Элементы данных . . . 181
3 Обнаружение проблем с помощью триггеров . . . 404
4 События . . . 455
5 Корреляция событий . . . 459
6 Тегирование . . . 466
7 Визуализация . . . 469
8 Шаблоны и группы шаблонов . . . 498
9 Готовые шаблоны . . . 498
10 Оповещения на события . . . 512
11 Макросы . . . 564
12 Пользователи и группы пользователей . . . 578
13 Хранение секретов . . . 588

14 Регулярные отчеты . . . 596
6 Мониторинг сервисов . . . 605
1 Дерево сервисов . . . 606
2 SLA . . . 610
4 Пример настройки . . . 611
7 Веб-мониторинг . . . 616
1 Элементы данных веб-мониторинга . . . 625
2 Сценарий из реальной жизни . . . 627
8 Мониторинг виртуальных машин . . . 634
1 Ключи элементов данных для мониторинга VMware . . . 636
2 Ключевые поля обнаружения виртуальных машин . . . 655
3 Примеры JSON для элементов данных VMware . . . 659
4 Пример настройки мониторинга VMware . . . 664
9 Обслуживание . . . 669
10 Регулярные выражения . . . 674
11 Подтверждение проблемы . . . 679
1 Подавление проблем . . . 682
12 Экспорт/импорт конфигурации . . . 683
1 Группы шаблонов . . . 685
2 Группы узлов сети . . . 685
3 Шаблоны . . . 685
4 Узлы сети . . . 702
5 Карты сети . . . 719
6 Способы оповещения . . . 727
13 Обнаружение . . . 734
1 Обнаружение сети . . . 734
2 Авторегистрация активных агентов . . . 743
3 Низкоуровневое обнаружение (LLD) . . . 747
14 Распределенный мониторинг . . . 804
1 Прокси . . . 804
15 Шифрование . . . 813
1 Использование сертификатов . . . 821
2 Использование общих ключей (pre-shared keys) . . . 827
3 Устранение неполадок . . . 830
16 Веб-интерфейс . . . 833
1 Меню . . . 833
2 Разделы веб-интерфейса . . . 842
3 Настройки пользователя . . . 1107
4 Глобальный поиск . . . 1112
5 Режим обслуживания веб-интерфейса . . . 1114
6 Параметры страницы . . . 1115
7 Определения . . . 1118
8 Создание своей темы . . . 1118
9 Режим отладки . . . 1119
10 Файлы cookie используемые Zabbix . . . 1120
11 Часовые пояса . . . 1121
12 Сброс пароля . . . 1122
13 Селекторы периода времени и узла сети . . . 1122
17 Лучшие практики . . . 1124
1 Рекомендации по обеспечению безопасности . . . 1124
2 Лучшие практики настройки . . . 1133
18 API . . . 1134
Справочник методов . . . 1139
Приложение 1. Справочные комментарии . . . 1976
Приложение 2. Изменения с 7.4 до 8.0 . . . 1982
Приложение 3. Изменения в 8.0 . . . 1984
19 Расширения . . . 1984
1 Загружаемые модули . . . 1986
3 Модули веб-интерфейса . . . 1995
20 Приложения . . . 1996
1 Установка и настройка . . . 1997
2 Настройка процесса . . . 2057
3 Протоколы . . . 2147

4 Элементы данных . . . 2173
5 Поддерживаемые функции . . . 2211
6 Макросы . . . 2211
8 Символы единиц измерения . . . 2247
9 Выполнение команд . . . 2249
9 Настройка периодов времени . . . 2250
10 Совместимость версий . . . 2251
12 Динамически подключаемая библиотека Zabbix sender для Windows . . . 2252
13 Обновление мониторинга сервисов . . . 2252
14 Другие проблемы . . . 2253
16 Примеры экранирования . . . 2254
21 Краткие справочные руководства . . . 2255
Обзор . . . 2255
1 Мониторинг Linux с помощью Zabbix агента . . . 2256
2 Monitor Windows with Zabbix agent . . . 2259
3 Мониторинг Apache по HTTP . . . 2263
4 Мониторинг MySQL с помощью Zabbix агент 2 . . . 2268
5 Мониторинг VMware с помощью Zabbix . . . 2275
6 Мониторинг сетевого трафика с помощью Zabbix . . . 2279
7 Мониторинг сетевого трафика с использованием активных проверок . . . 2282
8 Мониторинг веб-сайтов с помощью элементов данных браузера . . . 2285
9 Мониторинг сертификатов веб-сайтов с помощью Zabbix агент 2 (пассивный) . . . 2290
10 Мониторинг сетевого коммутатора или маршрутизатора с помощью Zabbix . . . 2294
11 Мониторинг журнала событий Windows с использованием активных проверок . . . 2302
Что такое Zabbix . . . 2305

Zabbix Cloud **2310**

Zabbix Cloud **2310**
Настройка и управление Zabbix Cloud . . . 2310
Узнайте о Zabbix Cloud . . . 2310
Развертывание Zabbix в облаке . . . 2310
Конфигурация узла . . . 2314
Обзор . . . 2314
Вкладка «Обзор» . . . 2315
Вкладка «Фильтры доступа» . . . 2316
Вкладка Encryption . . . 2317
Вкладка «Резервные копии» . . . 2318
Вкладка History . . . 2318
Вкладка «Обслуживание» . . . 2319
Вкладка Upgrade . . . 2320
Добавление пользователей . . . 2322
Введение . . . 2322
Добавление пользователей в организации . . . 2322
Добавление пользователей к узлам сети . . . 2323
Ключевые отличия Zabbix Cloud . . . 2324

Центр разработчика **2326**

Центр разработчика **2326**
Создание на основе Zabbix . . . 2326
Плагины . . . 2326
Интерфейсы плагинов . . . 2332
Библиотека Python для Zabbix . . . 2335
Установка . . . 2335
Краткое руководство . . . 2336
Использование Zabbix API . . . 2339
Сбор данных с помощью агента Zabbix . . . 2342
Отправка данных на сервер Zabbix или прокси . . . 2344
Отладочное журналирование . . . 2347
Модули . . . 2348
Файловая структура модуля . . . 2349
Виджеты . . . 2356
Руководства . . . 2365

Примеры . . . 2391
Изменения в разработке расширений . . . 2392

Справочные страницы Zabbix **2392**

zabbix_agent2 . . . 2392
НАИМЕНОВАНИЕ . . . 2392
СИНТАКСИС . . . 2392
ОПИСАНИЕ . . . 2392
ПАРАМЕТРЫ . . . 2392
ФАЙЛЫ . . . 2393
СМОТРИТЕ ТАКЖЕ . . . 2393
Указатель . . . 2393
zabbix_agentd . . . 2394
НАИМЕНОВАНИЕ . . . 2394
СИНТАКСИС . . . 2394
ОПИСАНИЕ . . . 2394
ПАРАМЕТРЫ . . . 2394
ФАЙЛЫ . . . 2395
СМОТРИТЕ ТАКЖЕ . . . 2395
Содержание . . . 2395
zabbix_get . . . 2396
НАИМЕНОВАНИЕ . . . 2396
СИНТАКСИС . . . 2396
ОПИСАНИЕ . . . 2396
ПАРАМЕТРЫ . . . 2396
ПРИМЕРЫ . . . 2397
СМОТРИТЕ ТАКЖЕ . . . 2397
Содержание . . . 2397
zabbix_js . . . 2398
НАИМЕНОВАНИЕ . . . 2398
СИНТАКСИС . . . 2398
ОПИСАНИЕ . . . 2398
ПАРАМЕТРЫ . . . 2398
ПРИМЕРЫ . . . 2398
СМОТРИТЕ ТАКЖЕ . . . 2399
Оглавление . . . 2399
zabbix_proxy . . . 2399
НАИМЕНОВАНИЕ . . . 2399
СИНТАКСИС . . . 2399
ОПИСАНИЕ . . . 2399
ПАРАМЕТРЫ . . . 2399
ФАЙЛЫ . . . 2400
СМОТРИТЕ ТАКЖЕ . . . 2401
Содержание . . . 2401
zabbix_sender . . . 2401
НАИМЕНОВАНИЕ . . . 2401
СИНТАКСИС . . . 2401
ОПИСАНИЕ . . . 2402
ПАРАМЕТРЫ . . . 2402
СТАТУС ЗАВЕРШЕНИЯ . . . 2404
ПРИМЕРЫ . . . 2404
СМОТРИТЕ ТАКЖЕ . . . 2405
Оглавление . . . 2405
zabbix_server . . . 2405
НАИМЕНОВАНИЕ . . . 2405
СИНТАКСИС . . . 2405
ОПИСАНИЕ . . . 2405
ПАРАМЕТРЫ . . . 2405
ФАЙЛЫ . . . 2407
СМОТРИТЕ ТАКЖЕ . . . 2407
Содержание . . . 2407
zabbix_web_service . . . 2407
НАИМЕНОВАНИЕ . . . 2408

СИНТАКСИС 2408
ОПИСАНИЕ 2408
ПАРАМЕТРЫ 2408
ФАЙЛЫ 2408
СМОТРИТЕ ТАКЖЕ 2408
Оглавление 2408

Уведомление об авторских правах **2408**

Руководство пользователя

Добро пожаловать в документацию Zabbix

Ваш основной ресурс для работы с мониторингом Zabbix — от базовых настроек до расширенных конфигураций. Это руководство охватывает всё необходимое для установки, настройки и эксплуатации Zabbix.

Что нового в Zabbix 8.0

Начало работы с Zabbix

Установка Пошаговые инструкции по установке Zabbix на предпочитаемую вами платформу с описанием различных операционных систем и конфигураций баз данных.

Требования Список поддерживаемых платформ и программных предварительных требований, который поможет вам подготовить среду для успешного развертывания Zabbix.

Краткие руководства Краткие практические инструкции, ориентированные на выполнение конкретных задач, которые помогут вам освоить основы — от первых шагов настройки до получения первого оповещения о проблеме.

Zabbix Cloud

Начало работы с Zabbix Cloud Пошаговые инструкции, которые помогут быстро запустить ваш экземпляр Zabbix Cloud, включая первоначальную настройку и подключение отслеживаемых устройств.

Zabbix Cloud и локальное развертывание Сравните различия в управлении, масштабируемости и обслуживании между двумя вариантами развертывания и узнайте, какой из них сможет наиболее эффективно поддержать ваши рабочие процессы.

Знакомство с Zabbix Cloud Получите обзор Zabbix Cloud, включая его ключевые возможности, преимущества и то, что входит в подписку. Ответы на распространенные вопросы по настройке и использованию см. в FAQ.

Центр разработчика

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 Описание структуры и логики виджетов, а также инструкции по созданию пользовательских элементов панели мониторинга, 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 на всех уровнях владения Zabbix.

Блог Zabbix Новости, руководства и практические примеры, подготовленные командой Zabbix и участниками сообщества.

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

1 Установка и первые шаги

Чтобы настроить Zabbix с нуля, выполните следующий процесс:

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

Ниже приведены более подробные сведения по каждому шагу.

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

- **Операционная система Linux:** Ubuntu, Debian, CentOS и 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 вам может потребоваться только настроить сервер базы данных и веб-сервер.

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

Установка Zabbix Самый **простой** способ установки Zabbix — из **пакетов дистрибутива**. Пакеты устанавливают второстепенные программные зависимости.

Точные инструкции по установке, адаптированные для вашей операционной системы, доступны на **сайте Zabbix**.

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

Частью процесса установки является создание базы данных (для хранения данных конфигурации и истории) и импорт в нее начальных данных.

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

Вам будет предложено 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 кратких руководствах объясняется, с какими шаблонами следует работать, чтобы начать мониторинг популярных ресурсов.

Вы можете начать с краткого руководства по мониторингу сервера Linux.

1 Установка веб-интерфейса

В этом разделе приведены пошаговые инструкции по установке веб-интерфейса Zabbix.
Выполните эти шаги после того, как вы установили серверную часть Zabbix — сервер Zabbix, агент и базу данных.

Веб-интерфейс Zabbix написан на PHP, поэтому для его запуска требуется веб-сервер с поддержкой PHP.

> **Note:**
> Вы можете найти более подробную информацию о настройке SSL для веб-интерфейса Zabbix, обращаясь к этим наилучшим практикам.

Экран приветствия

Откройте URL веб-интерфейса Zabbix в браузере. Если вы установили Zabbix из пакетов, URL будет следующим:

- для Apache: `http://<server_ip_or_name>/zabbix`
- для Nginx: `http://<server_ip_or_name>`

Вы должны увидеть первый экран мастера установки веб-интерфейса.

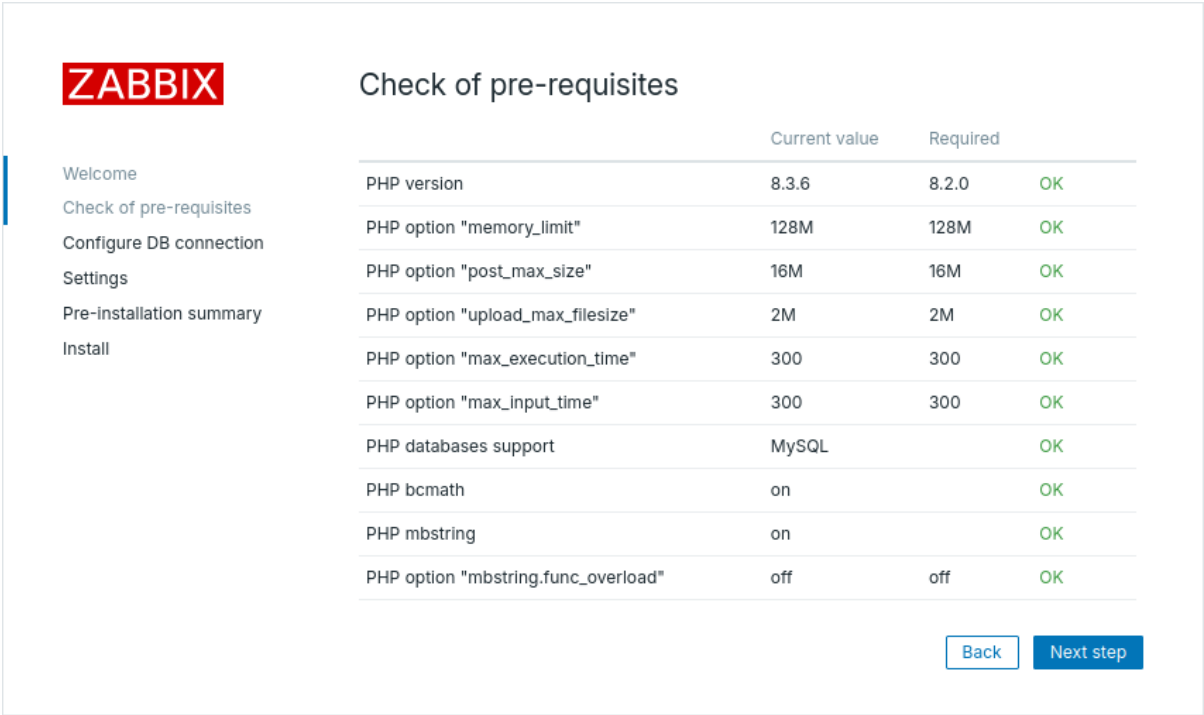
Используйте выпадающее меню *Default language*, чтобы изменить язык системы по умолчанию и продолжить процесс установки на выбранном языке (необязательно). Для получения дополнительной информации см. Установка дополнительных языков веб-интерфейса.

Обратите внимание, что установка языка *English (en_US)* также включит в веб-интерфейсе американский формат времени/даты.



Проверка предварительных требований

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



Предварительное требование	Минимальное значение	Описание
Версия PHP	8.2.0	
Параметр PHP "memory_limit"	128MB	В php.ini: memory_limit = 128M
Параметр PHP "post_max_size"	16MB	В php.ini: post_max_size = 16M
Параметр PHP "upload_max_filesize"	2MB	В php.ini: upload_max_filesize = 2M
Параметр PHP "max_execution_time"	300 секунд	В php.ini: max_execution_time = 300 (также допускаются значения 0 и -1)
Параметр PHP "max_input_time"	300 секунд	В php.ini: max_input_time = 300 (также допускаются значения 0 и -1)

Предварительное требование	Минимальное значение	Описание
Поддержка баз данных PHP	Одна из: MySQL, PostgreSQL	См. Requirements для получения списка всех обязательных и необязательных расширений PHP. Обратите внимание, что невыполненные необязательные предварительные требования отображаются с красным статусом *Warning*, и процесс установки может быть продолжен, даже если они не выполнены.
PHP bcmath	должен быть включен	
PHP mbstring	должен быть включен	
Параметр PHP "mbstring.func_overload"	должен быть отключен	В php.ini: mbstring.func_overload = 0
Параметр PHP "session.auto_start"	должен быть отключен	В php.ini: session.auto_start = 0
Параметр PHP "arg_separator.output"	&	В php.ini: arg_separator.output = "&" (также допускается значение "&")

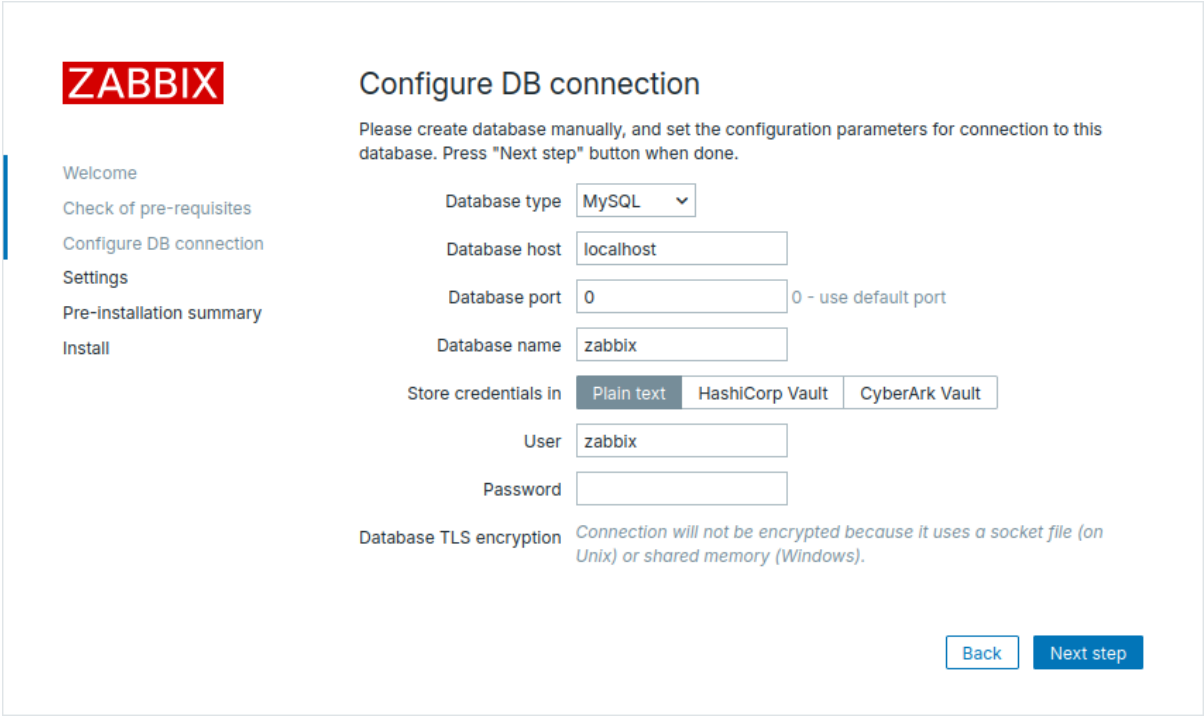
Attention:
Если необходимо изменить пользователя Apache или группу пользователей, проверьте права доступа к папке сессий; в противном случае установка Zabbix может не продолжиться.

Настройка подключения к БД

Введите данные для подключения к базе данных. База данных Zabbix должна быть уже создана.

Для MySQL ввод `localhost` или оставление поля *Database host* пустым приводит к подключению через Unix-сокет по умолчанию. Форма установки не содержит отдельного поля *Database socket*, поэтому для использования пользовательского сокета настройте его в параметрах сервера Zabbix (например, с помощью `DBSocket=` в zabbix_server.conf). Это позволяет веб-интерфейсу использовать те же настройки подключения сервера к базе данных.

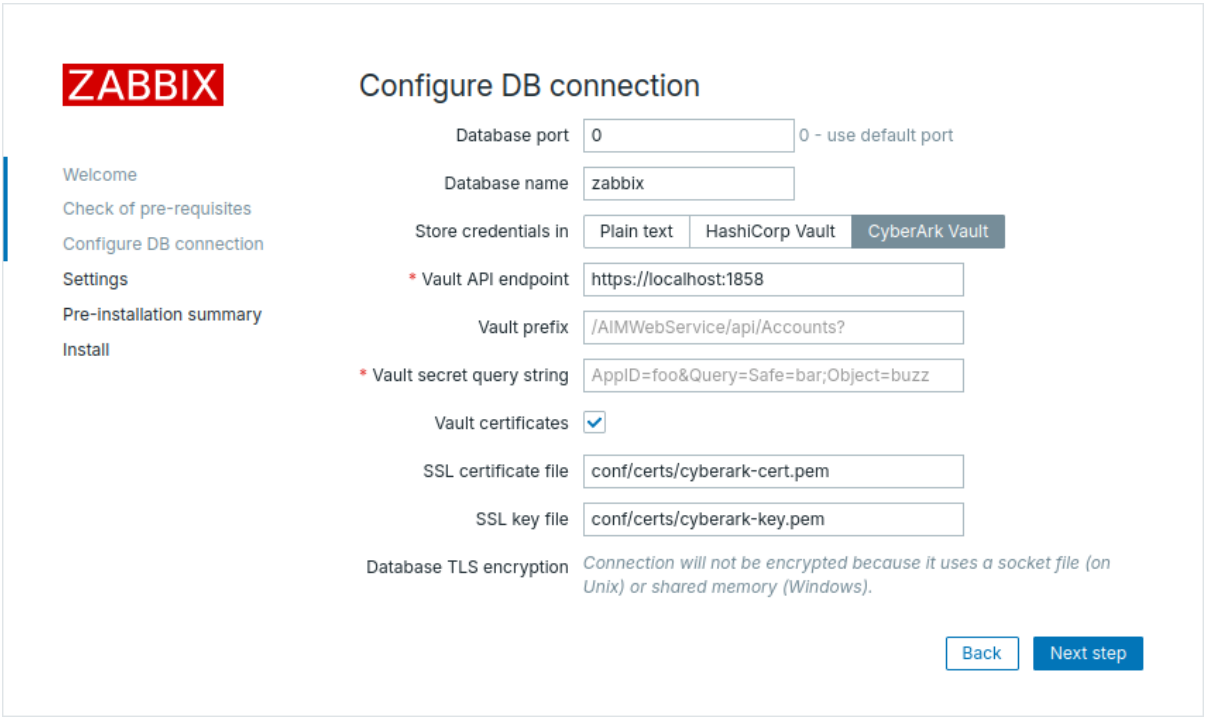
Для PostgreSQL, если поле *Database host* оставлено пустым, используется Unix-domain socket по умолчанию. Если указан путь к сокету (например, `/var/run/pgbouncer`), вместо этого будет использоваться данный Unix-domain socket.



Если отмечен параметр *Database TLS encryption*, в форме появятся дополнительные поля для настройки TLS-подключения к базе данных (только для MySQL или PostgreSQL).

Если в *Store credentials in* выбрано HashiCorp Vault или CyberArk Vault, станут доступны дополнительные параметры:

- для HashiCorp Vault: конечная точка Vault API, префикс хранилища, путь к секрету и токен аутентификации;
- для CyberArk Vault: конечная точка Vault API, префикс хранилища, строка запроса секрета и сертификаты. При установке флажка *Vault certificates* появятся два новых поля для указания путей к файлу SSL-сертификата и файлу SSL-ключа.

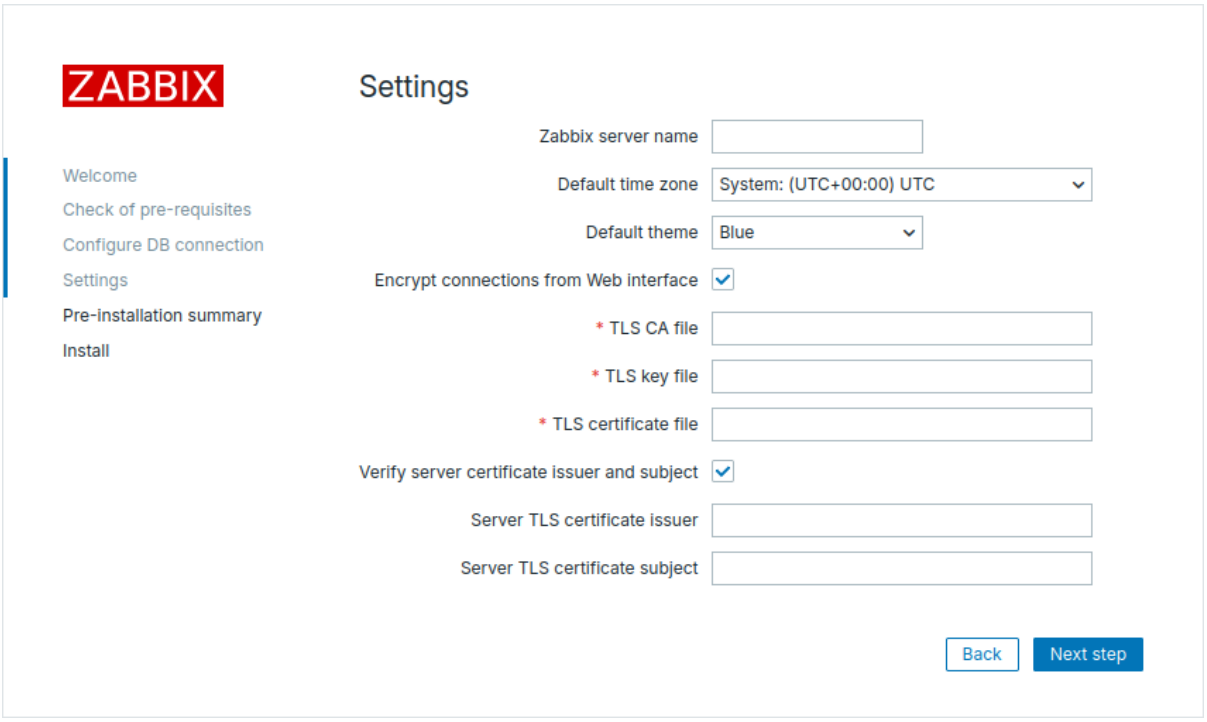


Настройки

Указывать имя сервера Zabbix необязательно, однако если оно задано, оно будет отображаться в строке меню и заголовках страниц.

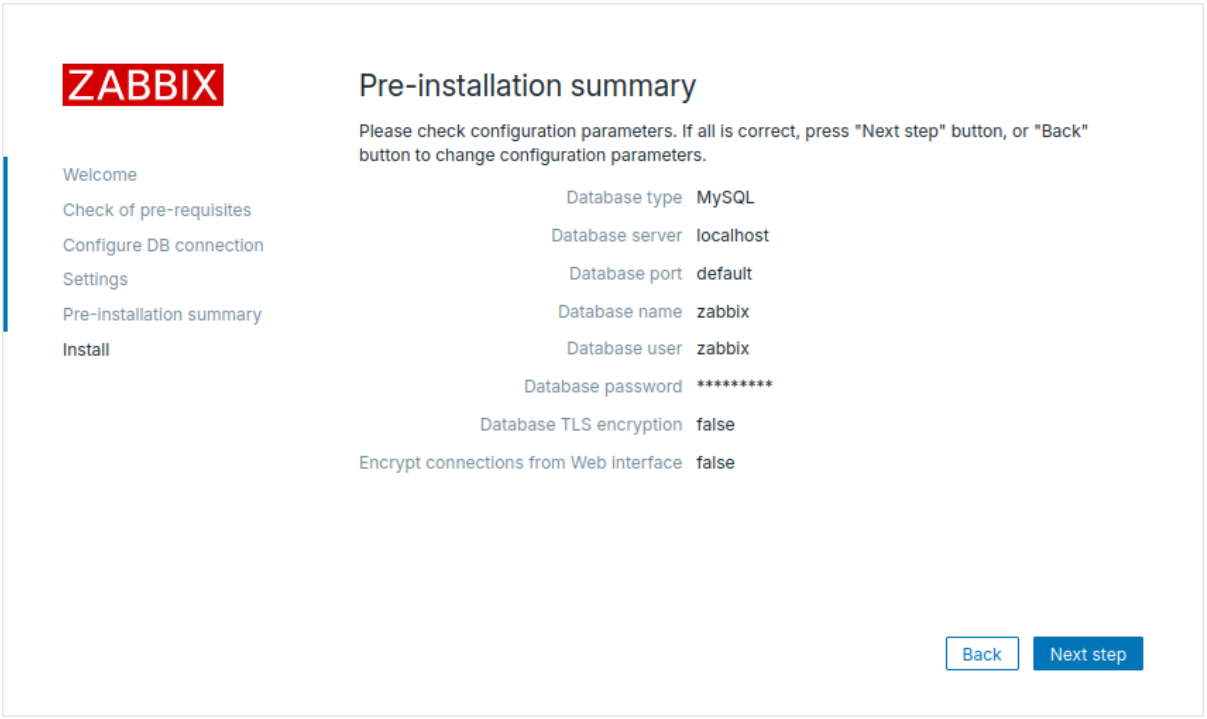
Установите часовой пояс time zone и тему по умолчанию для веб-интерфейса.

Если установлен флажок *Encrypt connections from Web interface*, то в форме появятся дополнительные поля для настройки TLS-соединения между сервером Zabbix и веб-интерфейсом.

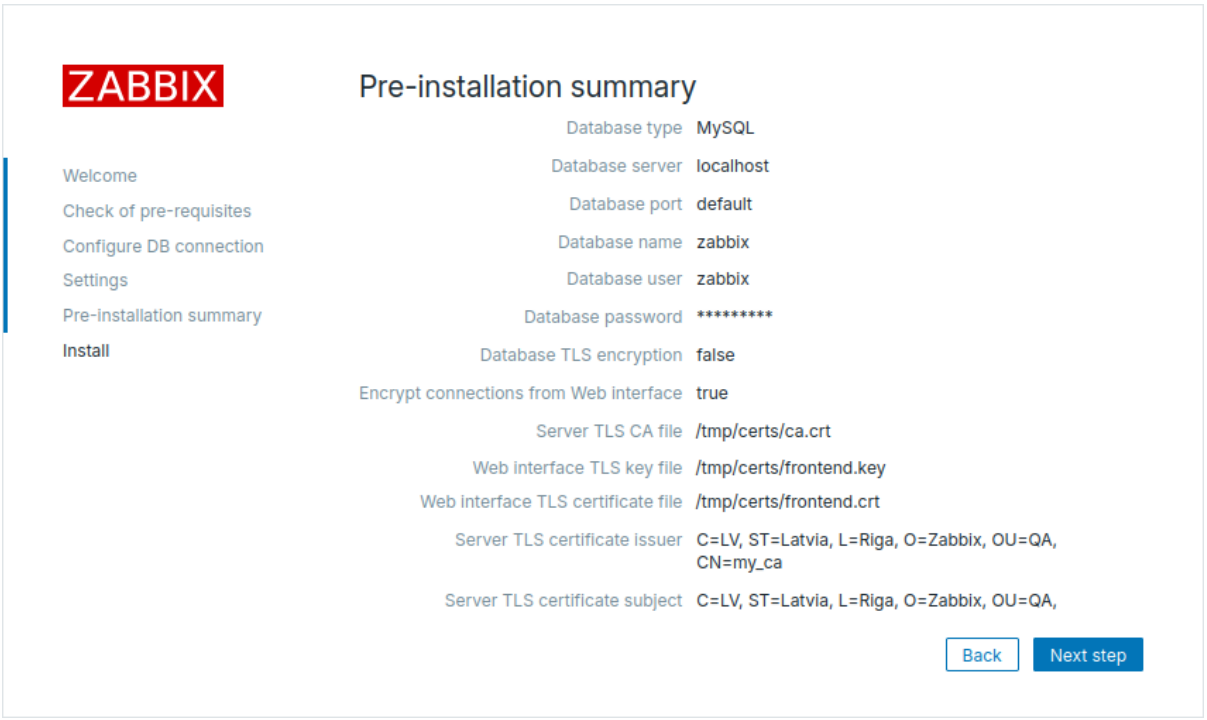


Сводка перед установкой

Просмотрите сводку настроек.

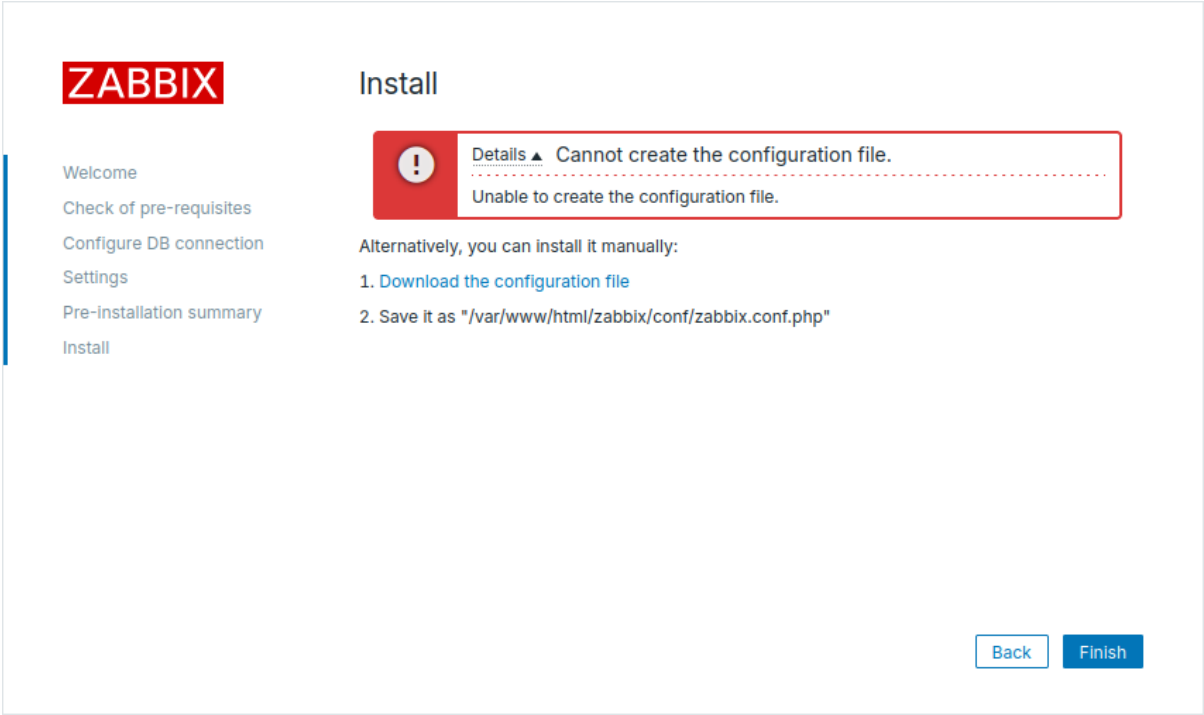


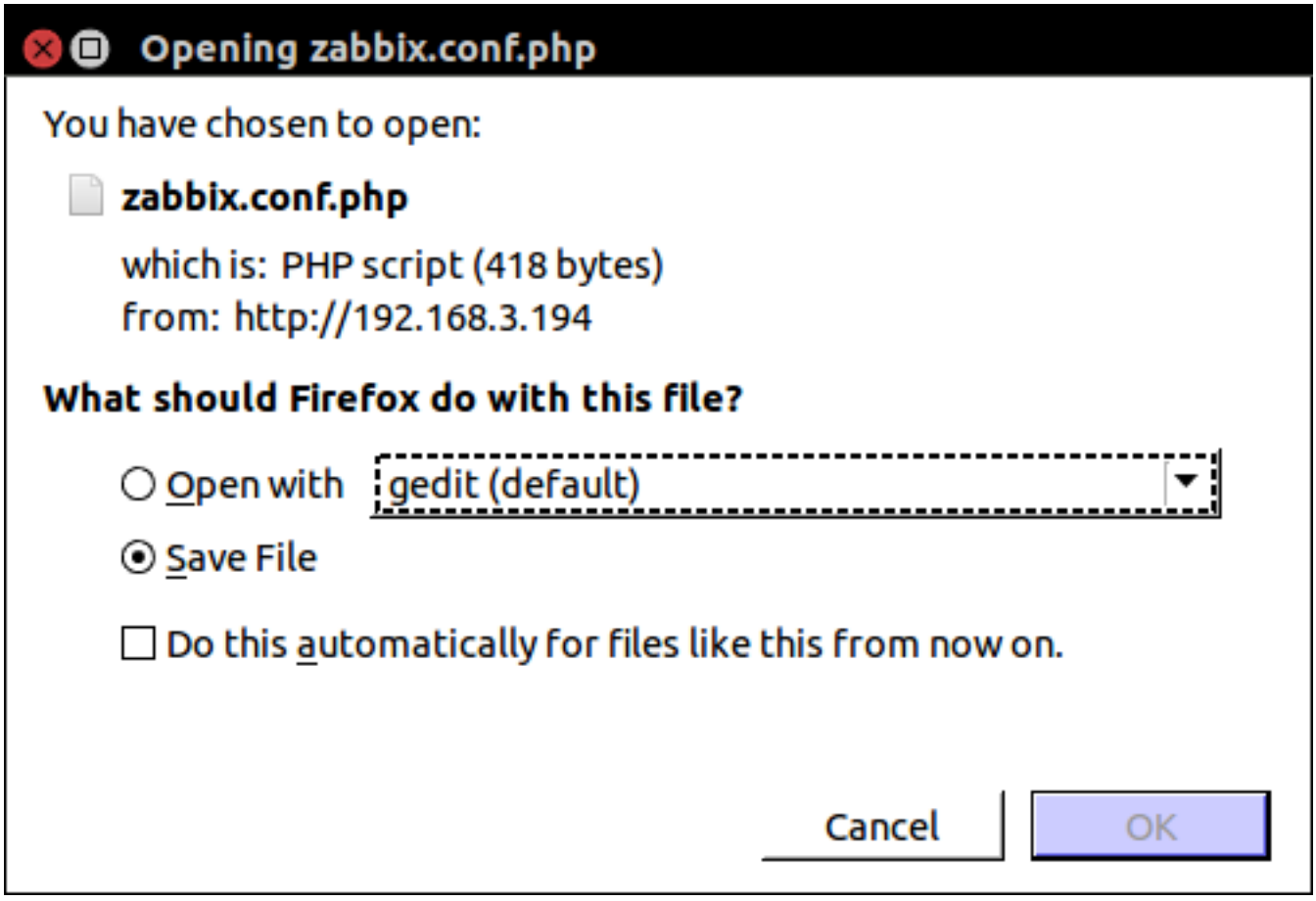
Подстраница отобразит данные, если была добавлена конфигурация TLS.



Установка

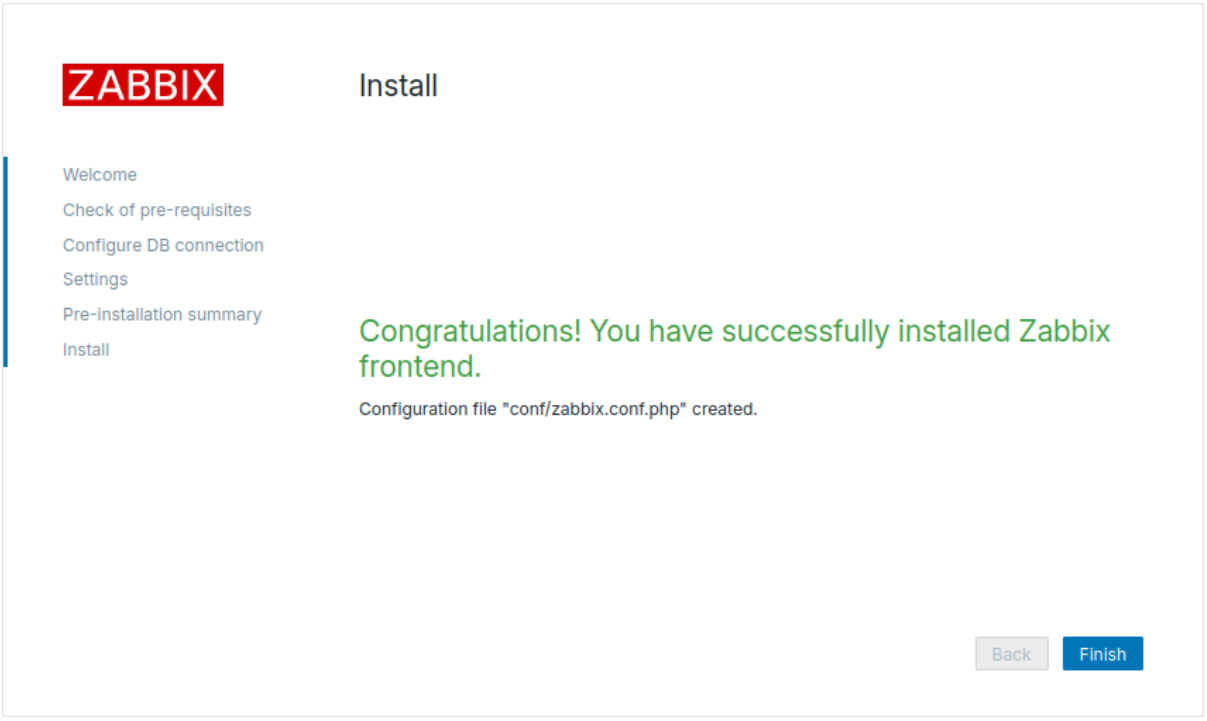
Если вы устанавливаете Zabbix из исходных кодов, загрузите файл конфигурации и поместите его в conf/ в подкаталоге HTML-документов веб-сервера, куда вы скопировали PHP-файлы Zabbix.





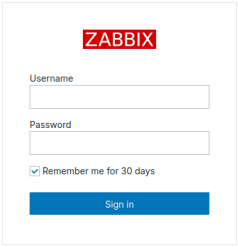
Note:
Если у пользователя веб-сервера есть права на запись в каталог conf/, файл конфигурации будет сохранён автоматически, и можно будет сразу перейти к следующему шагу.

Завершите установку.



Вход в систему

Веб-интерфейс Zabbix готов к работе! Имя пользователя по умолчанию — **Admin**, пароль — **zabbix**.



Перейдите к руководству по мониторингу Linux.

2 Краткие руководства по началу работы

После того как вы установили Zabbix и его веб-интерфейс, пора начинать мониторинг.

Используйте эти краткие руководства, чтобы начать мониторинг популярных ресурсов:

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

Краткие руководства по началу работы сосредоточены на использовании официальных шаблонов Zabbix для мониторинга.
Шаблон — это предварительно настроенный набор элементов мониторинга.
Шаблон позволяет масштабировать мониторинг: вы можете протестировать шаблон на одной цели мониторинга (устройстве, приложении и т. д.), настроить его, а затем применить к нескольким аналогичным целям мониторинга.

Хотя шаблоны хорошо подходят для мониторинга «из коробки», вас также может заинтересовать настройка их элементов.

Чтобы лучше понять основные элементы, из которых состоит шаблон, обратитесь к разделу Базовая настройка.

Базовая настройка

В этих руководствах представлены основные элементы настройки Zabbix:

- Вход в систему и настройка пользователя
- Новый узел сети
- Новый элемент данных
- Новый триггер

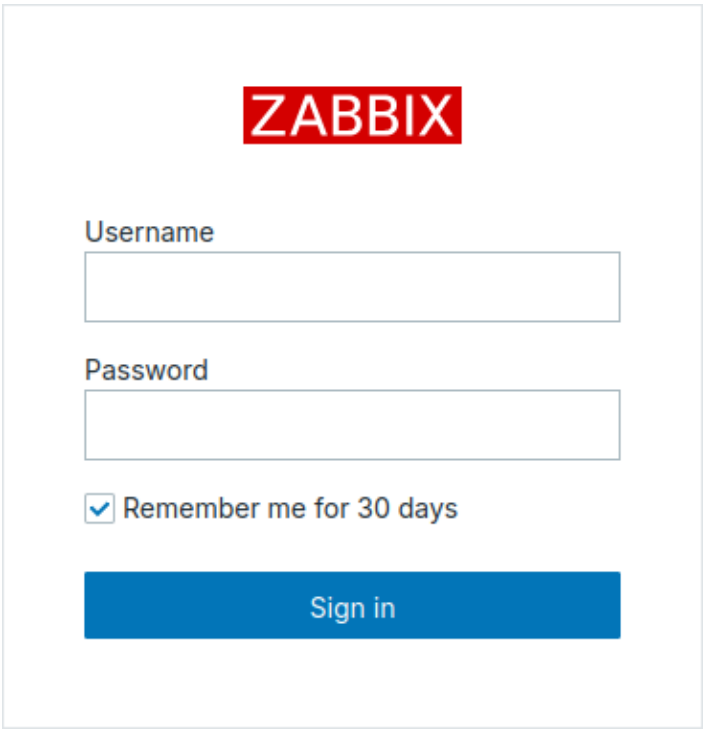
- Получение уведомления о проблеме
- Новый шаблон

1 Вход в систему и настройка пользователя

Обзор

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

Вход



Это экран приветствия 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 изменить пароль по умолчанию для учетной записи Admin сразу после первого входа в систему.

Постоянный вход в систему

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

Запомнить на 30 дней включено:

- Ваш сеанс остается активным в течение 30 дней.
- *Автовыход* переопределяется, сохраняя ваш вход в систему до окончания этого периода.
- При последующих посещениях в течение 30 дней вы будете автоматически входить в систему без повторного ввода учетных данных.

Запомнить на 30 дней отключено:

- Любой ранее включенный автоматический вход сбрасывается.
- Сеанс завершится в соответствии с настроенным интервалом *Автовыход*.

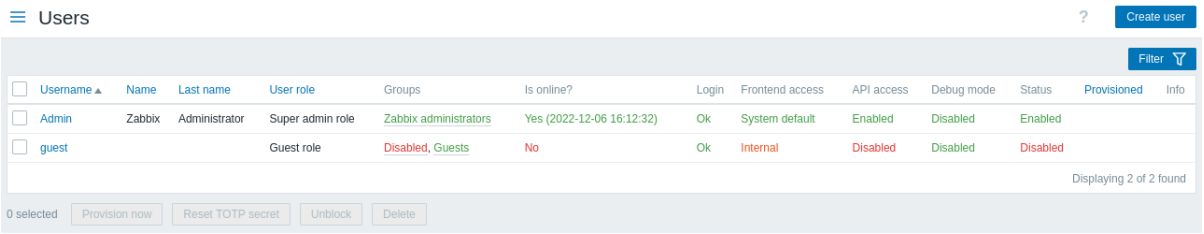
Защита от атак методом перебора

В случае пяти последовательных неудачных попыток входа веб-интерфейс Zabbix приостановит возможность входа на 30 секунд, чтобы предотвратить атаки методом перебора и по словарю.

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

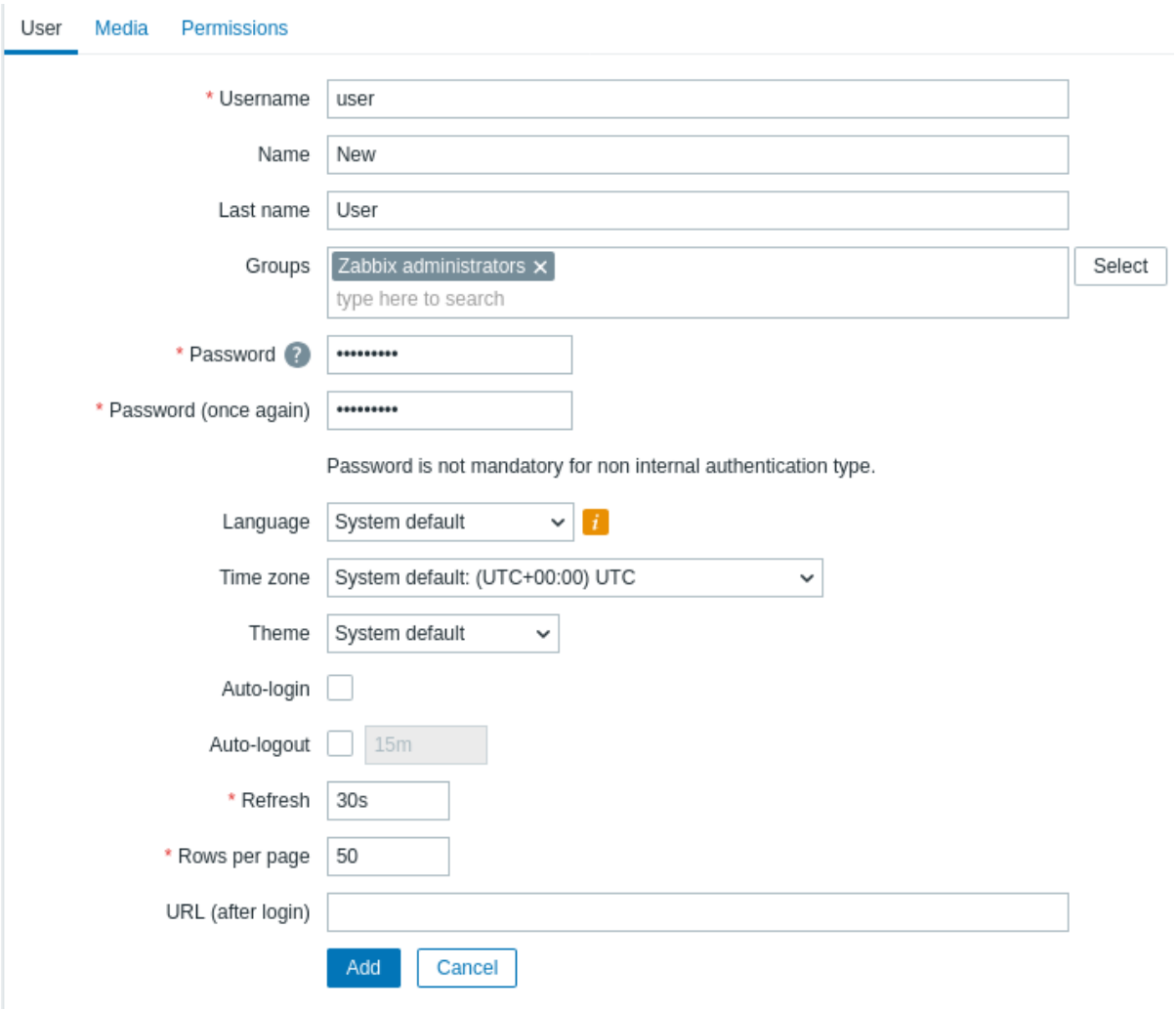
Добавление пользователя

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



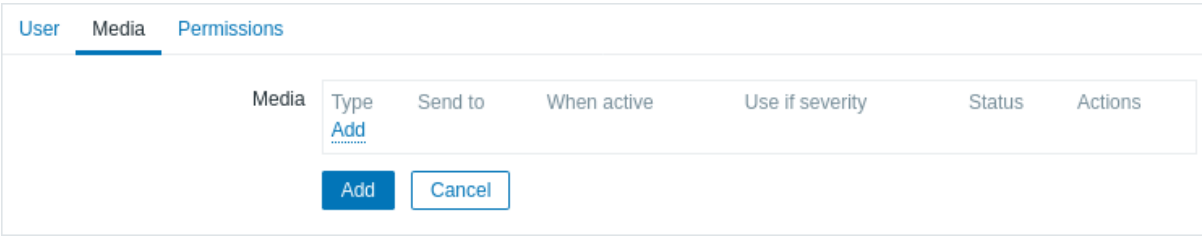
Чтобы добавить нового пользователя, выберите *Create user* в правом верхнем углу.

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



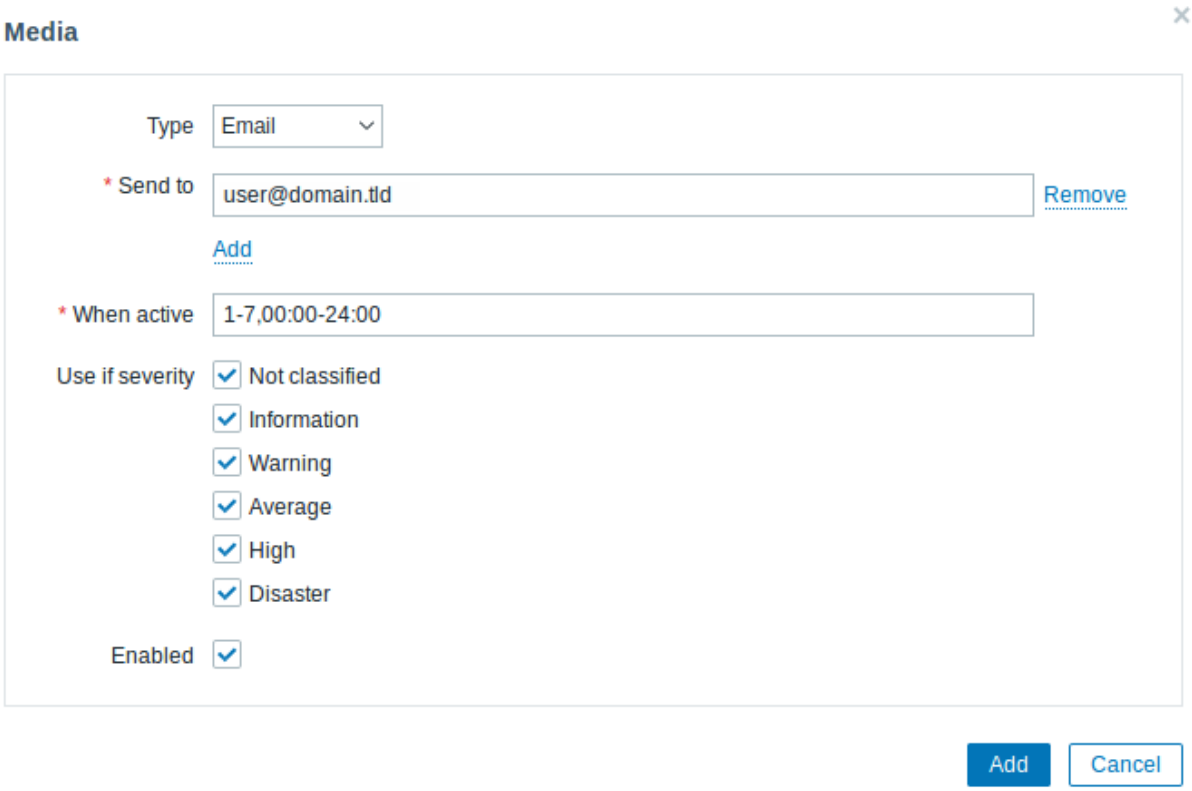
Все обязательные поля ввода отмечены красной звездочкой. Подробную информацию о полях ввода в этой форме настройки см. на странице Настройки пользователя.

По умолчанию для новых пользователей не определены способы оповещения (media). Чтобы создать такой способ, перейдите на вкладку 'Media' и нажмите *Add*.



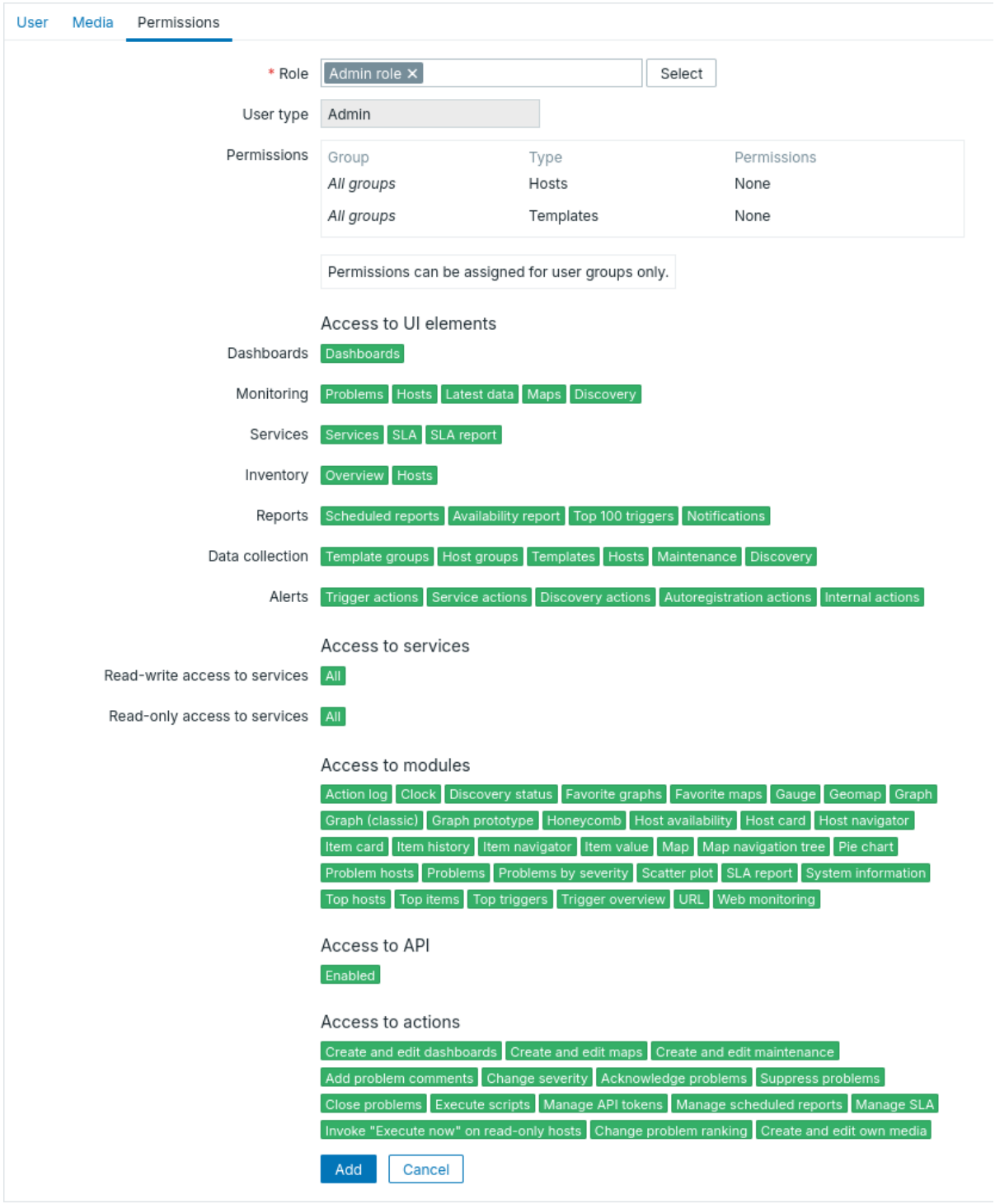
Во всплывающем окне введите адрес электронной почты пользователя.

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



Нажмите *Add*, чтобы сохранить способ оповещения, 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 в списке. Например, выберите *Admin role*, чтобы разрешить доступ ко всем разделам веб-интерфейса Zabbix, кроме Administration. Позже вы сможете изменить права доступа или создать дополнительные роли пользователей. После выбора роли права доступа появятся на этой же вкладке:

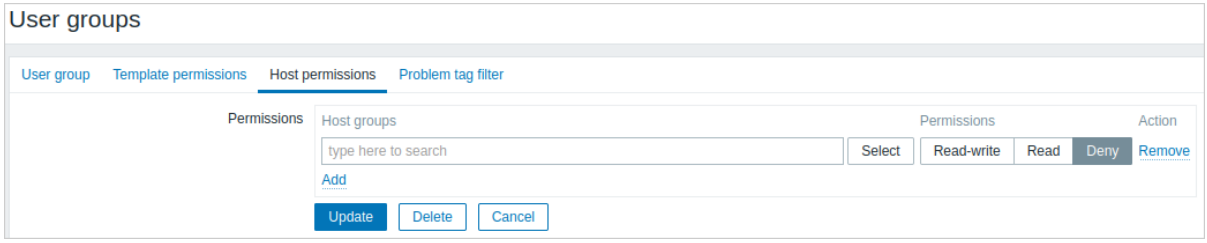


Нажмите *Add* в форме свойств пользователя, чтобы сохранить пользователя. Новый пользователь появится в списке пользователей.

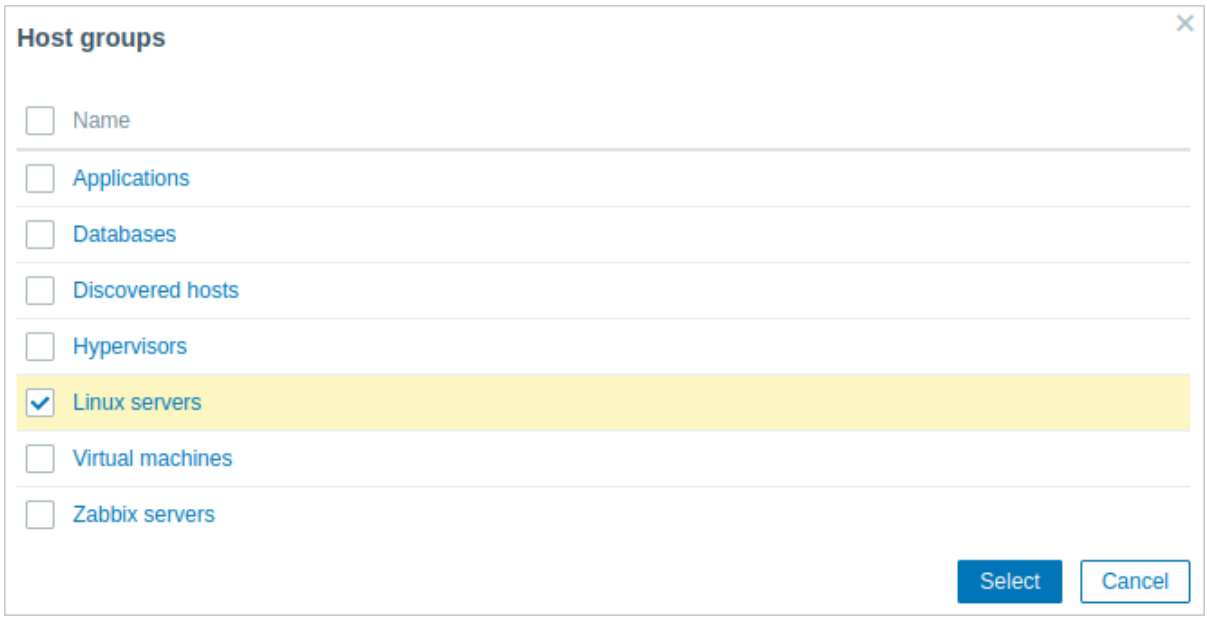


Добавление прав доступа

По умолчанию новый пользователь не имеет прав доступа к узлам сети и шаблонам. Чтобы предоставить пользователю права, щелкните по группе пользователя в столбце *Groups* (в данном случае — 'Zabbix administrators'). В форме свойств *User groups* перейдите на вкладку *Host permissions*, чтобы назначить права доступа группам узлов сети. Щелкните по Add, чтобы отобразилось поле выбора группы узлов сети:



Затем нажмите *Select* рядом с полем, чтобы увидеть список групп узлов сети. Этот пользователь должен иметь доступ только на чтение к группе *Linux servers*, поэтому установите соответствующий флажок в списке и нажмите *Select*, чтобы подтвердить выбор.



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

Чтобы предоставить права доступа к шаблонам, необходимо перейти на вкладку *Template permissions* и указать группы шаблонов. Эти действия аналогичны назначению прав доступа группам узлов сети. Обзор шаблонов доступен в разделе New template этого краткого руководства.

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.

2 Новый узел сети

Обзор

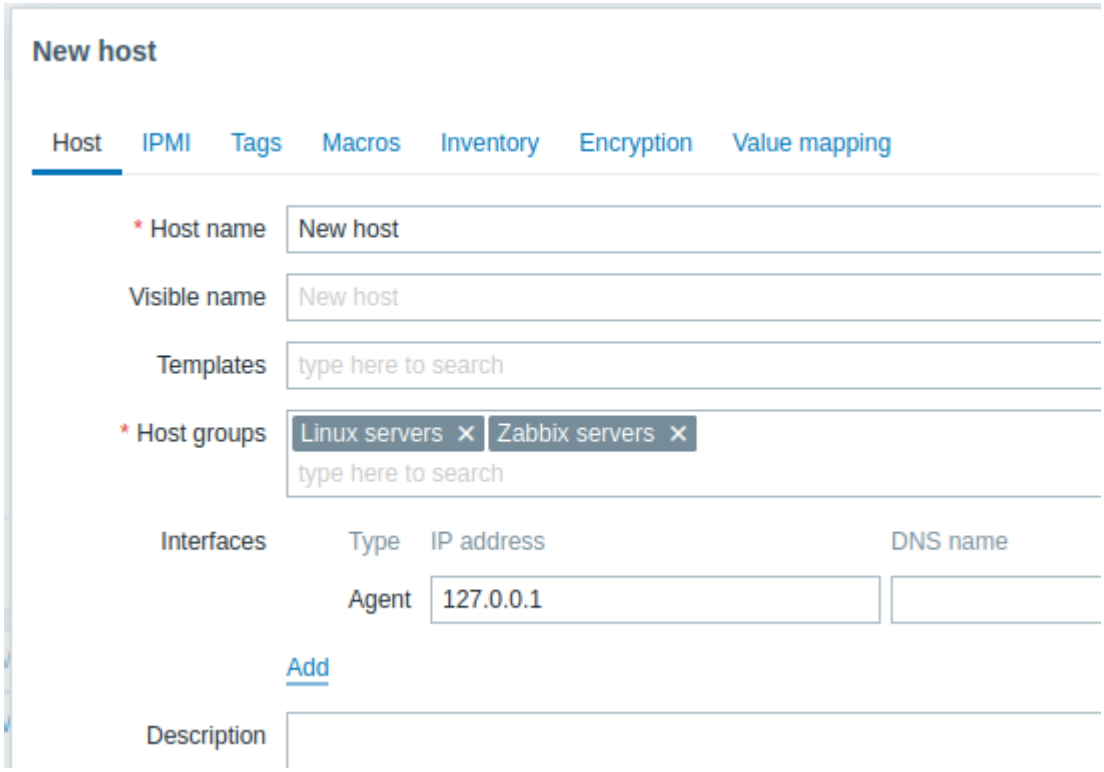
В этом разделе вы узнаете, как настроить новый узел сети.

Узел сети в Zabbix — это сетевая сущность (физическая, виртуальная), которую вы хотите отслеживать. Определение того, что может быть «узлом сети» в Zabbix, достаточно 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 разделах меню *Сбор данных* > *Узлы сети* и *Мониторинг* > *Узлы сети*. Уже существует один предопределенный узел сети с именем "Zabbix server", но мы хотим научиться добавлять другой.

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



Все обязательные поля ввода отмечены красной звездочкой.

Минимально необходимо указать:

Имя узла сети

- Введите имя узла сети. Допускаются буквенно-цифровые символы, 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.

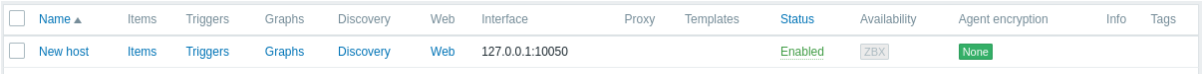
Note:
Все права доступа назначаются группам узлов сети, а не отдельным узлам сети. Поэтому узел сети должен принадлежать как минимум к одной группе.

Интерфейсы: IP-адрес

- Хотя технически это поле не является обязательным, интерфейс узла сети необходим для сбора некоторых метрик. Чтобы использовать пассивные проверки агента Zabbix, укажите в этом поле IP-адрес или DNS агента. Обратите внимание, что в файле конфигурации агента Zabbix в параметре 'Server' также необходимо указать IP-адрес или DNS сервера Zabbix. Если агент Zabbix и сервер Zabbix установлены на одной и той же машине, необходимо указать один и тот же IP/DNS в обоих местах.

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

По завершении нажмите *Добавить*. Новый узел сети должен появиться в списке узлов сети.



Столбец «Доступность» содержит индикаторы доступности узла сети для каждого интерфейса. Мы определили интерфейс агента Zabbix, поэтому можем использовать значок доступности агента (с надписью 'ZBX'), чтобы понять состояние доступности узла сети:

- ZBX - состояние узла сети еще не определено; проверка метрик еще не выполнялась
- ZBX - узел сети доступен, проверка метрик выполнена успешно
- ZBX - узел сети недоступен, проверка метрик завершилась ошибкой (наведите курсор мыши на значок, чтобы увидеть сообщение об ошибке). Возможна ошибка связи, вероятно вызванная неверными учетными данными

интерфейса. Убедитесь, что сервер Zabbix запущен, и также попробуйте обновить страницу позже.

3 Новый элемент данных

Обзор

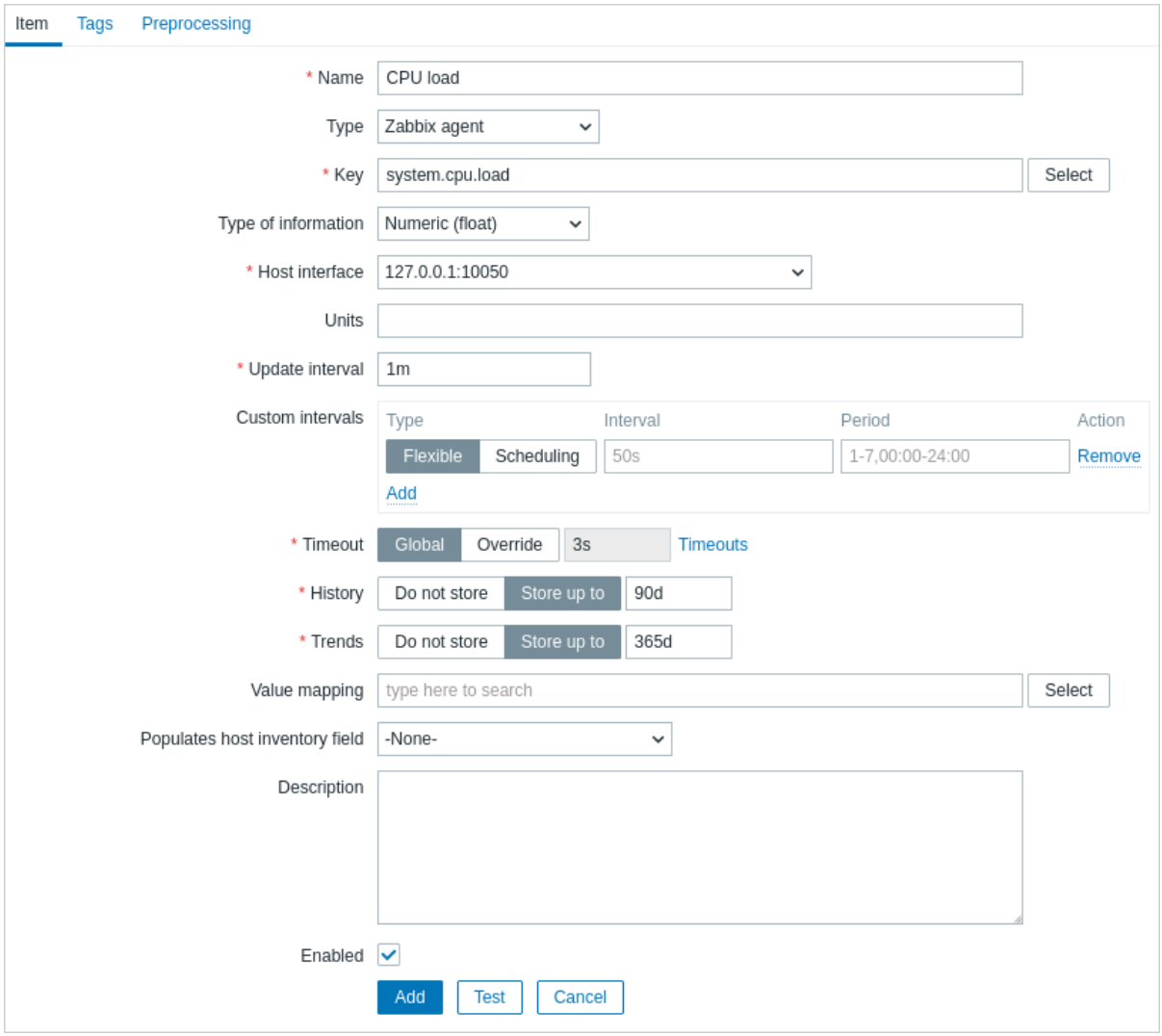
В этом разделе вы узнаете, как настроить элемент данных.

Элементы данных являются основой сбора данных в Zabbix. Без элементов данных нет 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. Поэтому, чтобы настроить пример элемента данных, перейдите в *Сбор данных* > *Узлы сети* и найдите созданный нами "New host".

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



Все обязательные поля ввода отмечены красной звёздочкой.

Для нашего примера элемента данных необходимо ввести следующую основную информацию:

Имя

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

Ключ

- Вручную введите значение *system.cpu.load*. Это техническое имя элемента данных, которое определяет тип информации, которая будет собираться. Этот конкретный ключ — лишь один из предопределённых ключей, поставляемых с Zabbix агентом.

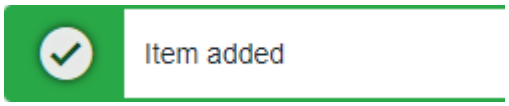
Тип информации

- Этот атрибут определяет формат ожидаемых данных. Для ключа *system.cpu.load* это поле будет автоматически установлено в значение *Числовой (с плавающей точкой)*.

Note:
Вы также можете уменьшить количество дней хранения истории элемента данных до 7 или 14. Это хорошая практика, позволяющая снизить нагрузку на базу данных за счёт уменьшения объёма хранимых исторических значений.

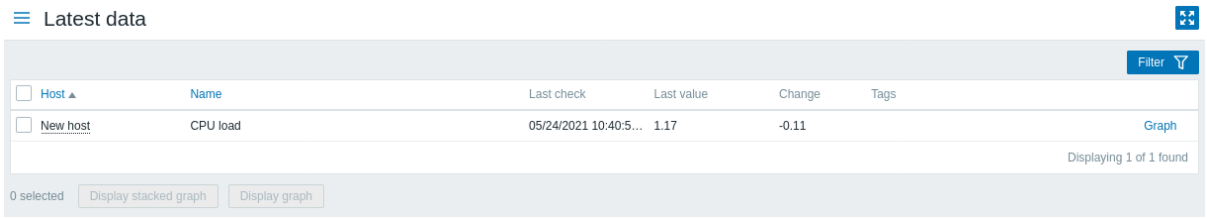
Другие параметры можно оставить без изменений, используя значения по умолчанию.

Когда всё будет готово, нажмите *Добавить*. Новый элемент данных должен появиться в списке элементов данных, и вы увидите сообщение об успешном выполнении.



Просмотр данных

После того как элемент данных создан, вам может быть интересно, действительно ли он собирает данные. Для этого перейдите в *Мониторинг* > *Последние данные*, выберите 'New host' в фильтре и нажмите *Применить*.



При этом получение первых данных может занять до 60 секунд. По умолчанию именно с такой периодичностью сервер считывает изменения конфигурации и подхватывает новые элементы данных для выполнения.

Если в столбце 'Изменение' нет значения, возможно, пока было получено только одно значение. Подождите 30 секунд, пока не поступит следующее значение.

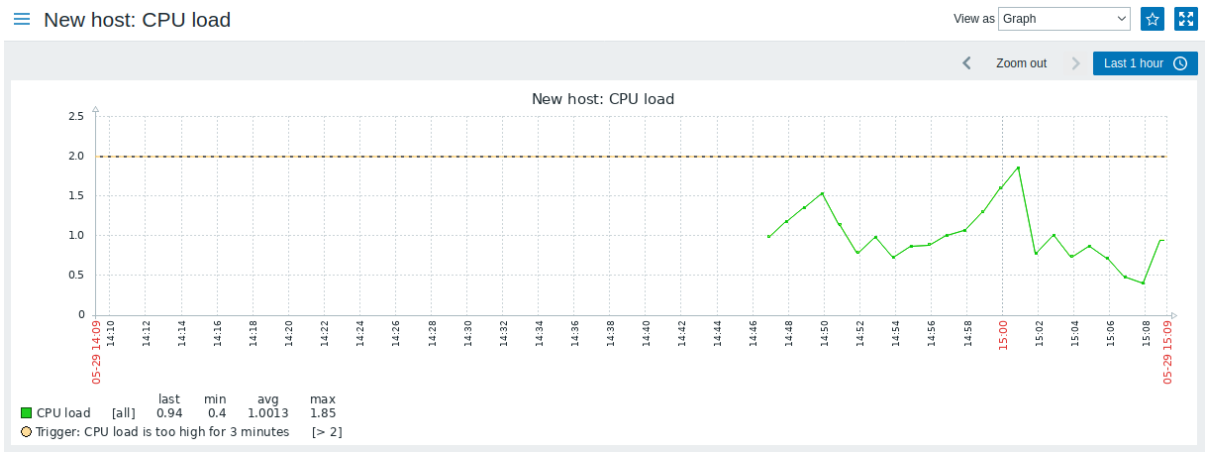
Если вы не видите информацию об элементе данных, как на снимке экрана, убедитесь, что:

- вы заполнили поля элемента данных 'Ключ' и 'Тип информации' в точности как на снимке экрана;
- и агент, и сервер запущены;
- статус узла сети — 'Наблюдается', а его значок доступности зеленый;
- в фильтре узлов сети выбран правильный узел сети;
- элемент данных включен.

Графики

Когда элемент данных некоторое время поработает, возможно, придёт время посмотреть на что-нибудь наглядное. Простые графики доступны для любого отслеживаемого числового элемента данных без какой-либо дополнительной настройки. Эти графики создаются во время выполнения.

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



4 Новый триггер

Обзор

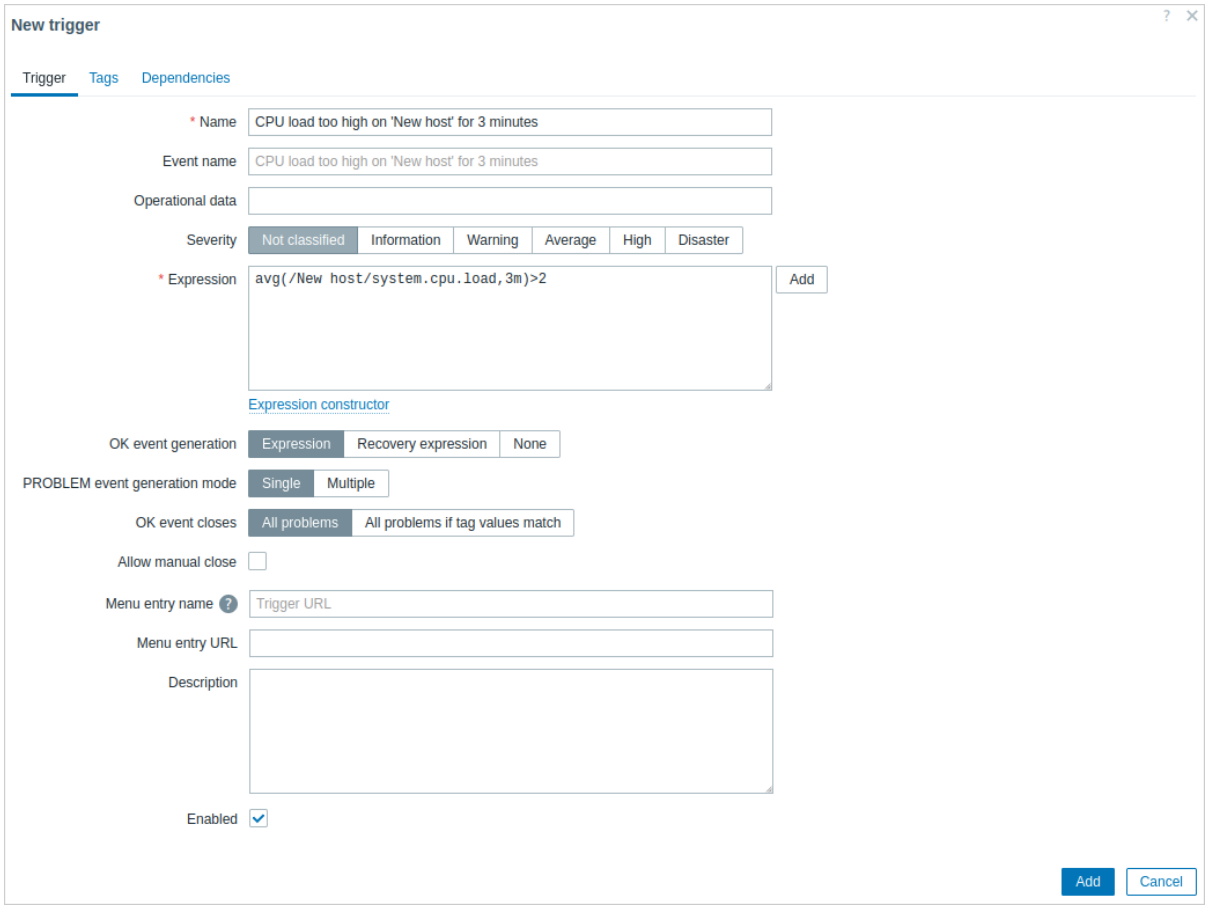
В этом разделе вы узнаете, как настроить триггер.

Элементы данных только собирают данные. Чтобы автоматически оценивать поступающие данные, необходимо определить триггеры. Триггер содержит выражение, которое задает порог того, какой уровень данных считается допустимым.

Если этот уровень превышается поступающими данными, триггер ”срабатывает” или переходит в состояние ’Проблема’, указывая на то, что произошло событие, которое может потребовать внимания. Если уровень снова становится допустимым, триггер возвращается в состояние ’Ок’.

Добавление триггера

Чтобы настроить триггер для нашего элемента данных, перейдите в *Сбор данных* > *Узлы сети*. Нажмите на ссылку *Триггеры* в строке ’New host’, а затем нажмите *Создать триггер*. Откроется форма настройки триггера.



Для нашего триггера здесь необходимо ввести следующую основную информацию:

Имя

- Введите *CPU load 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

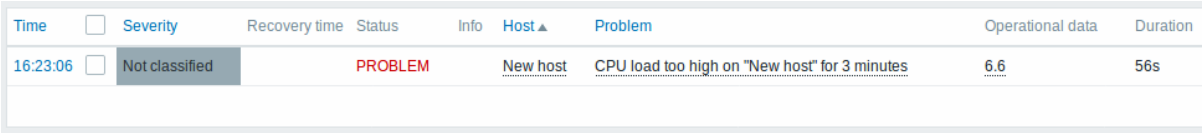
Это выражение триггера. Убедитесь, что выражение введено правильно, вплоть до последнего символа. Ключ элемента данных здесь (system.cpu.load) используется для ссылки на элемент данных. Это конкретное выражение, по сути, означает, что порог проблемы превышен, когда среднее значение загрузки CPU за 3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

По завершении нажмите *Добавить*. Новый триггер должен появиться в списке триггеров.

Отображение состояния триггера

Если триггер определён, вам может быть интересно посмотреть его состояние.

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



Time		Severity	Recovery time	Status	Info	Host ▲	Problem	Operational data	Duration
16:23:06		Not classified		PROBLEM		New host	CPU load too high on "New host" for 3 minutes	6.6	56s

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

5 Получение уведомления о проблеме

Обзор

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

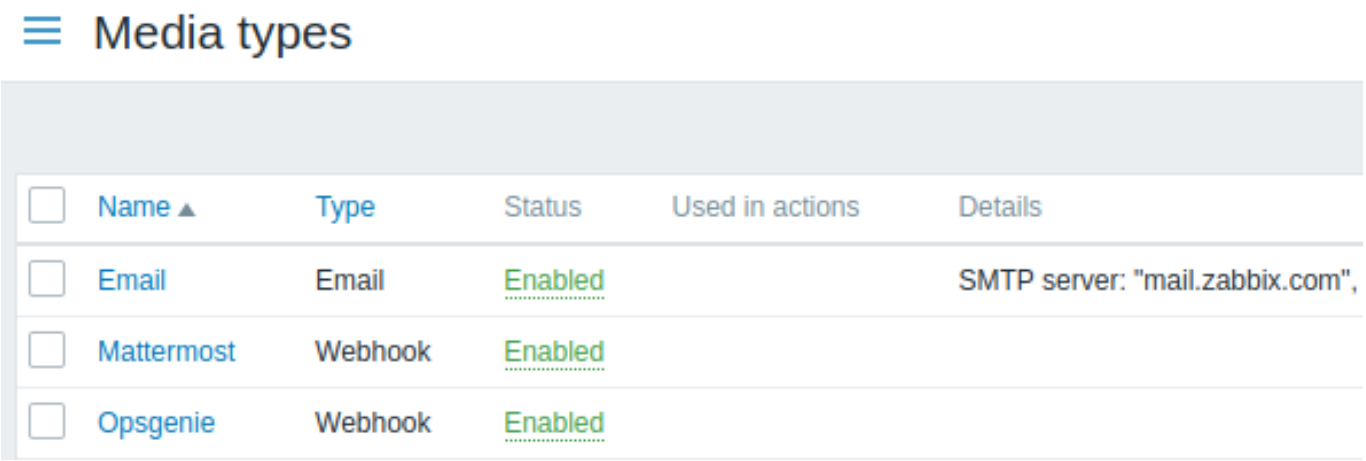
Поскольку элементы данных собирают данные, а триггеры предназначены для ”срабатывания” при возникновении проблемных ситуаций, также полезно иметь механизм оповещения, чтобы уведомлять о важных событиях, даже когда веб-интерфейс Zabbix активно не просматривается.

Именно для этого и нужны уведомления. Поскольку электронная почта является самым популярным способом доставки уведомлений о проблемах, мы рассмотрим, как настроить уведомление по электронной почте.

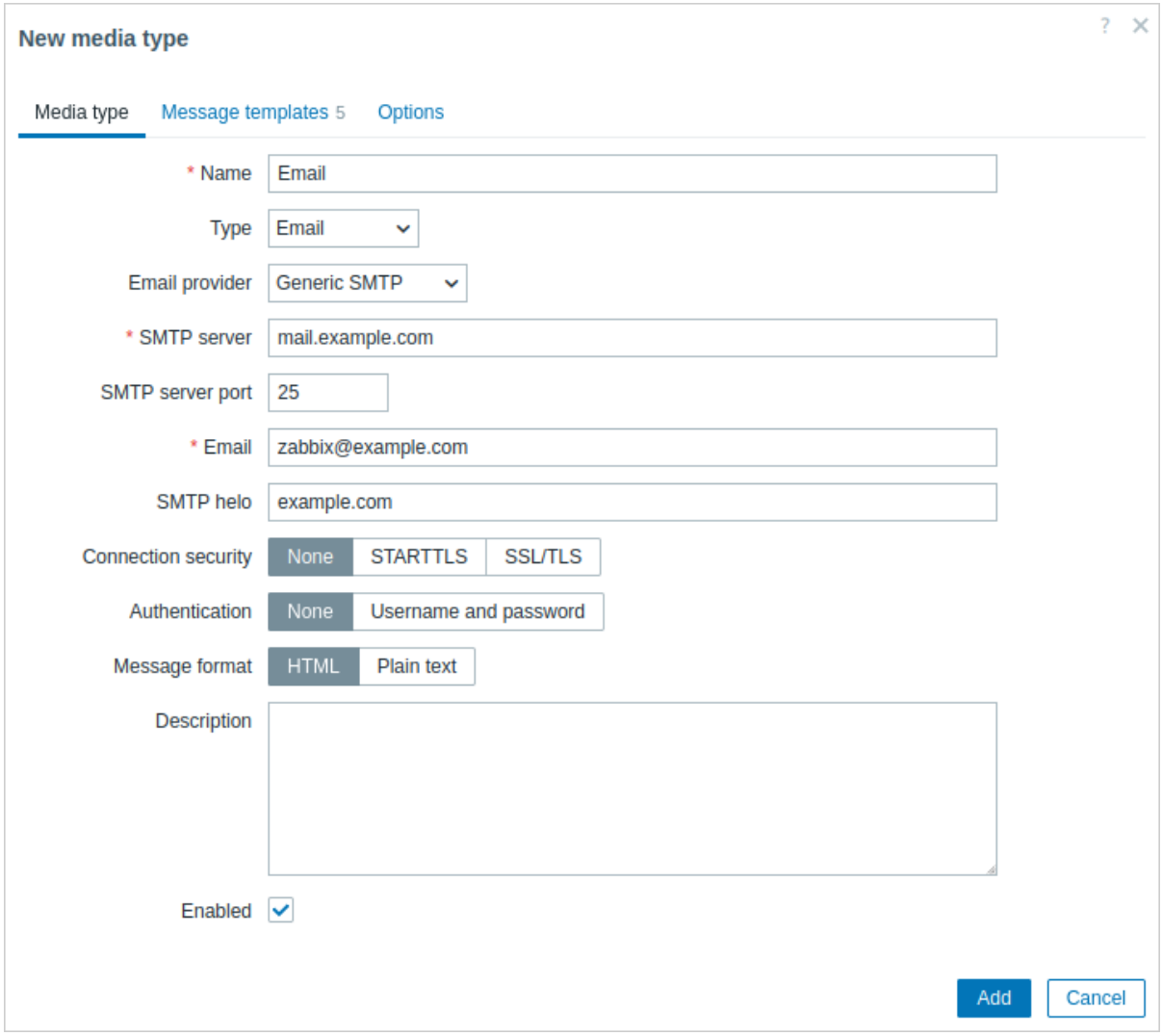
Настройки электронной почты

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

Чтобы настроить параметры электронной почты, перейдите в *Оповещения > Способы оповещений* и нажмите *Email* в списке предопределенных способов оповещений.



Откроется форма настройки параметров электронной почты.



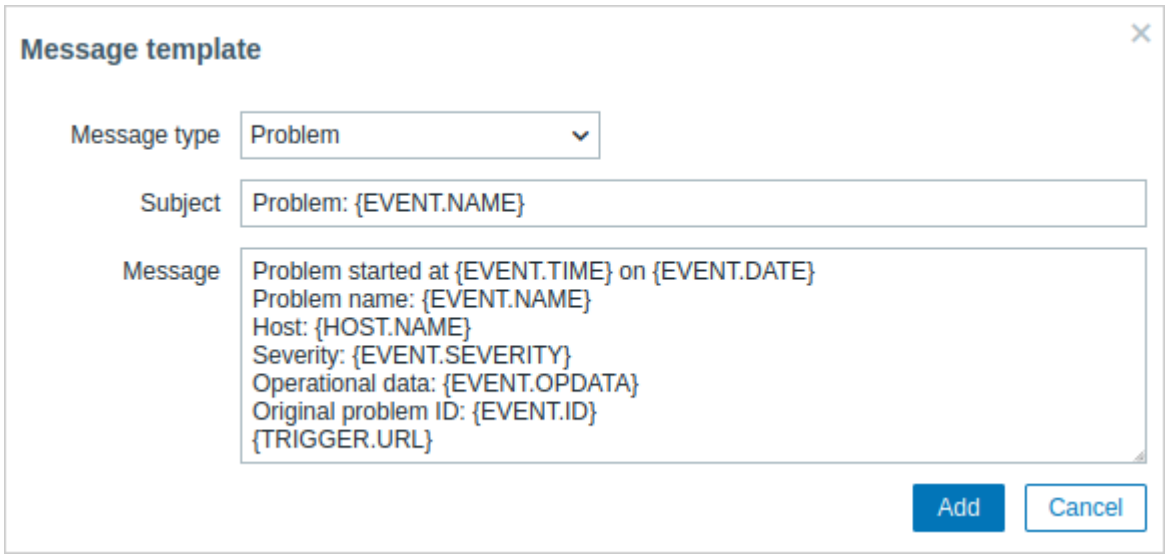
Все обязательные поля ввода отмечены красной звездочкой.

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

Note:
Значение в поле *Email* будет использоваться как адрес отправителя (’From’) для уведомлений, отправляемых из Zabbix.

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

Нажмите *Добавить*, чтобы создать шаблон сообщения, и выберите *Проблема* в качестве типа сообщения.

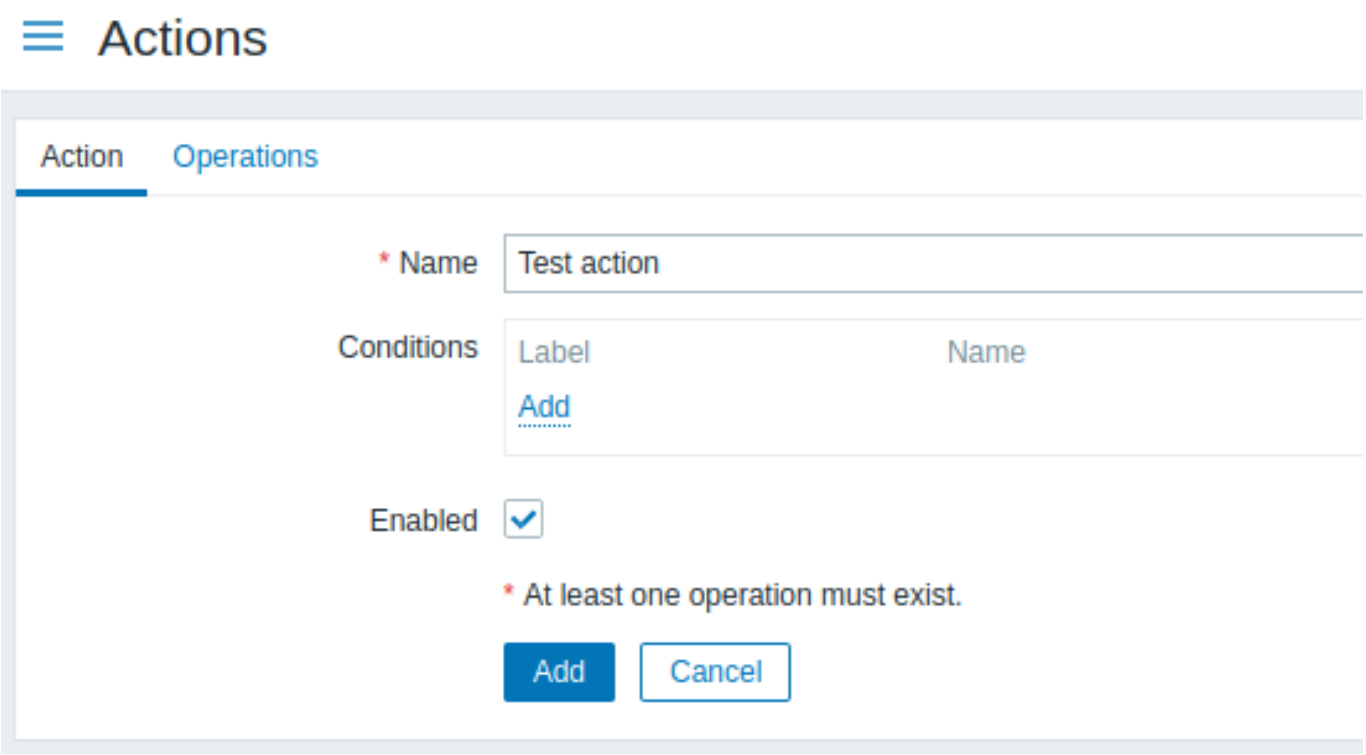


Когда все будет готово, нажмите *Добавить* и сохраните форму.

Теперь *Email* настроен как рабочий способ оповещений. Способ оповещений также необходимо связать с пользователями, указав конкретные адреса доставки (как это было сделано при настройке нового пользователя), иначе он не будет использоваться.

Новое действие

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

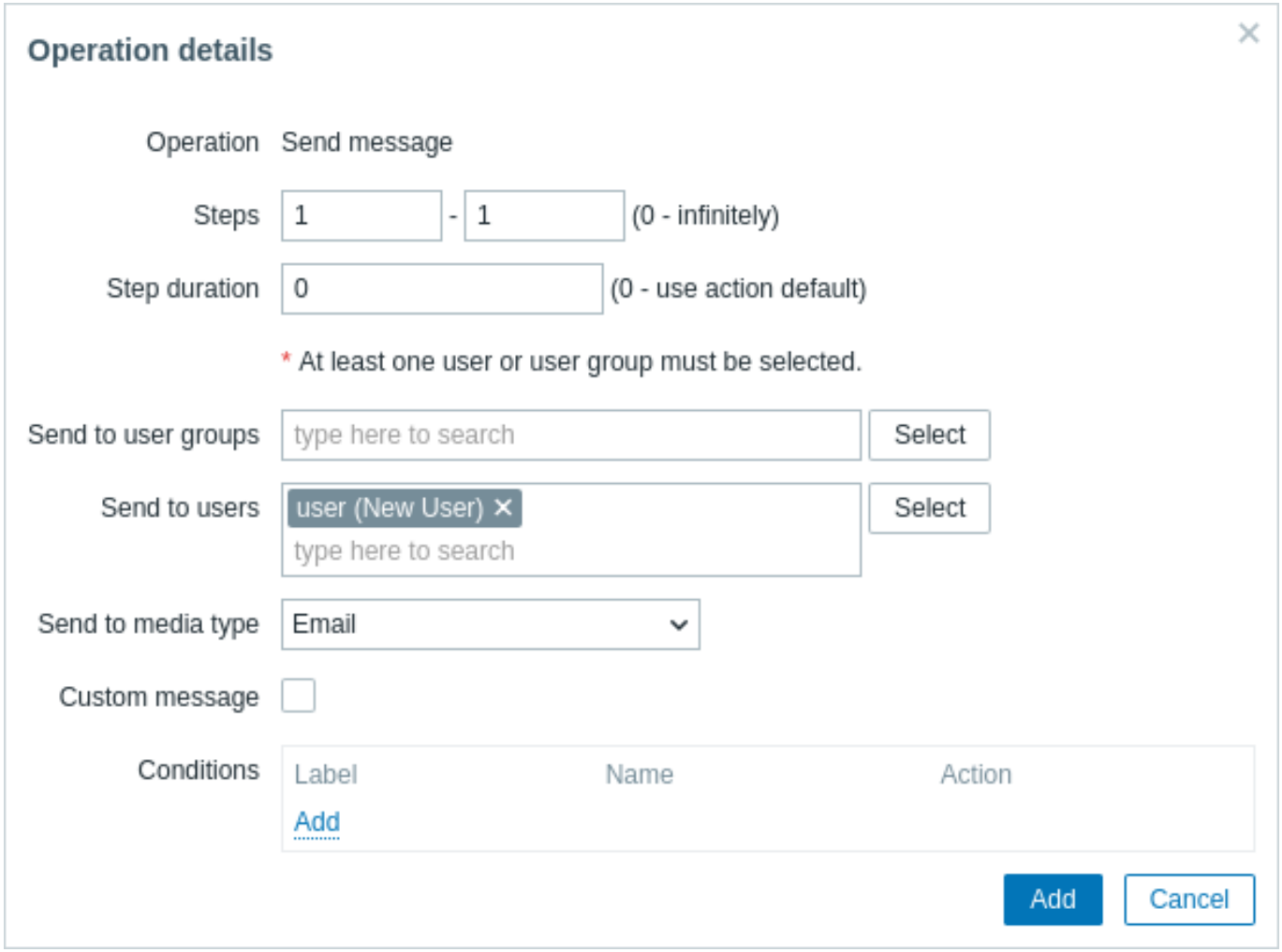


Все обязательные для заполнения поля отмечены красной звездочкой.

В этой форме введите имя действия.

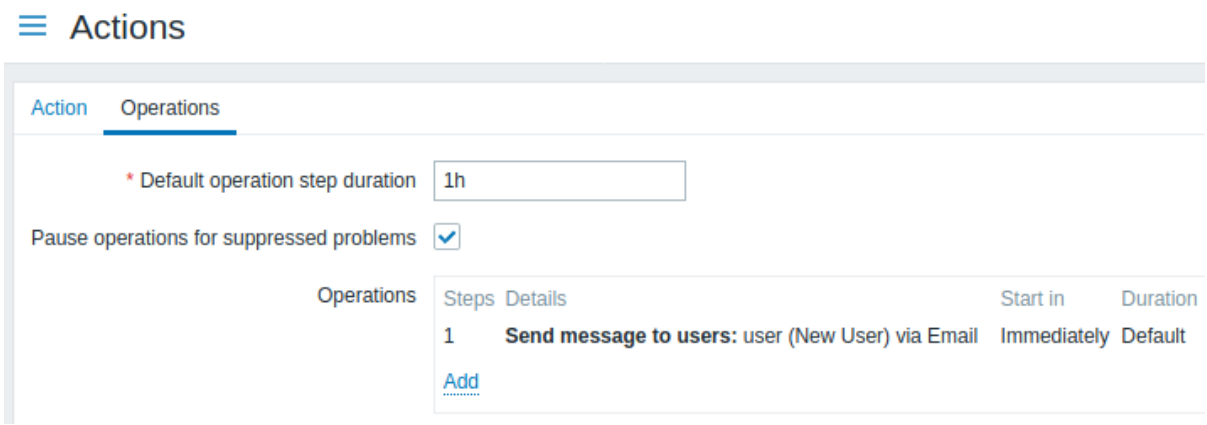
В самом простом случае, если мы не добавляем никаких более специфичных условий, действие будет выполняться при любом изменении триггера из состояния 'Ok' в состояние 'Проблема'.

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



Все обязательные для заполнения поля отмечены красной звездочкой.

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



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

Получение уведомления

Теперь, когда доставка уведомлений настроена, было бы неплохо действительно получить одно. Чтобы помочь с этим, мы можем намеренно увеличить нагрузку на наш узел сети — так, 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 таких 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 для узла сети, который сгенерировал событие, как указано на шаге *Добавление пользователя*. Пользователь, являясь 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 проверить журнал действий, перейдя в *Reports* > *Action log*.

6 Новый шаблон

Обзор

В этом разделе вы узнаете, как настроить шаблон.

Ранее мы узнали, как настроить элемент данных, триггер и как получать уведомление о проблеме для узла сети.

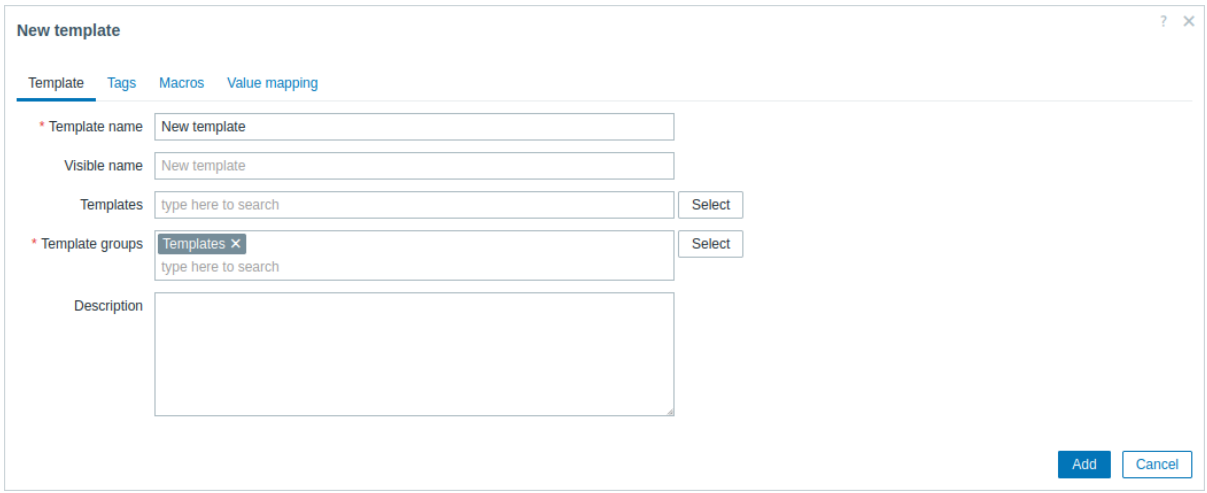
Хотя все эти шаги сами по себе обеспечивают большую гибкость, может показаться, что это слишком много действий, если нужно выполнить их, скажем, для тысячи узлов сети. Здесь была бы полезна некоторая автоматизация.

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

Когда шаблон привязывается к узлу сети, узел сети наследует все сущности шаблона. То есть, по сути, заранее подготовленный набор проверок можно применить очень быстро.

Добавление шаблона

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



Все обязательные поля ввода отмечены красной звёздочкой.

Здесь необходимо указать следующие параметры:

Имя шаблона

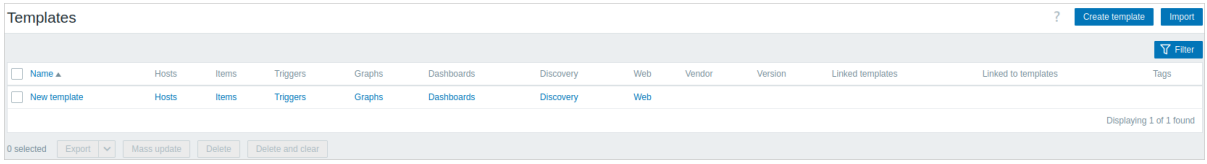
- Введите имя шаблона. Допускаются буквенно-цифровые символы, пробелы и символы подчёркивания.

Группы шаблонов

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

Note:
Права доступа к группам шаблонов назначаются при настройке группы пользователей на вкладке **Права доступа к шаблонам** так же, как и права доступа к узлам сети. Все права доступа назначаются группам, а не отдельным шаблонам, поэтому включение шаблона как минимум в одну группу является обязательным.

Когда всё будет готово, нажмите *Добавить*. Новый шаблон должен появиться в списке шаблонов. Вы также можете использовать фильтр, чтобы найти свой шаблон.



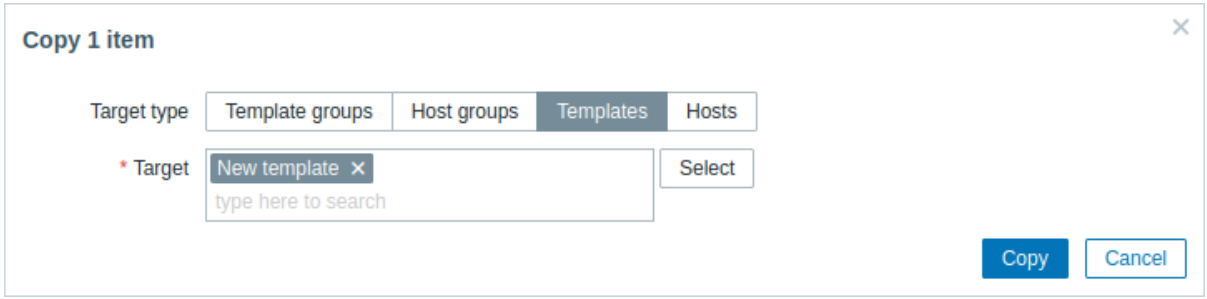
Как видите, шаблон присутствует в списке, но пока ничего не содержит — ни элементов данных, ни триггеров, ни других сущностей.

Добавление элемента данных в шаблон

Чтобы добавить элемент данных в шаблон, откройте список элементов данных для 'New host', перейдя в *Сбор данных* → *Узлы сети* и нажав *Элементы данных* рядом с 'New host'.

Затем:

- Установите флажок у элемента данных 'CPU Load' в списке.
- Нажмите *Копировать* под списком.
- Выберите вкладку *Шаблоны*.
- Выберите шаблон, в который нужно скопировать элемент данных.



Все обязательные поля ввода отмечены красной звёздочкой.

- Нажмите *Копировать*.

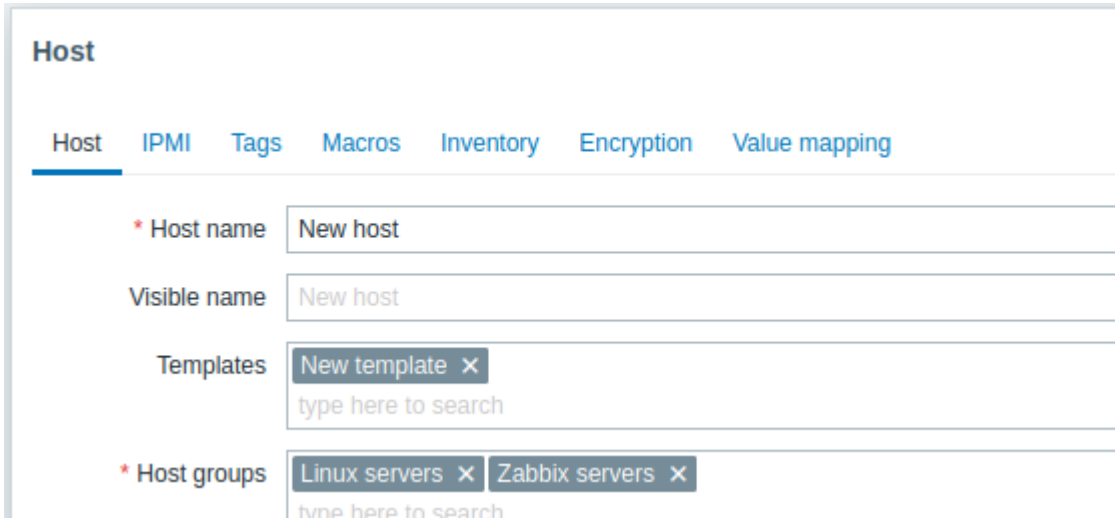
Если теперь перейти в *Сбор данных* > *Шаблоны*, в 'New template' должен появиться один новый элемент данных.

Пока мы ограничимся только одним элементом данных, но аналогичным образом вы можете добавить в шаблон любые другие элементы данных, триггеры или другие сущности, пока не получите достаточно полный набор сущностей для конкретной цели (мониторинг ОС, мониторинг отдельного приложения).

Привязка шаблона к узлу сети

Когда шаблон готов, остается только добавить его к узлу сети. Для этого перейдите в *Сбор данных* > *Узлы сети*, нажмите **Новый узел сети**, чтобы открыть форму его настройки, и найдите поле **Шаблоны**.

Начните вводить *New template* в поле *Шаблоны*. Имя созданного нами шаблона должно появиться в раскрывающемся списке. Прокрутите список вниз, чтобы выбрать его. Убедитесь, что он появился в поле *Шаблоны*.



Нажмите **Обновить** в форме, чтобы сохранить изменения. Теперь шаблон добавлен к узлу сети вместе со всеми сущностями, которые он содержит.

Таким образом его можно применять и к любому другому узлу сети. Любые изменения в элементах данных, триггерах и других сущностях на уровне шаблона будут распространяться на узлы сети, к которым привязан этот шаблон.

Привязка предопределенных шаблонов к узлам сети

Как вы, возможно, уже заметили, Zabbix поставляется с набором предопределенных шаблонов для различных ОС, устройств и приложений. Чтобы очень быстро приступить к мониторингу, вы можете привязать к узлу сети подходящий из них, однако имейте в виду, что эти шаблоны необходимо точно настроить под вашу среду. Некоторые проверки могут быть не нужны, а интервалы опроса могут оказаться слишком частыми.

Дополнительная информация о шаблонах доступна.

Мониторинг веб-сервера Apache

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

1. Убедитесь, что есть доступ к статистике Apache.
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 под учетной записью пользователя *Admin* по умолчанию.

Доступ к статистике Apache

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

В Apache информация о производительности и активности сервера в реальном времени предоставляется встроенным модулем `mod_status`. Когда он включен, создается доступная через веб страница (обычно `/server-status`), на которой отображаются данные о производительности и активности.

Чтобы проверить, включен ли `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, тогда необходимо включить модуль status.

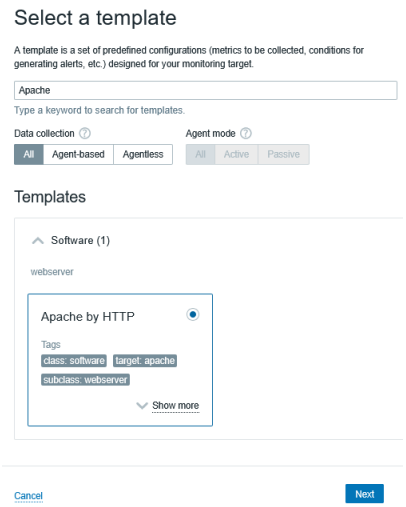
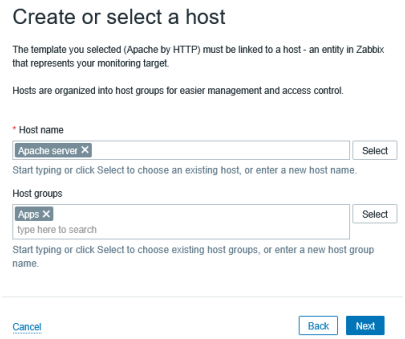
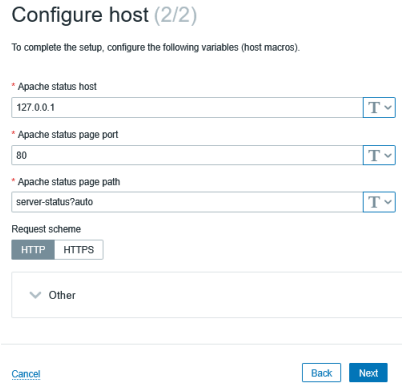
Применение шаблона мониторинга Apache

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

Чтобы применить шаблон, запустите Мастер узлов сети (в разделе *Сбор данных* > *Узлы сети*):

- **Выберите шаблон** ("Apache by HTTP")

- Создайте узел сети Apache и добавьте для него группу. Этот виртуальный узел сети будет содержать данные, связанные с Apache.
- Настройте узел сети. Важно: укажите адрес веб-сервера Apache (127.0.0.1 на локальной машине, IP-адрес на удалённой машине).

Шаг 1	Шаг 2	Шаг 3
		
Выбор шаблона.	Создание узла сети/группы узлов сети.	Настройка узла сети.

Остальные настройки можно оставить по умолчанию.

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

Просмотр последних данных



Latest data

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

Поздравляем, мониторинг веб-сервера Apache запущен!

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

Мониторинг Linux-сервера

Чтобы начать мониторинг Linux-сервера, выполните следующие шаги:

1. Примените шаблон мониторинга Linux.
2. Установите Zabbix агент на сервер.
3. Просмотрите последние данные.

Чтобы начать мониторинг удалённого Linux-сервера, необходимо знать его IP-адрес и IP-адрес Zabbix сервера.

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

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

Применение шаблона мониторинга Linux

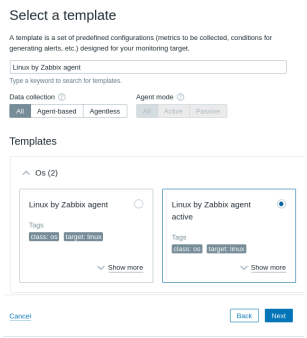
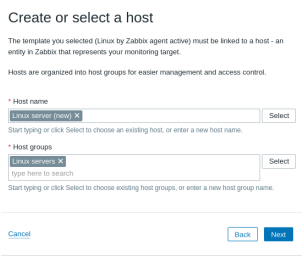
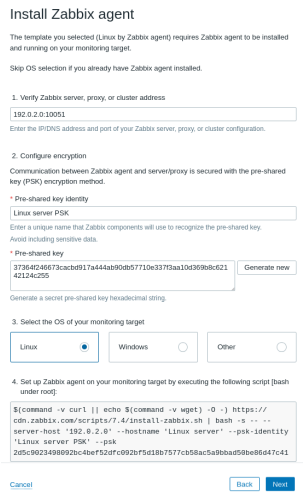
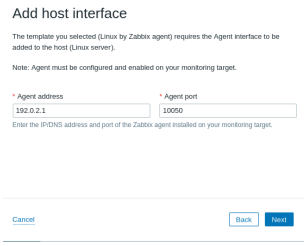
В Zabbix этот шаблон готов к использованию. Он содержит предварительно настроенные элементы данных, готовые для сбора данных, визуализации и анализа.

Чтобы применить шаблон, запустите Мастер узлов сети (в разделе *Сбор данных* > *Узлы сети*):

- **Выберите шаблон** (”Linux by Zabbix agent active”)
- Создайте узел сети для Linux-сервера, добавьте для него группу. Этот виртуальный узел сети будет содержать все данные, относящиеся к мониторингу Linux-сервера.
- 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 у вас уже установлен агент, вы можете пропустить этот шаг (нажмите *Далее*). Однако вам будет предложено указать идентификатор предварительно общего ключа и предварительно общий ключ для подключения к агенту.

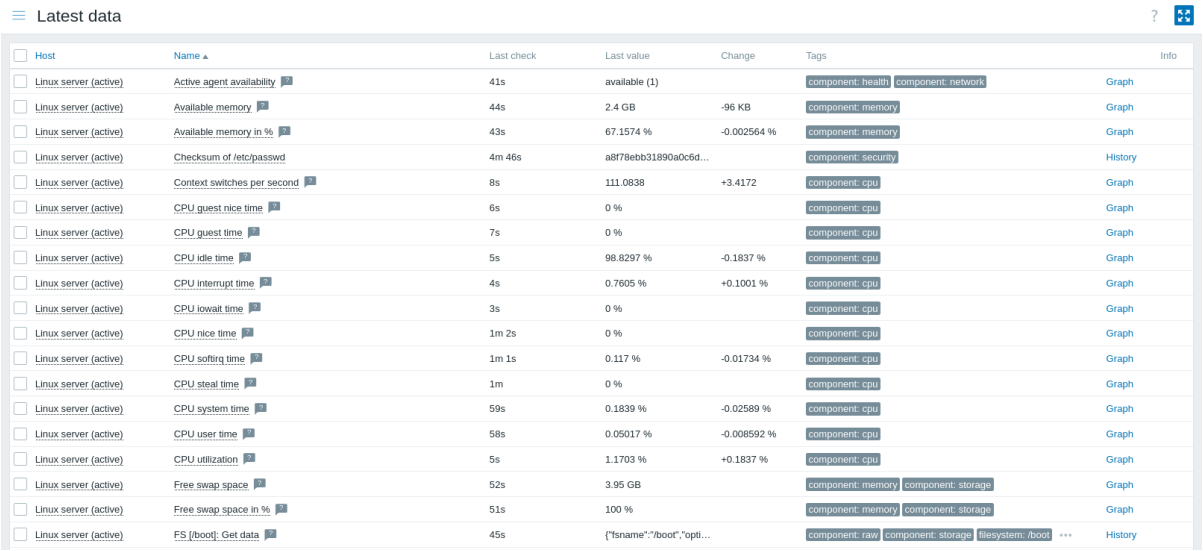
- Завершите настройку узла сети. Введите IP-адрес и порт агента (Linux-сервера).

Шаг 1	Шаг 2	Шаг 3	Шаг 4
			
Выбор шаблона.	Создание узла сети/группы узлов сети.	Настройка и установка агента на узле сети.	Добавление IP-адреса/порта Linux-сервера в качестве интерфейса агента.

Остальные настройки можно оставить по умолчанию.

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

Просмотр последних данных



Поздравляем, мониторинг Linux-сервера запущен!

Zabbix предоставляет возможности визуализации и отправляет оповещения при возникновении каких-либо проблем. Чтобы настроить оповещения по электронной почте, смотрите Получение уведомлений о проблемах.

Мониторинг сервера MySQL

Чтобы начать мониторинг сервера MySQL, выполните следующие шаги:

1. Создайте пользователя MySQL с ограниченными правами.
2. Примените шаблон мониторинга MySQL.
3. Просмотрите последние данные.

Чтобы начать мониторинг сервера MySQL, вам необходимо 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 воспользоваться этим руководством, у вас должны быть установлены сервер Zabbix и веб-интерфейс. Вы должны войти в Zabbix под учетной записью пользователя *Admin* по умолчанию.

Создание ограниченного пользователя MySQL

Рекомендуется создать пользователя MySQL с ограниченными правами, только необходимыми для мониторинга MySQL.

```
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 данных, готовые для сбора данных, визуализации и анализа.

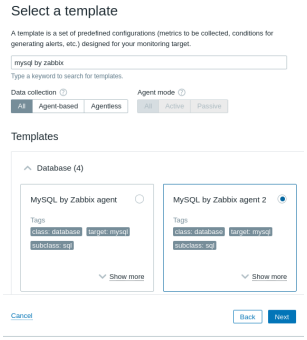
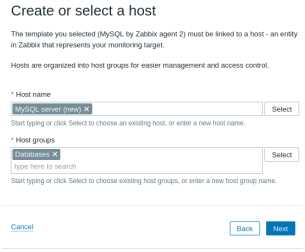
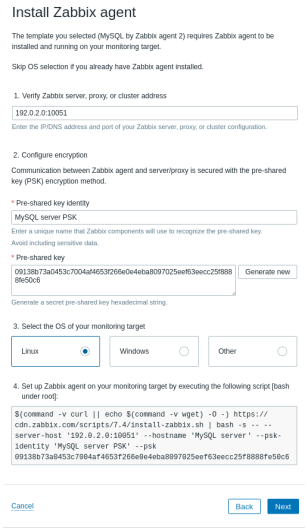
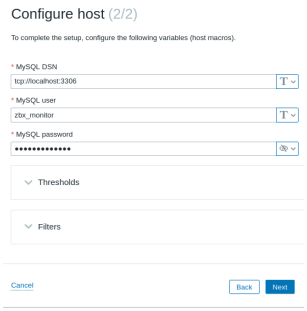
Чтобы применить шаблон, запустите Мастер узлов сети (в разделе *Сбор данных* > *Узлы сети*):

- **Выберите шаблон** (”MySQL by Zabbix agent 2”)
- Создайте узел сети сервера MySQL, добавьте для него группу. Этот виртуальный узел сети будет содержать все данные, относящиеся к мониторингу сервера MySQL.
- Настройте и установите агент 2 (Zabbix agent 2 поддерживает мониторинг MySQL из коробки).

Чтобы установить агент, выполните предоставленную команду на машине, где размещён сервер MySQL. Она установит агент и настроит подключение агента к серверу Zabbix с использованием шифрования.

Если у вас уже установлен агент 2, вы можете пропустить этот шаг (нажмите *Next*). Однако вам будет предложено указать идентификатор предварительно общего ключа и предварительно общий ключ для подключения к агенту.

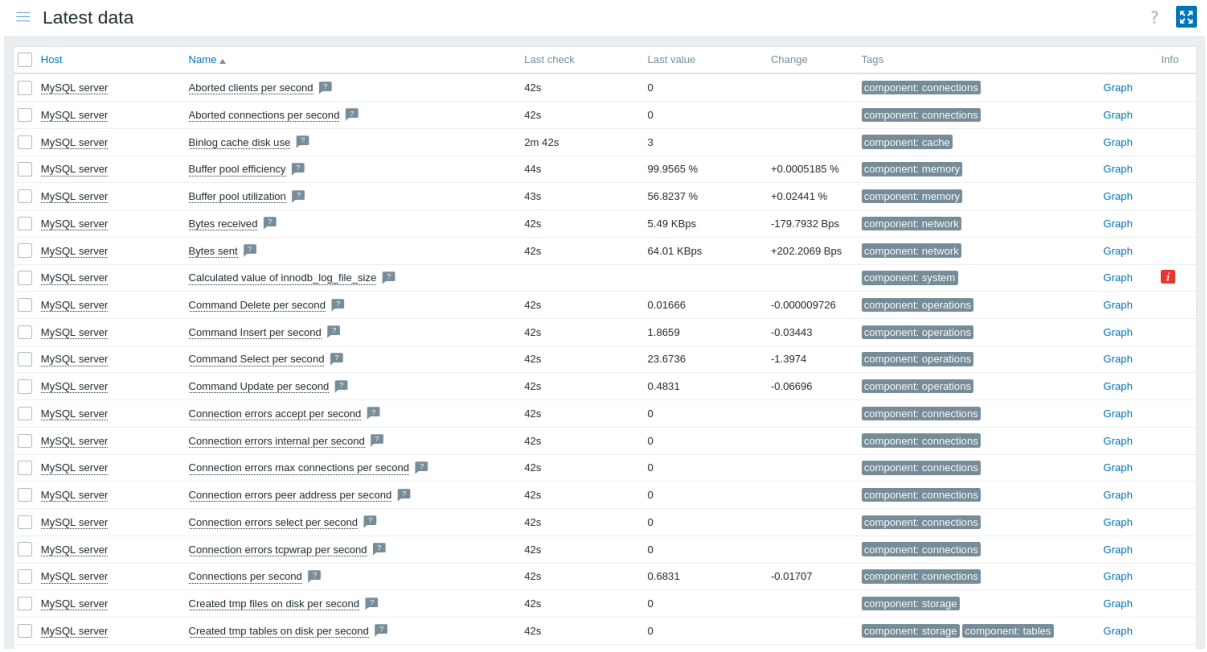
- Завершите настройку узла сети. Важно: вам будет предложено ввести DSN сервера MySQL, имя пользователя MySQL и пароль. Укажите пользователя `zbx_monitor` с паролем, который вы создали.

Шаг 1	Шаг 2	Шаг 3	Шаг 4
			
Выбор шаблона.	Создание узла сети/группы узлов сети.	Настройка и установка агента на узле сети.	Добавление DNS, имени пользователя и пароля сервера MySQL.

Остальные настройки можно оставить по умолчанию.

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

Просмотр последних данных



Latest data

Host	Name	Last check	Last value	Change	Tags		Info
MySQL server	Aborted clients per second	42s	0		component: connections	Graph	
MySQL server	Aborted connections per second	42s	0		component: connections	Graph	
MySQL server	Binlog cache disk use	2m 42s	3		component: cache	Graph	
MySQL server	Buffer pool efficiency	44s	99.9565 %	+0.0005185 %	component: memory	Graph	
MySQL server	Buffer pool utilization	43s	56.8237 %	+0.02441 %	component: memory	Graph	
MySQL server	Bytes received	42s	5.49 KBps	-179.7932 Bps	component: network	Graph	
MySQL server	Bytes sent	42s	64.01 KBps	+202.2069 Bps	component: network	Graph	
MySQL server	Calculated value of innodb_log_file_size				component: system	Graph	
MySQL server	Command Delete per second	42s	0.01666	-0.000009726	component: operations	Graph	
MySQL server	Command Insert per second	42s	1.8659	-0.03443	component: operations	Graph	
MySQL server	Command Select per second	42s	23.6736	-1.3974	component: operations	Graph	
MySQL server	Command Update per second	42s	0.4831	-0.06696	component: operations	Graph	
MySQL server	Connection errors accept per second	42s	0		component: connections	Graph	
MySQL server	Connection errors internal per second	42s	0		component: connections	Graph	
MySQL server	Connection errors max connections per second	42s	0		component: connections	Graph	
MySQL server	Connection errors peer address per second	42s	0		component: connections	Graph	
MySQL server	Connection errors select per second	42s	0		component: connections	Graph	
MySQL server	Connection errors tcpwrap per second	42s	0		component: connections	Graph	
MySQL server	Connections per second	42s	0.6831	-0.01707	component: connections	Graph	
MySQL server	Created tmp files on disk per second	42s	0		component: storage	Graph	
MySQL server	Created tmp tables on disk per second	42s	0		component: storage component: tables	Graph	

Поздравляем, мониторинг сервера MySQL запущен!

Zabbix предоставляет возможности визуализации и отправляет оповещения при возникновении каких-либо проблем. Чтобы настроить оповещения по электронной почте, смотрите Получение уведомлений о проблемах.

3 Установка: дополнительные руководства

Если вы ищете самый простой способ установки Zabbix, пожалуйста, смотрите Установка и первые шаги.

Приведенные ниже руководства охватывают **дополнительные** или специфичные для системы способы установки Zabbix или его компонентов:

- Установка из контейнеров

- Установка из исходного кода
- Примечания по установке из пакетов
- Установка агента Zabbix из Windows MSI
- Сборка агента Zabbix 2 в Windows
- Сборка агента Zabbix в Windows
- Установка агента в macOS из PKG
- Сборка агента Zabbix в macOS
- Устройство Zabbix

Установка из контейнеров

Обзор

В этом разделе описывается, как развернуть Zabbix с помощью Docker или Docker Compose.

Zabbix официально предоставляет:

- Отдельные Docker-образы для каждого компонента Zabbix, чтобы запускать их как переносимые и самодостаточные контейнеры.
- Файлы Compose для определения и запуска многоконтейнерных компонентов Zabbix в Docker.

Attention:
Начиная с Zabbix 6.0, детерминированные триггеры необходимо создавать во время установки. Если для MySQL/MariaDB включено бинарное логирование, для этого требуются привилегии суперпользователя или установка переменной/параметра конфигурации *log_bin_trust_function_creators = 1*. Инструкции по установке этой переменной см. в разделе Скрипты создания базы данных.
Обратите внимание, что при выполнении из консоли переменная будет установлена только временно и будет сброшена после перезапуска Docker. В этом случае оставьте SQL-службу работающей, остановите только службу zabbix-server, выполнив 'docker compose down zabbix-server', а затем 'docker compose up -d zabbix-server'.
В качестве альтернативы можно установить эту переменную в файле конфигурации.

Исходные файлы

Исходные файлы Dockerfile хранятся в официальном репозитории Zabbix на GitHub, где вы можете отслеживать последние изменения файлов или создать форк проекта, чтобы собирать собственные образы.

Docker

Zabbix предоставляет образы на основе различных базовых образов ОС. Чтобы получить список поддерживаемых образов базовых операционных систем для конкретного компонента Zabbix, смотрите описание компонента в Docker Hub. Все образы Zabbix настроены на пересборку последних образов при обновлении базовых образов.

Установка

Чтобы получить образ компонента Zabbix, выполните:

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

Замените `zabbix/zabbix-server-mysql` именем требуемого docker-репозитория.

Эта команда загрузит последнюю стабильную версию компонента Zabbix на основе ОС Alpine Linux. Вы можете добавить теги к имени репозитория, чтобы получить образ на основе другой операционной системы или конкретной основной либо промежуточной версии Zabbix.

В Docker Hub доступны следующие репозитории:

Компонент	Docker-репозиторий
Агент Zabbix	zabbix/zabbix-agent
Сервер Zabbix	
с поддержкой MySQL	zabbix/zabbix-server-mysql
с поддержкой PostgreSQL	zabbix/zabbix-server-pgsql

Компонент		Docker-репозиторий
Веб-интерфейс Zabbix		
	на основе веб-сервера Apache2 с поддержкой MySQL	zabbix/zabbix-web-apache-mysql
	на основе веб-сервера Apache2 с поддержкой PostgreSQL	zabbix/zabbix-web-apache-pgsql
	на основе веб-сервера Nginx с поддержкой MySQL	zabbix/zabbix-web-nginx-mysql
	на основе веб-сервера Nginx с поддержкой PostgreSQL	zabbix/zabbix-web-nginx-pgsql
Прокси Zabbix		
	с поддержкой SQLite3	zabbix/zabbix-proxy-sqlite3
	с поддержкой MySQL	zabbix/zabbix-proxy-mysql
Zabbix Java gateway		zabbix/zabbix-java-gateway

Note:
Поддержка SNMP trap предоставляется в отдельном репозитории zabbix/zabbix-snmptraps. Его можно связать с сервером Zabbix и прокси Zabbix.

Теги

Официальные образы компонентов Zabbix могут содержать следующие теги:

Tag	Description	Example
latest	Последняя стабильная версия компонента Zabbix на основе образа Alpine Linux.	zabbix-agent:latest
<OS>-trunk	Последняя nightly-сборка версии Zabbix, которая в настоящее время разрабатывается для конкретной операционной системы. **<OS>** — базовая операционная система. Поддерживаемые значения: *alpine* — Alpine Linux; *ltsc2019* — Windows 10 LTSC 2019 (только агент); *ol* — Oracle Linux; *ltsc2022* — Windows 11 LTSC 2022 (только агент); *ubuntu* — Ubuntu	zabbix-agent:ubuntu-trunk
<OS>-latest	Последняя стабильная версия компонента Zabbix для конкретной операционной системы. **<OS>** — базовая операционная система. Поддерживаемые значения: *alpine* — Alpine Linux; *ltsc2019* — Windows 10 LTSC 2019 (только агент); *ol* — Oracle Linux; *ltsc2022* — Windows 11 LTSC 2022 (только агент); *ubuntu* — Ubuntu	zabbix-agent:ol-latest

Tag	Description	Example
<OS>-X.X-latest	Последняя минорная версия компонента Zabbix для указанной мажорной версии и операционной системы. **<OS>** — базовая операционная система. Поддерживаемые значения: *alpine* — Alpine Linux; *ltsc2019* — Windows 10 LTSC 2019 (только агент); *ol* — Oracle Linux; *ltsc2022* — Windows 11 LTSC 2022 (только агент); *ubuntu* — Ubuntu **X.X** — мажорная версия Zabbix (например, *6.0*, *7.4*, *8.0*).	zabbix-agent:alpine-8.0-latest
<OS>-X.X.*	Конкретная минорная версия компонента Zabbix для указанной мажорной версии и операционной системы. **<OS>** — базовая операционная система. Поддерживаемые значения: *alpine* — Alpine Linux; *ltsc2019* — Windows 10 LTSC 2019 (только агент); *ol* — Oracle Linux; *ltsc2022* — Windows 11 LTSC 2022 (только агент); *ubuntu* — Ubuntu **X.X** — мажорная версия Zabbix (например, *6.0*, *7.4*, *8.0*). * — минорная версия Zabbix	zabbix-agent:alpine-8.0.1

Начальная настройка

После загрузки образов запустите контейнеры, выполнив команду `docker run` с дополнительными аргументами для указания необходимых переменных окружения и/или точек монтирования. Ниже приведены некоторые примеры настройки.

Note:
Чтобы включить взаимодействие между компонентами Zabbix, некоторые порты, такие как 10051/TCP для сервера Zabbix (trapper), 10050/TCP для агента Zabbix, 162/UDP для SNMP-ловушек и 80/TCP для веб-интерфейса Zabbix, будут открыты на хост-машине. Полный список портов по умолчанию, используемых компонентами Zabbix, доступен на странице Требования. Для сервера Zabbix и агента порт по умолчанию можно изменить, задав переменную окружения ZBX_LISTENPORT environment variable.

Переменные окружения

Все образы компонентов Zabbix предоставляют переменные окружения для управления конфигурацией. Поддерживаемые переменные окружения перечислены в репозитории компонентов.

Эти переменные окружения являются параметрами из файлов конфигурации Zabbix, но используют другой способ именования. Например, `ZBX_LOGSLOWQUERIES` эквивалентен `LogSlowQueries` из файлов конфигурации Zabbix сервера или Zabbix прокси.

Attention:
Некоторые параметры конфигурации (например, `PIDFile` и `LogType`) не могут быть изменены.

Следующие переменные окружения относятся только к компонентам Docker и отсутствуют в файлах конфигурации Zabbix:

Переменная	Компоненты	Значение по умолчанию	Описание
`DB_SERVER_HOST`	Сервер Прокси Веб-интерфейс	`mysql-server` для MYSQL `postgres-server` для PostgreSQL	IP-адрес или DNS-имя сервера MySQL или PostgreSQL.

Переменная	Компоненты	Значение по умолчанию	Описание
DB_SERVER_PORT	Сервер Прокси Веб-интерфейс	3306 для MYSQL 5432 для PostgreSQL	Порт сервера MySQL или PostgreSQL.
MYSQL_USER	Сервер Прокси Веб-интерфейс	zabbix	Пользователь базы данных MySQL.
MYSQL_PASSWORD	Сервер Прокси Веб-интерфейс	zabbix	Пароль базы данных MySQL.
MYSQL_DATABASE	Сервер Прокси Веб-интерфейс	zabbix для сервера Zabbix zabbix_proxy для прокси Zabbix	Имя базы данных Zabbix.
POSTGRES_USER	Сервер Веб-интерфейс	zabbix	Пользователь базы данных PostgreSQL.
POSTGRES_PASSWORD	Сервер Веб-интерфейс	zabbix	Пароль базы данных PostgreSQL.
POSTGRES_DB	Сервер Веб-интерфейс	zabbix для сервера Zabbix zabbix_proxy для прокси Zabbix	Имя базы данных Zabbix.
PHP_TZ	Веб-интерфейс	Europe/Riga	Часовой пояс в формате PHP. Полный список поддерживаемых часовых поясов доступен на php.net.
ZBX_SERVER_NAME	Веб-интерфейс	Zabbix Docker	Отображаемое имя установки Zabbix под логотипом Zabbix в вертикальном меню веб-интерфейса.
ZBX_JAVAGATEWAY_ENABLE	Сервер Прокси	false	Включает взаимодействие с Zabbix Java gateway для сбора проверок, связанных с Java.
ZBX_ENABLE_SNMP_TRAPS	Сервер Прокси	false	Включает функцию SNMP trap. Для этого требуется экземпляр **zabbix-snmptraps** и общий том */var/lib/zabbix/snmptraps* для сервера Zabbix или прокси Zabbix.

Тома

Образы позволяют монтировать тома, используя следующие точки монтирования:

Том	Описание
Zabbix агент	
/etc/zabbix/zabbix_agentd.d	Том позволяет включать файлы **.conf* и расширять Zabbix агент с помощью функции UserParameter
/var/lib/zabbix/modules	Том позволяет загружать дополнительные модули и расширять Zabbix агент с помощью функции LoadModule
/var/lib/zabbix/enc	Том используется для хранения файлов, связанных с TLS. Имена этих файлов задаются с помощью переменных окружения ZBX_TLSCAFILE, ZBX_TLSCRLFILE, ZBX_TLSKEY_FILE и ZBX_TLSPSKFILE
Zabbix сервер	
/usr/lib/zabbix/alertscripts	Том используется для пользовательских скриптов оповещений. Это параметр AlertScriptsPath в zabbix_server.conf
/usr/lib/zabbix/externalscripts	Том используется внешними проверками. Это параметр ExternalScripts в zabbix_server.conf
/var/lib/zabbix/modules	Том позволяет загружать дополнительные модули и расширять Zabbix сервер с помощью функции LoadModule
/var/lib/zabbix/enc	Том используется для хранения файлов, связанных с TLS. Имена этих файлов задаются с помощью переменных окружения ZBX_TLSCAFILE, ZBX_TLSCRLFILE, ZBX_TLSKEY_FILE и ZBX_TLSPSKFILE
/var/lib/zabbix/ssl/certs	Том используется как расположение файлов клиентских SSL-сертификатов для аутентификации клиента. Это параметр SSLCertLocation в zabbix_server.conf

Том	Описание
Zabbix прокси	
/var/lib/zabbix/ssl/keys	Том используется как расположение файлов закрытых SSL-ключей для аутентификации клиента. Это параметр `SSLKeyLocation` в zabbix_server.conf
/var/lib/zabbix/ssl/ssl_ca	Том используется как расположение файлов центра сертификации (CA) для проверки SSL-сертификата сервера. Это параметр `SSLCALocation` в zabbix_server.conf
/var/lib/zabbix/snmptraps	Том используется как расположение файла snmptraps.log. Он может совместно использоваться контейнером zabbix-snmptraps и наследоваться с помощью опции Docker volumes_from при создании нового экземпляра Zabbix сервера. Возможность обработки SNMP trap может быть включена с помощью общего тома и переключения переменной окружения `ZBX_ENABLE_SNMP_TRAPS` в значение 'true'
/var/lib/zabbix/mibs	Том позволяет добавлять новые файлы MIB. Подкаталоги не поддерживаются, все MIB должны быть размещены в `/var/lib/zabbix/mibs`
Zabbix прокси	
/usr/lib/zabbix/externalscripts	Том используется внешними проверками. Это параметр `ExternalScripts` в zabbix_proxy.conf
/var/lib/zabbix/db_data	Том позволяет хранить файлы базы данных на внешних устройствах. Поддерживается только для Zabbix прокси с SQLite3
/var/lib/zabbix/modules	Том позволяет загружать дополнительные модули и расширять Zabbix сервер с помощью функции LoadModule
/var/lib/zabbix/enc	Том используется для хранения файлов, связанных с TLS. Имена этих файлов задаются с помощью переменных окружения `ZBX_TLSCAFILE`, `ZBX_TLSCRLFILE`, `ZBX_TLSKEY_FILE` и `ZBX_TLSPSKFILE`
/var/lib/zabbix/ssl/certs	Том используется как расположение файлов клиентских SSL-сертификатов для аутентификации клиента. Это параметр `SSLCertLocation` в zabbix_proxy.conf
/var/lib/zabbix/ssl/keys	Том используется как расположение файлов закрытых SSL-ключей для аутентификации клиента. Это параметр `SSLKeyLocation` в zabbix_proxy.conf
/var/lib/zabbix/ssl/ssl_ca	Том используется как расположение файлов центра сертификации (CA) для проверки SSL-сертификата сервера. Это параметр `SSLCALocation` в zabbix_proxy.conf
/var/lib/zabbix/snmptraps	Том используется как расположение файла snmptraps.log. Он может совместно использоваться контейнером zabbix-snmptraps и наследоваться с помощью опции Docker volumes_from при создании нового экземпляра Zabbix сервера. Возможность обработки SNMP trap может быть включена с помощью общего тома и переключения переменной окружения `ZBX_ENABLE_SNMP_TRAPS` в значение 'true'
/var/lib/zabbix/mibs	Том позволяет добавлять новые файлы MIB. Подкаталоги не поддерживаются, все MIB должны быть размещены в `/var/lib/zabbix/mibs`
Веб-интерфейс Zabbix на базе веб-сервера Apache2	
/etc/ssl/apache2	Том позволяет включить HTTPS для веб-интерфейса Zabbix. Том должен содержать два файла `ssl.crt` и `ssl.key`, подготовленные для SSL-соединений Apache2
Веб-интерфейс Zabbix на базе веб-сервера Nginx	
/etc/ssl/nginx	Том позволяет включить HTTPS для веб-интерфейса Zabbix. Том должен содержать два файла `ssl.crt`, `ssl.key` и `dhparam.pem`, подготовленные для SSL-соединений Nginx
Zabbix snmptraps	

Том	Описание
/var/lib/zabbix/snmptraps	Том содержит файл журнала `snmptraps.log` с полученными SNMP trap
/var/lib/zabbix/mibs	Том позволяет добавлять новые файлы MIB. Подкаталоги не поддерживаются, все MIB должны быть размещены в `/var/lib/zabbix/mibs`

Для получения дополнительной информации см. официальные репозитории Zabbix в Docker Hub.

Примеры

Пример 1

В этом примере показано, как запустить сервер Zabbix с поддержкой базы данных MySQL, веб-интерфейс Zabbix на базе веб-сервера Nginx и Zabbix Java gateway.

1. Создайте сеть, выделенную для контейнеров компонентов Zabbix:

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

2. Запустите пустой экземпляр сервера MySQL:

```
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=utf8 --collation-server=utf8_bin \
    --default-authentication-plugin=caching_sha2_password
```

3. Запустите экземпляр Zabbix Java gateway:

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

4. Запустите экземпляр сервера Zabbix и свяжите его с созданным экземпляром 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 MYSQL_ROOT_PASSWORD="root_pwd" \
    -e ZBX_JAVAGATEWAY="zabbix-java-gateway" \
    --network=zabbix-net \
    -p 10051:10051 \
    --restart unless-stopped \
    -d zabbix/zabbix-server-mysql:alpine-8.0-latest
```

5. Запустите веб-интерфейс Zabbix и свяжите его с созданными экземплярами сервера MySQL и сервера Zabbix:

```
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 MYSQL_ROOT_PASSWORD="root_pwd" \
    --network=zabbix-net \
    -p 80:8080 \
    --restart unless-stopped \
    -d zabbix/zabbix-web-nginx-mysql:alpine-8.0-latest
```

Пример 2

В этом примере показано, как запустить сервер Zabbix с поддержкой базы данных PostgreSQL, веб-интерфейс Zabbix на базе веб-сервера Nginx и функцией SNMP trap.

1. Создайте сеть, выделенную для контейнеров компонентов Zabbix:

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

2. Запустите пустой экземпляр сервера PostgreSQL:

```
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 экземпляр Zabbix snmptraps:

```
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 экземпляр сервера Zabbix и свяжите его с созданным экземпляром сервера PostgreSQL:

```
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 \
    -d zabbix/zabbix-server-pgsql:alpine-8.0-latest
```

5. Запустите веб-интерфейс Zabbix и свяжите его с созданными экземплярами сервера PostgreSQL и сервера Zabbix:

```
docker run --name zabbix-web-nginx-pgsql -t \
    -e ZBX_SERVER_HOST="zabbix-server-pgsql" \
    -e DB_SERVER_HOST="postgres-server" \
    -e POSTGRES_USER="zabbix" \
    -e POSTGRES_PASSWORD="zabbix_pwd" \
    -e POSTGRES_DB="zabbix" \
    --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
```

Пример 3

В этом примере показано, как запустить сервер Zabbix с поддержкой базы данных MySQL, веб-интерфейс Zabbix на базе веб-сервера Nginx и Zabbix Java gateway с использованием `podman` в Red Hat 8.

1. Создайте новый pod с именем `zabbix` и открытыми портами (веб-интерфейс, trapper сервера Zabbix):

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

2. (необязательно) Запустите контейнер агента Zabbix в расположении 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-74:latest
```

3. Создайте каталог `./mysql/` на узле сети и запустите сервер Oracle MySQL 8.4:

```
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=utf8 --collation-server=utf8_bin \
    --default-authentication-plugin=caching_sha2_password
```

4. Запустите контейнер сервера Zabbix:

```
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 MYSQL_ROOT_PASSWORD="root_pwd" \
                -e ZBX_JAVAGATEWAY="127.0.0.1" \
                --restart=always \
                --pod=zabbix \
                -d registry.connect.redhat.com/zabbix/zabbix-server-mysql-74
```

5. Запустите контейнер Zabbix Java Gateway:

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

6. Запустите контейнер веб-интерфейса Zabbix:

```
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 MYSQL_ROOT_PASSWORD="root_pwd" \
                --restart=always \
                --pod=zabbix \
                -d registry.connect.redhat.com/zabbix/zabbix-web-mysql-74
```

Note:
Pod `zabbix` предоставляет узлу сети порт 80/TCP (HTTP) из порта 8080/TCP контейнера `zabbix-web-mysql`.

Docker Compose

В качестве альтернативы Zabbix можно установить с помощью плагина Docker Compose. Файлы Compose для определения и запуска многоконтейнерных компонентов Zabbix доступны в официальном репозитории Zabbix Docker на GitHub.

Attention:
Официальные compose-файлы Zabbix поддерживают Docker Compose версии 3.

Эти compose-файлы добавлены в качестве примеров; они перегружены. Например, они содержат прокси с поддержкой как MySQL, так и SQLite3.

Чтобы получить Docker compose-файлы, предоставляемые Zabbix, клонируйте репозиторий:

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

Переключитесь на требуемую версию:

```
git checkout 8.0
```

Настройте compose-файлы и создайте и запустите контейнеры:

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

Замените `docker-compose_v3_alpine_mysql_latest.yaml` в приведенной выше команде на требуемый файл конфигурации.

Доступны следующие варианты:

Имя файла	Описание
`docker-compose_v3_alpine_mysql_latest.yaml`	Compose-файл загружает последнюю версию компонентов Zabbix 8.0 на Alpine Linux с поддержкой базы данных MySQL.
`docker-compose_v3_alpine_mysql_local.yaml`	Compose-файл локально собирает последнюю версию Zabbix 8.0 и запускает компоненты Zabbix на Alpine Linux с поддержкой базы данных MySQL.
`docker-compose_v3_alpine_pgsql_latest.yaml`	Compose-файл загружает последнюю версию компонентов Zabbix 8.0 на Alpine Linux с поддержкой базы данных PostgreSQL.
`docker-compose_v3_alpine_pgsql_local.yaml`	Compose-файл локально собирает последнюю версию Zabbix 8.0 и запускает компоненты Zabbix на Alpine Linux с поддержкой базы данных PostgreSQL.
`docker-compose_v3_ol_mysql_latest.yaml`	Compose-файл загружает последнюю версию компонентов Zabbix 8.0 на Oracle Linux с поддержкой базы данных MySQL.
`docker-compose_v3_ol_mysql_local.yaml`	Compose-файл локально собирает последнюю версию Zabbix 8.0 и запускает компоненты Zabbix на Oracle Linux с поддержкой базы данных MySQL.
`docker-compose_v3_ol_pgsql_latest.yaml`	Compose-файл загружает последнюю версию компонентов Zabbix 8.0 на Oracle Linux с поддержкой базы данных PostgreSQL.
`docker-compose_v3_ol_pgsql_local.yaml`	Compose-файл локально собирает последнюю версию Zabbix 8.0 и запускает компоненты Zabbix на Oracle Linux с поддержкой базы данных PostgreSQL.
`docker-compose_v3_ubuntu_mysql_latest.yaml`	Compose-файл загружает последнюю версию компонентов Zabbix 8.0 на Ubuntu 24.04 (noble) с поддержкой базы данных MySQL.
`docker-compose_v3_ubuntu_mysql_local.yaml`	Compose-файл локально собирает последнюю версию Zabbix 8.0 и запускает компоненты Zabbix на Ubuntu 24.04 (noble) с поддержкой базы данных MySQL.
`docker-compose_v3_ubuntu_pgsql_latest.yaml`	Compose-файл загружает последнюю версию компонентов Zabbix 8.0 на Ubuntu 24.04 (noble) с поддержкой базы данных PostgreSQL.
`docker-compose_v3_ubuntu_pgsql_local.yaml`	Compose-файл локально собирает последнюю версию Zabbix 8.0 и запускает компоненты Zabbix на Ubuntu 24.04 (noble) с поддержкой базы данных PostgreSQL.

Хранилище

Файлы Compose настроены для поддержки локального хранения на машине узла сети. Docker Compose создаст каталог `zbx_env` в папке с compose-файлом, когда вы запустите компоненты Zabbix с использованием compose-файла. Каталог будет содержать ту же структуру, что описана в разделе Volumes, а также каталог для хранения базы данных.

Также имеются тома в режиме только для чтения для файлов `/etc/localtime` и `/etc/timezone`.

Переменные окружения

Файлы переменных имеют следующую структуру именования: `.env_<type of component>` и расположены в каталоге *env_vars*. Подробные сведения о правилах именования переменных и доступных вариантах выбора см. в разделе переменные окружения.

Примеры

Пример 1

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

Команда загрузит последние образы Zabbix 8.0 для каждого компонента Zabbix и запустит их в фоновом режиме.

> **Attention:**
> Не забудьте загрузить файлы `.env_<type of component>` из официального репозитория Zabbix на github.com вместе с compose-файлами.

Пример 2

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

Команда загрузит базовый образ Ubuntu 24.04 (noble), затем локально соберёт компоненты Zabbix 8.0 и запустит их в фоновом режиме.

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 Zabbix сервер, Zabbix прокси или Zabbix агент требуется C99 с расширениями GNU. Эту версию можно явно указать, задав CFLAGS=”-std=gnu99”:

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

Note:
Если установка выполняется из репозитория Zabbix Git, необходимо сначала выполнить:

```
./bootstrap.sh
```

При настройке исходного кода для Zabbix сервер или прокси необходимо указать тип базы данных, который будет использоваться. Одновременно с процессом сервера или прокси может быть скомпилирован только один тип базы данных.

Чтобы увидеть все поддерживаемые параметры конфигурации, в распакованном каталоге с исходным кодом Zabbix выполните:

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

Чтобы настроить исходный код для Zabbix сервер и агент, можно выполнить примерно такую команду:

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

Чтобы настроить исходный код для Zabbix сервер (с PostgreSQL и т. д.), можно выполнить:

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

Чтобы настроить исходный код для Zabbix прокси (с SQLite и т. д.), можно выполнить:

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

Чтобы настроить исходный код для Zabbix агент, можно выполнить:

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

или, для Zabbix agent 2:

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

Note:
Для сборки Zabbix agent 2 должна быть установлена поддерживаемая версия Go.

Примечания по параметрам компиляции:

- `--enable-agent` — компилирует Zabbix агент, а также утилиты командной строки Zabbix get и Zabbix sender.
- `--with-libcurl` — требуется для мониторинга виртуальных машин, аутентификации SMTP и элементов данных Zabbix агент `web.page.*`. 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 конкретной клиентской библиотеки MySQL. Полезно, если установлено несколько версий MySQL или MariaDB.
- `--enable-static` — статически компонует библиотеки (не поддерживается в Solaris). Используйте этот параметр, если планируете распространять скомпилированные двоичные файлы в системы без необходимых библиотек. Не рекомендуется при сборке Zabbix сервер. Для статической сборки сервера требуется статическая версия каждой внешней библиотеки. Скрипт configure не проверяет это автоматически.

- `--with-stacksize=<value>` — задаёт размер стека на поток в килобайтах (например, `--with-stacksize=512`). Вы можете увеличить это значение, если Zabbix аварийно завершается или зависает из-за переполнения стека (например, во время предобработки в системах с низкими значениями ограничения размера стека потока по умолчанию).

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 `config.log` показывает фактическую причину:
```
/usr/bin/ld: cannot find -lssl
/usr/bin/ld: cannot find -lcrypto
```

См. также:

- Сборка Zabbix с поддержкой шифрования
- Известные проблемы компиляции

5 Сборка и установка всего

Note:
Если установка выполняется из репозитория Zabbix Git, необходимо сначала выполнить:

```
$ make dbschema
```

`make install`

Этот шаг должен выполняться пользователем с достаточными правами (обычно 'root' или с использованием sudo).

При выполнении `make install` по умолчанию будут установлены двоичные файлы демонов (zabbix_server, zabbix_agentd, zabbix_proxy) в /usr/local/sbin, а двоичные файлы клиента (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 демонов будут установлены в <prefix>/sbin, а утилиты — в <prefix>/bin. Man-страницы будут установлены в <prefix>/share.

6 Просмотрите и отредактируйте файлы конфигурации

- отредактируйте файл конфигурации агента Zabbix **/usr/local/etc/zabbix_agentd.conf**

Вам необходимо настроить этот файл для каждого узла сети, на котором установлен zabbix_agentd.

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

- отредактируйте файл конфигурации сервера Zabbix **/usr/local/etc/zabbix_server.conf**

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

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

- если вы установили прокси Zabbix, отредактируйте файл конфигурации прокси **/usr/local/etc/zabbix_proxy.conf**

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

Note:
При использовании SQLite необходимо указать полный путь к файлу базы данных; пользователь БД и пароль 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 PHP-файлов

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

Распространённые расположения каталогов HTML-документов для веб-серверов Apache включают:

- /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 корневого HTML-каталога. 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 планируется использовать любой другой язык, кроме английского, инструкции см. в разделе Установка дополнительных языков веб-интерфейса.

Установка веб-интерфейса

Пожалуйста, смотрите страницу Установка веб-интерфейса для получения информации о мастере установки веб-интерфейса Zabbix.

Установка загружаемых плагинов Zabbix агент 2

Установка загружаемых плагинов Zabbix агент 2 требуется только в том случае, если вы хотите выполнять мониторинг целей, не поддерживаемых встроенными плагинами (например, серверов или кластеров MongoDB, PostgreSQL и его форков и т. д.). См. полный список загружаемых плагинов и встроенных плагинов.

Attention:
Перед установкой плагина, пожалуйста, ознакомьтесь с его файлом README. Он может содержать особые требования и инструкции по установке.

Чтобы установить из исходного кода, сначала загрузите и распакуйте архив с исходным кодом загружаемого плагина.

Чтобы скомпилировать плагин, перейдите в каталог с распакованным плагином и выполните `make`:

```
make
```

Note:
Для сборки загружаемых плагинов Zabbix агент 2 должна быть установлена поддерживаемая версия Go.

Исполняемый файл плагина может быть размещен где угодно, если он может быть загружен Zabbix агент 2. Укажите путь к бинарному 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 агент 2:

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

Перейдите к разделу настройки, чтобы получить более подробную информацию 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` - запустить самотестирование (требуется реальная цель мониторинга, например база данных PostgreSQL)
- `make style` - проверить стиль кода Go с помощью `golangci-lint`
- `make format` - отформатировать код Go с помощью `go fmt`
- `make dist` - создать архив исходного кода, включающий все зависимости

Установка Java gateway

Устанавливать Java gateway требуется только в том случае, если вы хотите осуществлять мониторинг JMX-приложений. Java gateway является легковесным и не требует базы данных.

Для установки из исходных кодов сначала загрузите и распакуйте архив с исходным кодом.

Чтобы скомпилировать Java gateway, запустите скрипт `./configure` с параметром `--enable-java`. Рекомендуется указать параметр `--prefix`, чтобы задать путь установки, отличный от используемого по умолчанию /usr/local, поскольку установка Java gateway создаст целое дерево каталогов, а не только один исполняемый файл.

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

Чтобы скомпилировать и упаковать Java gateway в JAR-файл, выполните `make`. Обратите внимание, что для этого шага вам потребуются 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, то 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 разделу настройки, чтобы получить более подробную информацию о настройке и запуске Java gateway.

Установка Zabbix web service

Установка Zabbix web service требуется только в том случае, если вы хотите использовать запланированные отчеты.

Для установки из исходного кода сначала загрузите и распакуйте архив с исходным кодом.

Чтобы скомпилировать Zabbix web service, запустите скрипт `./configure` с параметром `--enable-webservice`.

> **Note:**
> Для сборки Zabbix web service должна быть установлена поддерживаемая версия Go.

Запустите zabbix_web_service на машине, где установлен веб-сервис:

```
zabbix_web_service
```

Перейдите к разделу настройки, чтобы получить более подробную информацию о настройке генерации запланированных отчетов.

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 в некоторых системах

Если Zabbix аварийно завершается или зависает из-за переполнения стека, вы можете увеличить размер стека для каждого потока с помощью опции `--with-stacksize` при настройке исходного кода. Эта проблема может возникать в системах с низкими значениями ограничения размера стека потока по умолчанию, особенно во время предобработки, когда создаётся несколько потоков.

Следующий пример задаёт размер стека 512 КБ для каждого потока:

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

Вы можете проверить ограничения размера стека потока 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 сайте Zabbix.

Выберите вашу операционную систему и компонент Zabbix, чтобы сгенерировать инструкции по установке, подходящие для вашей среды. См. также примечания по установке пакетов на этой странице для получения важной дополнительной информации, а также раздел Установка и настройка.

Пакеты доступны для следующих дистрибутивов Linux:

- Red Hat Enterprise Linux и его производные: 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 базе Debian) предоставляют собственные пакеты Zabbix. Эти пакеты **не** поддерживаются Zabbix и могут быть устаревшими или не содержать новейших возможностей и исправлений ошибок. Рекомендуется использовать только официальные пакеты из официального репозитория Zabbix. Если вы ранее установили Zabbix из репозитория вашей операционной системы, см. шаги по обновлению пакетов Zabbix из репозиториев ОС.

Пакеты поддерживают базу данных MySQL/PostgreSQL и веб-сервер Apache/Nginx. Обратите внимание, что сервер Zabbix и прокси не могут использовать одну и ту же базу данных; используйте разные имена баз данных, если оба компонента установлены на одном узле сети.

При необходимости отдельные пакеты для Zabbix агент/агент 2, Zabbix get и Zabbix sender доступны в официальном репозитории Zabbix.

Zabbix также предоставляет предварительно скомпилированные двоичные файлы Zabbix агент для операционных систем, отличных от Linux; см.:

- Установка агента Windows из MSI
- Установка агента macOS из PKG
- Устаревшие двоичные файлы (для старых/менее распространённых систем, таких как HP-UX, NetBSD, Tru64 и старые версии 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; инструкции по соответствующей настройке см. в разделе Конфигурация SELinux.
- При использовании TimescaleDB см. дополнительные инструкции в разделе Настройка TimescaleDB.
- Если вы устанавливаете Zabbix Java gateway (для мониторинга JMX-приложений), см. дополнительные инструкции по настройке для систем на базе RHEL и систем на базе Debian.
- Для запуска Zabbix агент от имени root см. раздел Запуск агента от имени root.

Следующие примечания относятся к RHEL и производным от него системам:

- Если вы включили репозиторий EPEL для EL9, 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. В системах, где включён Security-Enhanced Linux (SELinux), может потребоваться добавить правила SELinux, чтобы разрешить Zabbix создавать/использовать сокеты домена UNIX в каталоге SocketDir. Файлы сокетов используются Zabbix сервером (alerter, preprocessing, IPMI) и Zabbix прокси (IPMI) и присутствуют, пока процесс выполняется.

Если SELinux включён в режиме enforcing, выполните следующие команды, чтобы разрешить взаимодействие между веб-интерфейсом Zabbix и сервером:

Для RHEL 7 (и более поздних версий), AlmaLinux, CentOS Stream, Oracle Linux, Rocky Linux 8 (и более поздних версий):

```
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 версии 7:

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

После применения настроек SELinux перезапустите Apache:

```
systemctl restart httpd
```

При необходимости вы можете установить предопределённый пакет `zabbix-selinux-policy` из официального репозитория Zabbix. Этот пакет предоставляется для всех поддерживаемых версий ОС, чтобы упростить развёртывание Zabbix и предотвратить отключение SELinux пользователями из-за сложности настройки.

> **Attention:**
> Для максимальной безопасности рекомендуется задать пользовательские настройки SELinux.

Пакет `zabbix-selinux-policy` содержит базовую политику SELinux, которая разрешает Zabbix создавать и использовать сокеты, а также включает возможность подключения HTTPd 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 бинарных файлов Zabbix. Они не требуются для обычной установки или работы, но полезны для расширенного поиска и устранения неисправностей.

Чтобы включить репозиторий `zabbix-debuginfo`:

- В RHEL 7 отредактируйте `/etc/yum.repos.d/zabbix.repo` и установите `enabled=1` для секции `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 секции `zabbix-debuginfo`:

```
[zabbix-debuginfo]
name=Zabbix Official Repository debuginfo
type=rpm-md
baseurl=https://repo.zabbix.com/zabbix/8.0/stable/sles/15/x86_64/debuginfo/
gpgcheck=1
gpgkey=https://repo.zabbix.com/zabbix/8.0/stable/sles/15/x86_64/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 конкретных компонентов:

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

Включение репозиториев нестабильных выпусков

Приведённые ниже инструкции предназначены для включения репозиториев нестабильных выпусков Zabbix (по умолчанию отключены), используемых для кандидатов в релиз минорных версий Zabbix.

Сначала установите или обновите пакет zabbix-release до последней версии. Чтобы включить rc-пакеты в вашей системе, выполните следующие действия:

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

агент Zabbix можно установить в Windows с помощью 32-битных или 64-битных MSI-пакетов установки, доступных для загрузки.

Минимальные требования к ОС для установки из MSI:

- **Для агента Zabbix:** Windows XP (64-bit) или Windows Server 2003
- **Для агента Zabbix 2:** Windows 10 (32-bit) или Windows Server 2016

32-битные пакеты нельзя устанавливать в 64-битных системах.

Пакеты включают:

- поддержку TLS (настройка TLS необязательна)
- утилиты Zabbix get и Zabbix sender (могут быть установлены вместе с агентом Zabbix/агентом 2 или отдельно)

> **Attention:**
> Пакеты агента Zabbix 2 не включают загружаемые плагины (MongoDB, PostgreSQL, MSSQL), которые необходимо загрузить и установить отдельно.

Установка может быть выполнена с помощью мастера установки или командной строки.

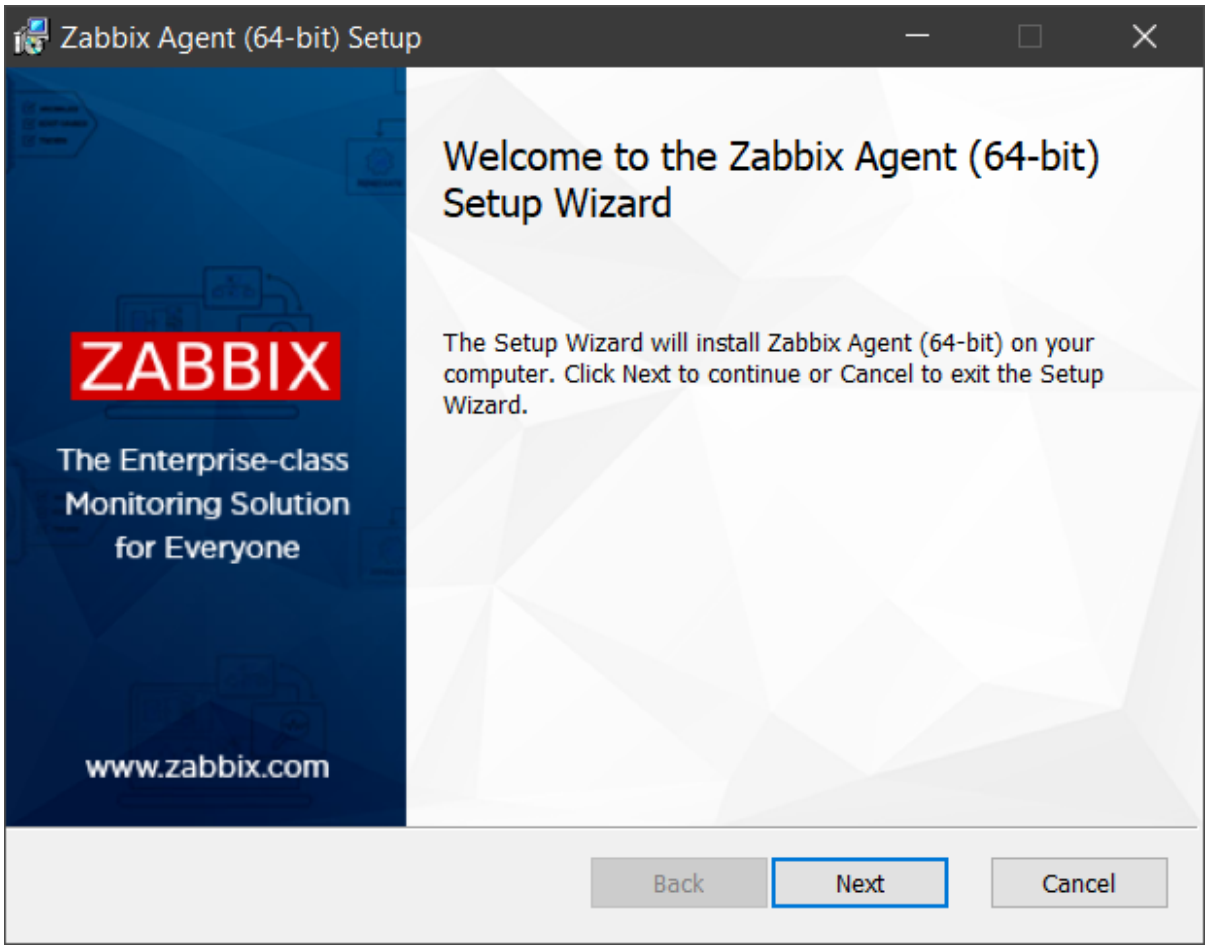
Хотя установка с использованием MSI-пакетов полностью поддерживается, для корректной обработки ошибок рекомендуется установить как минимум Microsoft .NET Framework 2.

Attention:
Рекомендуется использовать расположение установки по умолчанию, предлагаемое установщиком. Использование пользовательского расположения без необходимых разрешений может поставить под угрозу безопасность установки.

Установка с помощью мастера установки

Следующие шаги установки применимы как к Zabbix агент, так и к Zabbix агент 2.

1. Дважды щелкните загруженный MSI-файл, чтобы начать установку:

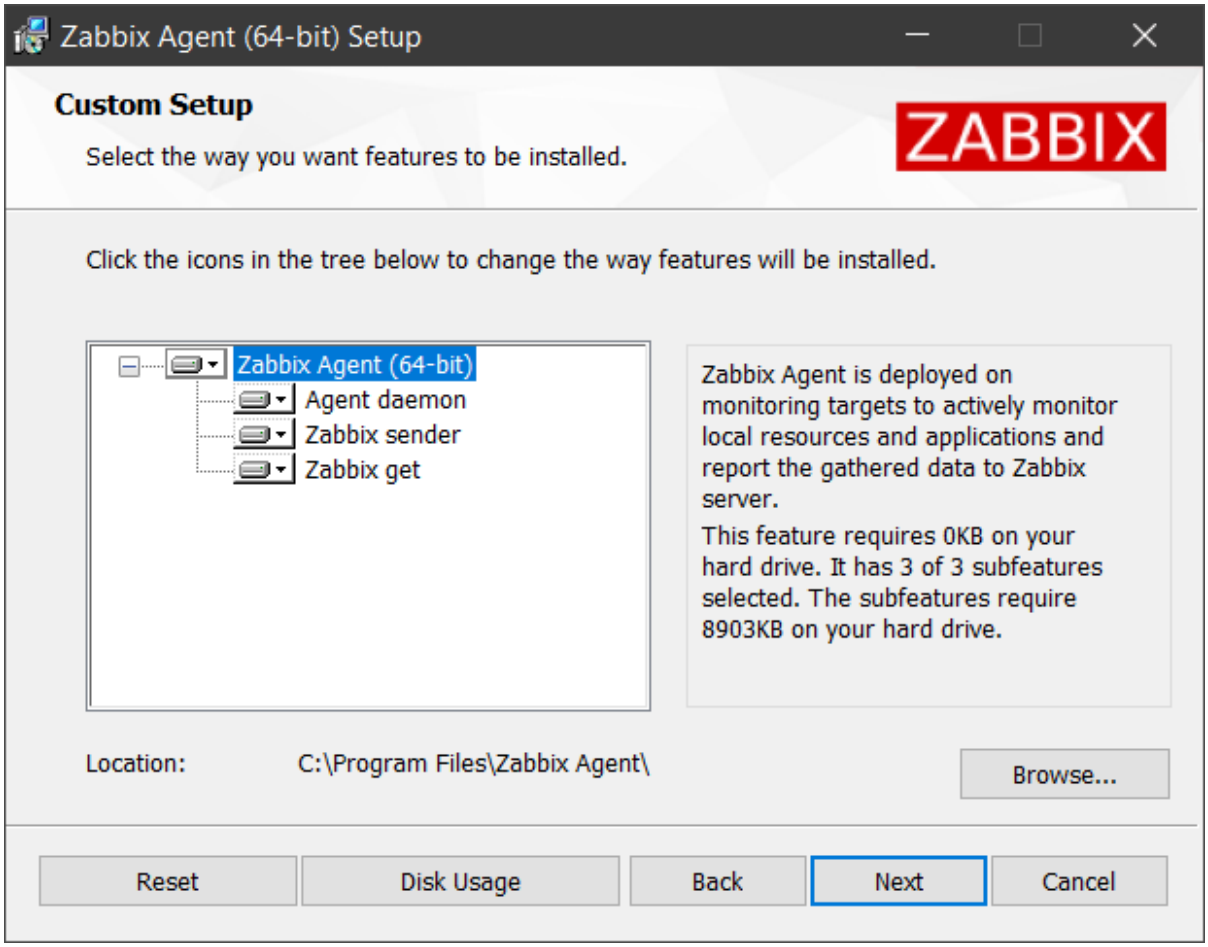


2. Примите лицензионное соглашение с конечным пользователем:



3. Выберите компоненты Zabbix (демон агента, Zabbix sender, Zabbix get), которые необходимо установить:

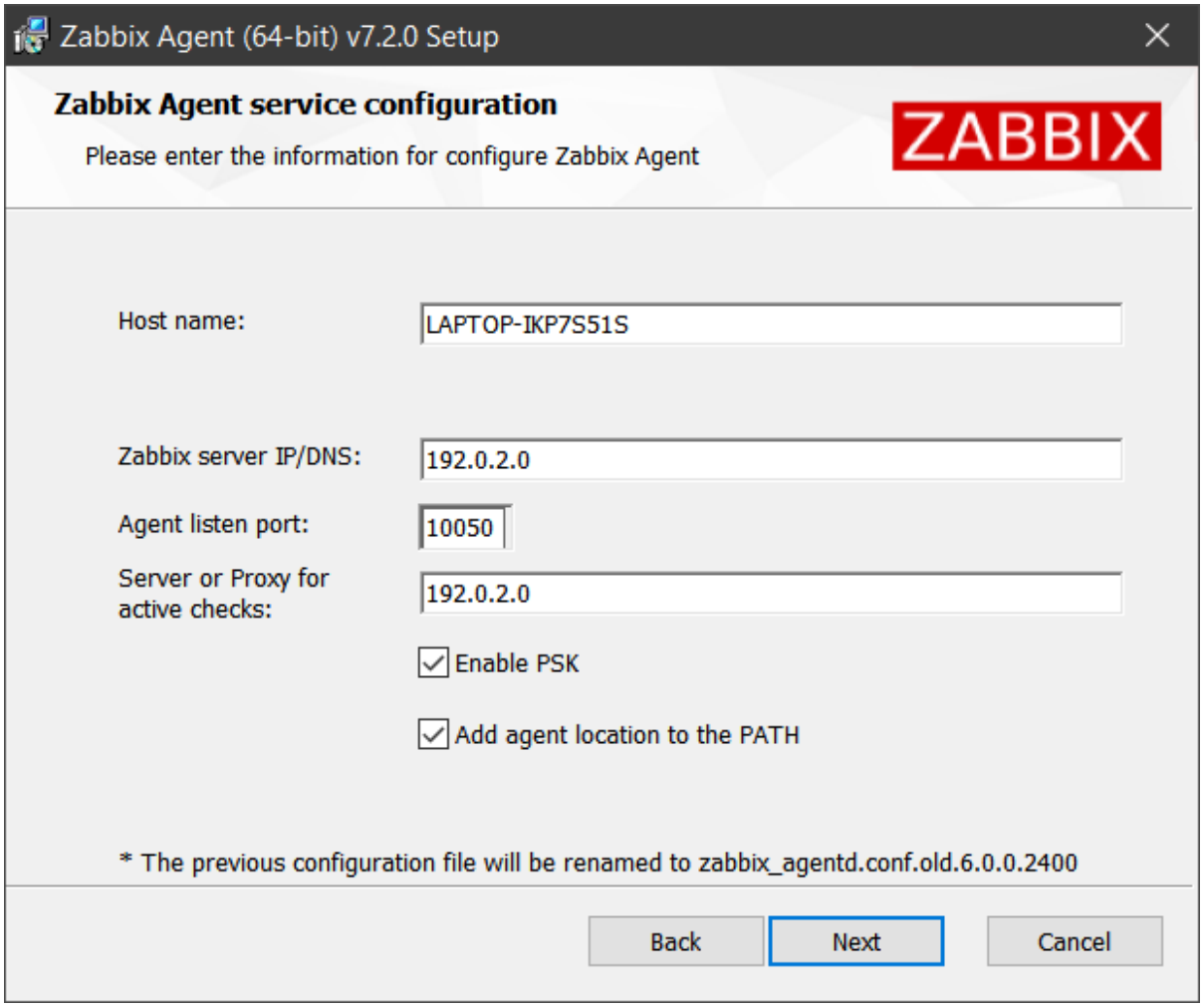
Attention:
Рекомендуется использовать расположение установки по умолчанию, предлагаемое установщиком. Использование пользовательского расположения без необходимых разрешений может поставить под угрозу безопасность установки.



4. Настройте следующие параметры. Их значения будут заданы в файле конфигурации Zabbix агент:

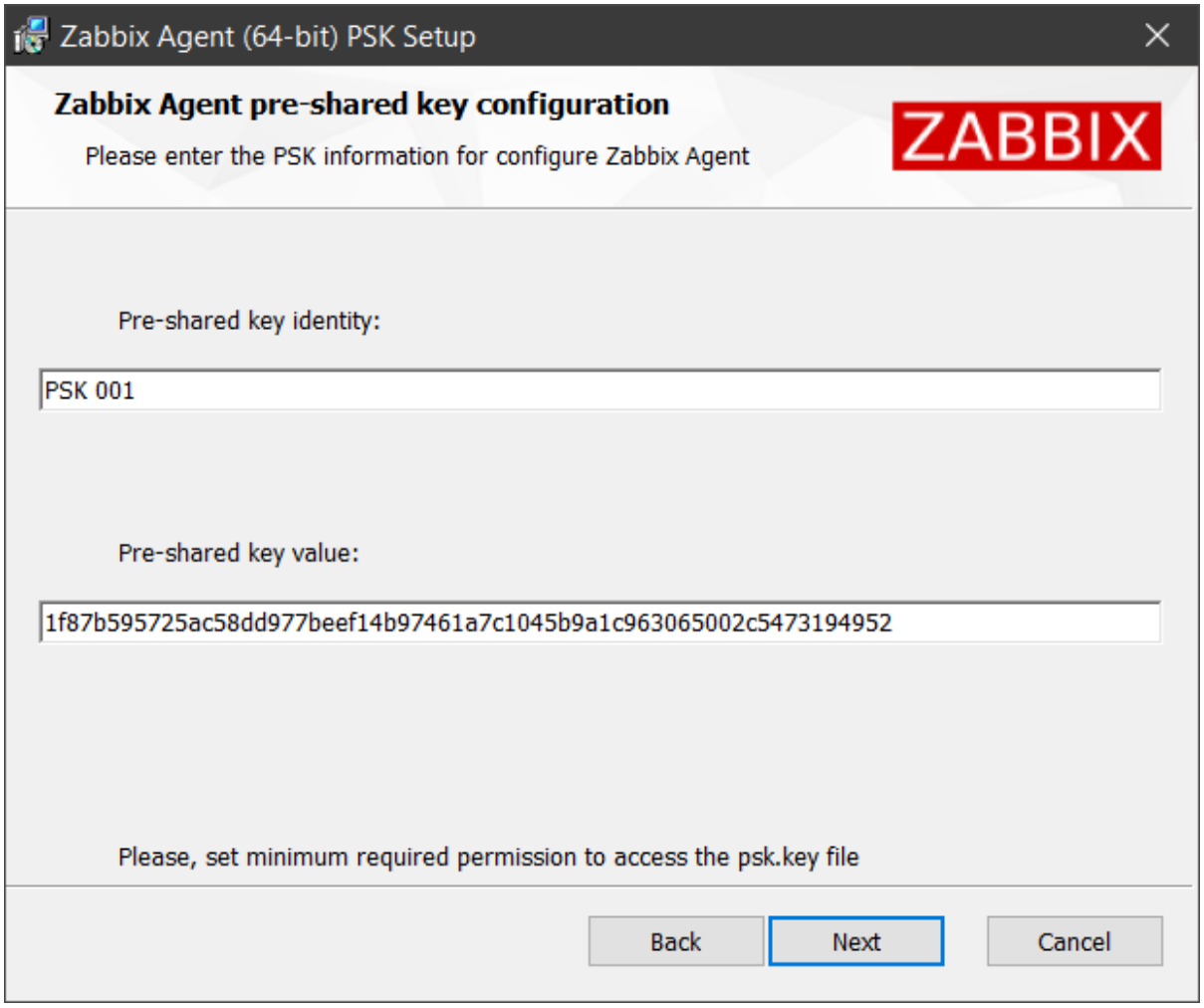
Параметр	Описание
Имя узла сети	Имя узла сети машины, на которой устанавливается Zabbix агент. Задает параметр Hostname.
IP/DNS Zabbix сервера	Список IP-адресов, разделенных запятыми, при необходимости в нотации CIDR, или DNS-имен Zabbix серверов или Zabbix прокси. Этот параметр **обязателен**. Задает параметр Server.
Порт прослушивания агента	агент будет прослушивать этот порт для соединений от сервера. Задает параметр ListenPort.
Сервер или прокси для активных проверок	Адрес Zabbix сервера/прокси или конфигурация кластера для получения активных проверок. Адрес сервера/прокси представляет собой IP-адрес или DNS-имя и необязательный порт, разделенные двоеточием. Задает параметр ServerActive.
Включить PSK	Установите этот флажок, чтобы включить поддержку TLS с использованием предварительно разделяемых ключей. Задает параметрам TLSConnect и TLSAccept значение psk.
Добавить расположение агента в PATH	Установите этот флажок, чтобы добавить расположение Zabbix агент в системную переменную PATH.

Note:
Если будет обнаружен существующий Zabbix агент, будут отображены значения параметров из его файла конфигурации. Кроме того, существующий файл конфигурации будет переименован во время установки, и будет создан новый файл конфигурации.



5. Настройте параметры PSK, если на предыдущем шаге вы установили флажок *Включить PSK*. Эти параметры также будут заданы в файле конфигурации Zabbix агент:

Параметр	Описание
Идентификатор предварительно разделяемого ключа	Строка идентификатора предварительно разделяемого ключа. Задает параметр TLSPSKIdentity.
Значение предварительно разделяемого ключа	Строковое значение предварительно разделяемого ключа. Создает файл psk.key, содержащий ключ, и задает параметру TLSPSKFile расположение ключа (по умолчанию: `C:\Program Files\Zabbix Agent\psk.key`). **Рекомендуется** ограничить доступ к файлу предварительно разделяемого ключа, настроив параметры безопасности файла так, чтобы только Zabbix агент (или пользователь, от имени которого запущен агент) мог его читать.

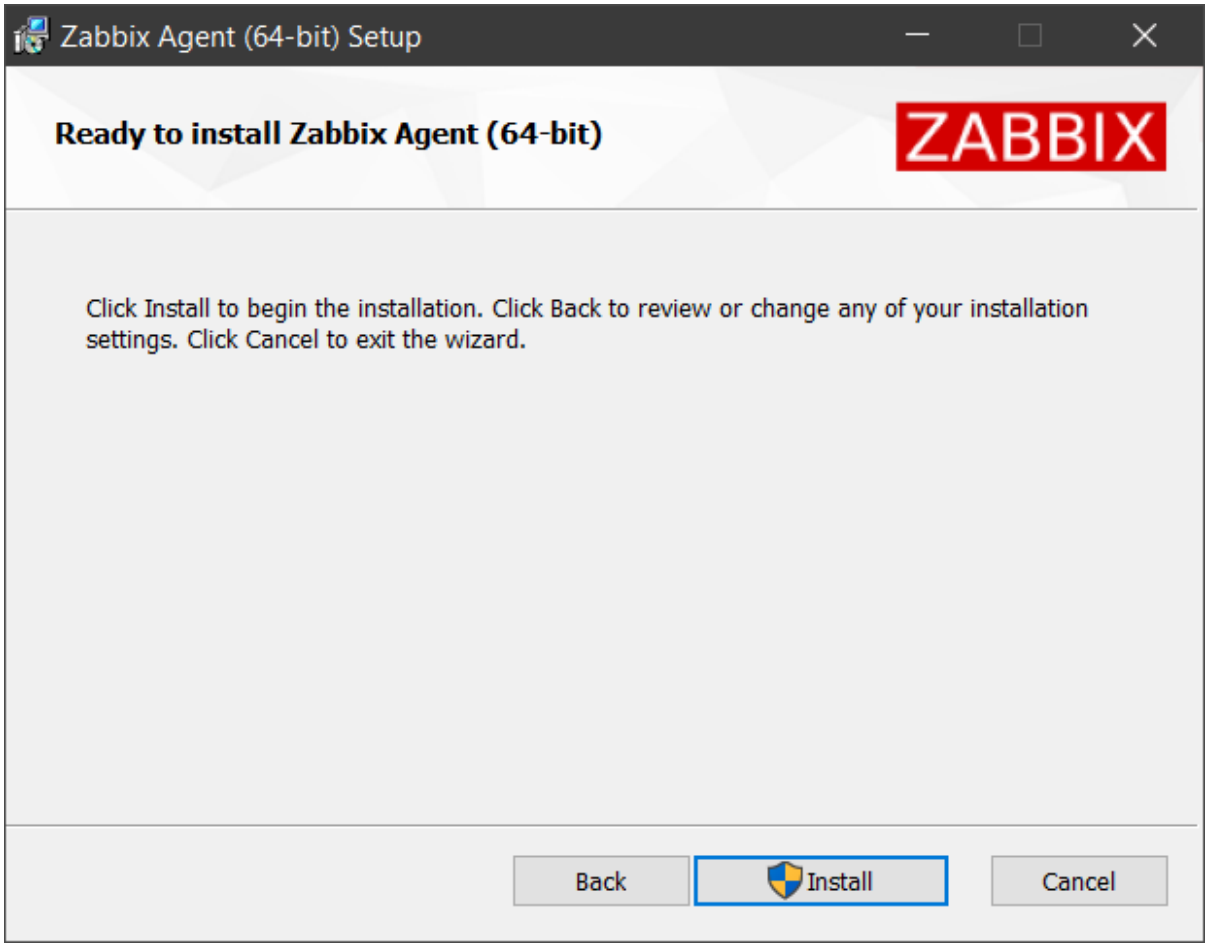


6. Нажмите *Install*, чтобы начать установку.

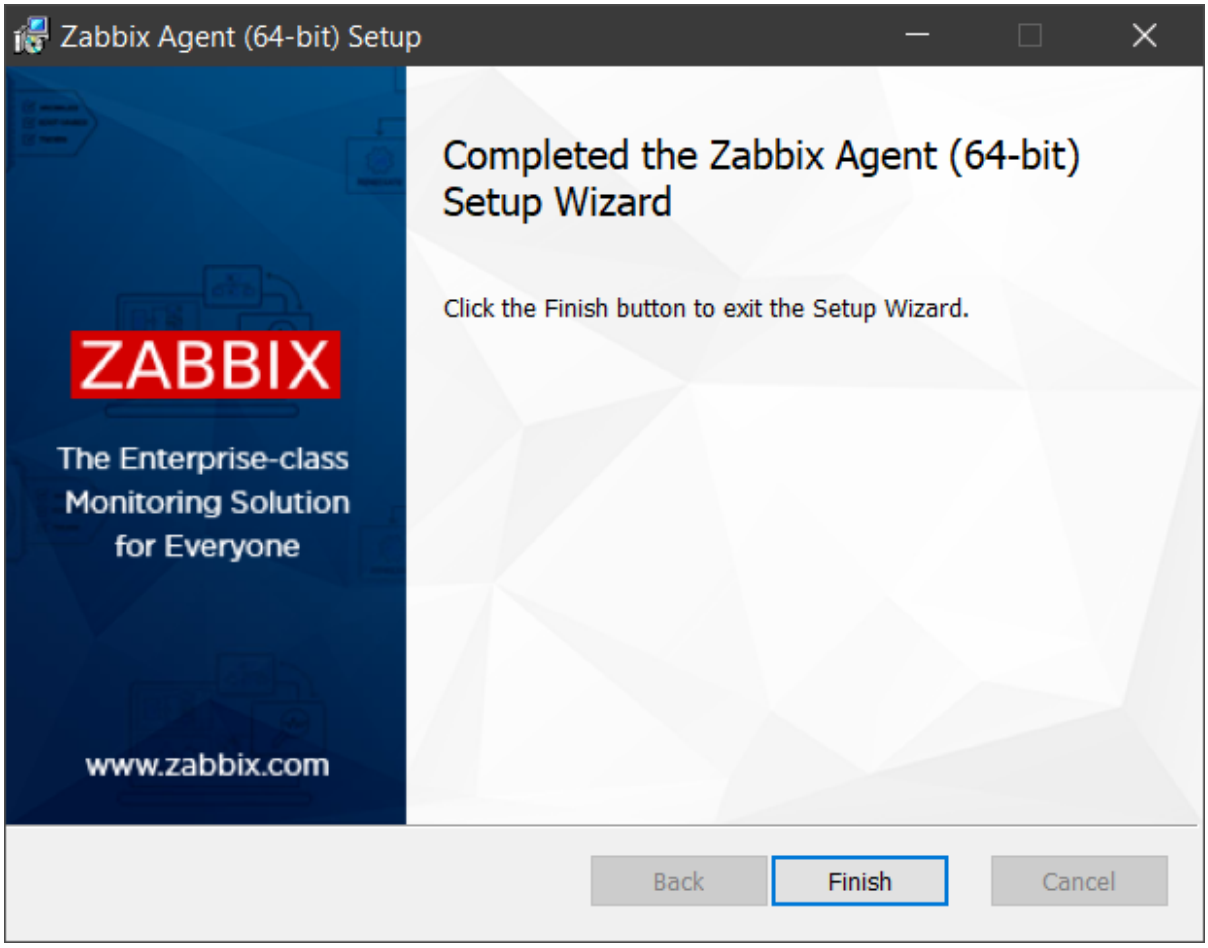
Все выбранные компоненты Zabbix и файл конфигурации Zabbix агент будут установлены в указанное вами расположение (по умолчанию: `C:\Program Files\Zabbix Agent`). То же относится и к Zabbix агент 2, за исключением того, что дополнительные файлы конфигурации для его встроенных плагинов будут установлены 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 запущена другая версия Zabbix агент, вам будет предложено либо закрыть приложение и попытаться перезапустить его, либо оставить его открытым, и в этом случае потребуется перезагрузка.



7. Нажмите кнопку *Finish*, 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 метод позволяет выполнять автоматическую установку без участия пользователя и использовать пользовательские параметры конфигурации.

Поддерживаемые параметры

Пакеты MSI-установщика Zabbix агент поддерживают следующие параметры как для Zabbix агент, так и для Zabbix агент 2.

Note:
Параметры Zabbix агент/agent2 задаются в файле конфигурации во время установки. Нажмите на имя параметра, чтобы просмотреть его подробное описание и примеры настройки на странице Zabbix агент (Windows). Для Zabbix агент 2 см. страницу Zabbix агент 2 (Windows).

Параметр	Описание
ADDDEFAULT	Список компонентов, разделённых запятыми, которые будут установлены с конфигурацией по умолчанию. Подробнее см. свойство ADDDEFAULT. Возможные значения: `AgentProgram`, `GetProgram`, `SenderProgram`, `ALL` Пример: `ADDDEFAULT=AgentProgram,GetProgram`
ADDLOCAL	Список компонентов, разделённых запятыми, которые будут установлены локально. Подробнее см. свойство ADDLOCAL. Возможные значения: `AgentProgram`, `GetProgram`, `SenderProgram`, `ALL` Пример: `ADDLOCAL=AgentProgram,SenderProgram`
ALLOWDENYKEY	Список параметров AllowKey или DenyKey, разделённых точкой с запятой, для ограничения проверок Zabbix агент. При необходимости используйте обратную косую черту для экранирования разделителя (`\;`). Устанавливает 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 шаблонному файлу конфигурации для Zabbix агент. 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 агент.
ENABLEPATH	Используйте `ENABLEPATH=1`, чтобы добавить расположение Zabbix агент в системную переменную PATH.
ENABLEPERSISTENTBUFFER	Только для Zabbix агент 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 или всех файлов в каталоге, разделённых точкой с запятой, которые нужно включить в файл конфигурации Zabbix агент.
INSTALLFOLDER	Полный путь к папке, в которую будут установлены компоненты Zabbix и файл конфигурации Zabbix агент. Для Zabbix агент 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, разделённых запятыми, на которых агент должен прослушивать соединения.
LISTENPORT	Порт, на котором агент будет прослушивать подключения от сервера.
LOGFILE	Имя файла журнала Zabbix агент.
LOGTYPE	Тип вывода журнала.

Параметр	Описание
NONMSICONFNAME	Полный путь к пользовательскому файлу конфигурации для Zabbix агент. Во время установки все 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 агент 2. Файл, в котором Zabbix агент 2 должен хранить базу данных SQLite.
PERSISTENTBUFFERPERIOD	Только для Zabbix агент 2. Период времени, в течение которого данные должны храниться при отсутствии соединения с сервером или прокси.
SERVER	Список IP-адресов, разделённых запятыми, при необходимости в нотации CIDR, или DNS-имён серверов Zabbix или прокси Zabbix. Этот параметр **обязателен**, кроме случая, когда STARTAGENTS установлен в 0.
SERVERACTIVE	Адрес сервера/прокси Zabbix или конфигурация кластера для получения активных проверок.
SKIP	Используйте `SKIP=fw`, чтобы MSI-установщик не добавлял правило исключения в брандмауэр Windows для Zabbix агент.
SOURCEIP	Исходный IP-адрес для исходящих подключений к серверу Zabbix или прокси Zabbix, либо для выполнения подключений при обработке 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 агент. Возможные значения: **automatic** — запускать службу автоматически при запуске Windows; **delayed** — *(по умолчанию)* отложить запуск службы до 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 её нельзя было запустить пользователем или приложением. Пример: `STARTUPTYPE=disabled`
STATUSPORT	Только для Zabbix агент 2. Если задан, агент будет прослушивать этот порт для HTTP-запросов состояния (http://localhost:<port>/status).
TIMEOUT	Определяет, как долго ждать (в секундах) установления соединения и обмена данными с прокси или сервером Zabbix.
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 корневых центров сертификации для проверки сертификата другой стороны.
TLSCERTFILE	Полный путь к файлу, содержащему сертификат агента или цепочку сертификатов.
TLSCONNECT	Как агент должен подключаться 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 агент.
TLSPSKFILE	Полный путь к файлу, содержащему предварительно согласованный ключ Zabbix агент. Если заданы и TLSPSKFILE, и TLSPSKVALUE, значение TLSPSKVALUE будет записано в файл, указанный в TLSPSKFILE. **Рекомендуется** ограничить доступ к файлу предварительно согласованного ключа, настроив параметры безопасности файла так, чтобы только Zabbix агент (или пользователь, под которым запущен агент) мог его читать.
TLSPSKIDENTITY	Строка идентификатора предварительно согласованного ключа.
TLSPSKVALUE	Строковое значение предварительно согласованного 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	Разрешает передавать любые символы в аргументах пользовательских параметров.

Примеры

Следующий пример устанавливает Zabbix агент с пользовательской конфигурацией. Он также включает поддержку TLS с использованием предварительно разделяемых 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=192.0.2.0^
 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 Zabbix агент 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:
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 все 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 должны быть сохранены.

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

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

Подключаемые плагины Zabbix агент 2

Подключаемые плагины Zabbix агент 2 loadable plugins можно установить в Windows с помощью 64-разрядных пакетов установщика MSI, 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.

Аналогично Zabbix агент/agent2, подключаемые плагины можно установить с помощью мастера установки или из командной строки.

Установка с помощью мастера установки

1. Дважды щелкните загруженный MSI-файл, чтобы начать установку.

2. Примите лицензионное соглашение с конечным пользователем.

3. Выберите загружаемые плагины Zabbix агент 2, которые необходимо установить.

Attention:
Рекомендуется использовать расположение установки по умолчанию, предлагаемое программой установки. Использование пользовательского расположения без необходимых разрешений может поставить под угрозу безопасность установки.

4. Нажмите *Install*, чтобы начать установку.

Все выбранные загружаемые плагины Zabbix агент 2 будут установлены в указанное вами расположение (по умолчанию: `C:\Program Files\Zabbix Agent 2`), а их файлы конфигурации — в подкаталог `zabbix_agent2.d`.

5. Нажмите кнопку *Finish*, чтобы выйти из мастера установки.

Установка из командной строки

Подключаемые плагины Zabbix агент 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 агент 2 поддерживают следующие параметры.

Parameter	Description
ADDDEFAULT	Список компонентов, разделённых запятыми, для установки в их конфигурации по умолчанию. Дополнительную информацию смотрите в свойстве ADDDEFAULT. Возможные значения: `ALL`, `CephPlugin`, `EmberplusPlugin`, `MongodbPlugin`, `MssqlPlugin`, `NvidiagpuPlugin`, `PostgresqlPlugin` Пример: `ADDDEFAULT=MongodbPlugin,PostgresqlPlugin`
ADDLOCAL	Список компонентов, разделённых запятыми, для локальной установки. Дополнительную информацию смотрите 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 будут установлены 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 Zabbix агент 2 из исходных кодов в Windows 10 (64-разрядной или 32-разрядной).

И 32-разрядные, и 64-разрядные версии можно собрать на 64-разрядной платформе, но на 32-разрядной платформе можно собрать только 32-разрядную версию.

Для сборки Zabbix агент 2 требуется:

- инструменты сборки MinGW
- язык программирования Go
- OpenSSL (для возможностей шифрования в Zabbix)
- PCRE2 (Perl Compatible Regular Expressions; для возможностей сопоставления с образцом с использованием регулярных выражений в Zabbix)

Вы можете собрать Zabbix агент 2 одним из следующих способов:

- С использованием vcpkg — автоматизированный подход, упрощающий управление зависимостями с помощью менеджера пакетов C++.
- Ручная сборка — 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 защищённые 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 `C:\Zabbix\x86` для 32-разрядных сборок).

Сборка Zabbix агента 2 с помощью vcpkg

В этом разделе содержатся инструкции по сборке Zabbix агента с помощью vcpkg — менеджера пакетов, который упрощает управление зависимостями и интеграцию с проектами на C++.

1. Загрузите и установите Build Tools for Visual Studio 2022. Во время установки обязательно выберите рабочую нагрузку *Desktop development with C++*, которая включает менеджер пакетов vcpkg.

2. Загрузите и установите Go (доступен в виде MSI-установщика). Во время установки обязательно укажите `C:\Zabbix\Go` в качестве папки установки.

3. Загрузите дистрибутив MinGW, использующий 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`

Затем распакуйте его в `C:\Zabbix\mingw64` (или `C:\Zabbix\mingw32` для 32-разрядных сборок).

4. Инициализируйте vcpkg и установите зависимости, необходимые для сборки Zabbix агента 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

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

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

5. Загрузите 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:

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

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

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

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

- Для 32-разрядных сборок:

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

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

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

:: Run the build process:
mingw32-make GOFLAGS="-buildvcs=false" ARCH=x86 ^
    PCRE2="%vcpkg%" ^
    OPENSSL="%vcpkg%" ^
    all
```

7. Скомпилируйте Zabbix агент 2, выполнив скрипт:

```
build.bat
```

После компиляции исполняемый файл Zabbix агента 2 будет находиться в `C:\Zabbix\zabbix-8.0.0\bin\win64` (для 64-разрядных сборок) или `C:\Zabbix\zabbix-8.0.0\bin\win32` (для 32-разрядных сборок). Файлы конфигурации Zabbix агента 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

#### For 64-bit builds:
copy C:\Zabbix\zabbix-8.0.0\bin\win64\zabbix_agent2.exe C:\Zabbix\agent2\

#### For 32-bit builds:
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.win.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.win.conf
```

При необходимости продолжите со сборкой подключаемых плагинов для Zabbix агента 2.

Компиляция загружаемых плагинов Zabbix агент 2

1. Загрузите исходный код плагина Zabbix, соответствующий версии вашего Zabbix агент 2 (например, `zabbix-agent2-plugin-pos` и распакуйте его в `C:\Zabbix`.

> **Attention:**
> Перед установкой плагина проверьте его файл README. Он может содержать особые требования и инструкции по установке.

2. Перейдите в каталог распакованного 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 исполняемый файл плагина Zabbix агент 2 и его файл конфигурации будут находиться в том же каталоге плагина.

Исполняемый файл плагина можно разместить в любом месте, если он может быть загружен Zabbix агент 2. Укажите путь к бинарному 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 необходимо указать в параметре Include файла конфигурации Zabbix агент 2:

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

Для получения дополнительных сведений о настройке плагинов перейдите к разделу настройка.

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. Загрузите и установите MSYS2 (доступен в виде MSI-установщика). Во время установки обязательно укажите `C:\Zabbix\msys64` в качестве папки установки.

2. Загрузите и установите Go (доступен в виде MSI-установщика; см. поддерживаемые в настоящее время версии Go). Во время установки обязательно укажите `C:\Zabbix\Go` в качестве папки установки.

3. Загрузите дистрибутив MinGW, использующий 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`

Затем распакуйте его в `C:\Zabbix\mingw64` (или в `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 каталог с распакованным OpenSSL и создайте следующий скрипт `build.sh`:

- Для 64-битных 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-битных 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:
Убедитесь, что у пользователей без прав администратора отозван доступ на запись к каталогу C:\Zabbix\x64\OpenSSL-Win64-350-static. В противном случае агент будет загружать настройки SSL из пути, который может быть изменён непривилегированными пользователями, что создаёт потенциальную уязвимость 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 Windows без библиотек OpenSSL; однако при выпуске новой версии OpenSSL с исправлениями ошибок потребуется повторно скомпилировать агент Zabbix.
- Без параметра no-shared Zabbix использует DLL OpenSSL во время выполнения. Это означает, что обновления OpenSSL могут не требовать повторной компиляции агента Zabbix; однако при копировании на другие компьютеры DLL OpenSSL также необходимо будет скопировать.

Для получения дополнительной информации о других параметрах конфигурации OpenSSL см. документацию 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 Zabbix агент 2

1. Загрузите архив с исходным кодом Zabbix и распакуйте его в C:\Zabbix\zabbix-8.0.0.

Если вам нужно создать архив с исходным кодом из необработанного репозитория исходного кода (например, чтобы применить пользовательские патчи или выполнить сборку из последней версии исходного кода), выполните следующие команды на машине с **Linux** и установленным Go (требуется для настройки Zabbix агент 2):

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

Это создаст архив с исходным кодом, который затем можно скопировать на машину с Windows.

2. Откройте командную строку с правами администратора. Затем перейдите в каталог сборки Zabbix и скомпилируйте Zabbix агент; убедитесь, что правильно указали каталоги, в которых установлены OpenSSL и PCRE2:

- Для 64-битных 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-битных 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-Win64-350-static"

####         TLS:
mingw32-make ARCH=x86 PCRE2="C:\Zabbix\x86\PCRE2"
```

После компиляции исполняемый файл Zabbix агент 2 будет находиться в C:\Zabbix\zabbix-8.0.0\bin\win64 (или C:\Zabbix\zabbix-8.0.0\bin\win32 для 32-битных сборок). Файлы конфигурации Zabbix агент 2 находятся в C:\Zabbix\zabbix-8.0.0\src\go\conf.

Чтобы запустить агент, скопируйте исполняемый файл zabbix_agent2.exe и его файлы конфигурации в отдельную папку (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.win.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.win.conf
```

При необходимости продолжите с компиляцией подключаемых плагинов Zabbix агент 2.

Сборка агента Zabbix в Windows

Обзор

На этой странице показано, как собрать агент Zabbix из исходного кода в Windows 10 (64-разрядной).

Эти инструкции применимы к версиям Windows, которые поддерживают Visual Studio 2022.

Для сборки агента Zabbix требуется:

- компилятор C (входит в состав Build Tools for Visual Studio 2022)
- OpenSSL (для функций шифрования в Zabbix)
- PCRE2 (Perl Compatible Regular Expressions; для функций сопоставления с образцом регулярных выражений в Zabbix)

Вы можете собрать агент Zabbix одним из следующих способов:

- С помощью vcpkg — автоматизированный подход, который упрощает управление зависимостями с помощью менеджера пакетов C++.
- Ручная сборка — ручной подход, который требует установки всех зависимостей перед компиляцией агента.

В зависимости от ваших потребностей в мониторинге могут потребоваться дополнительные библиотеки. Для получения дополнительной информации смотрите Требования.

Attention:
Перед началом процесса сборки, пожалуйста, имейте в виду:

- 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 защищённые 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 содержатся инструкции по сборке агента Zabbix с помощью vcpkg, менеджера пакетов, который упрощает управление зависимостями и интеграцию с проектами C++.

1. Загрузите и установите Build Tools for Visual Studio 2022. Во время установки обязательно выберите рабочую нагрузку *Desktop development with C++*, которая включает инструменты, необходимые для сборки агента с помощью vcpkg:

- компилятор C (Microsoft Visual C++)
- инструмент командной строки NMake
- менеджер пакетов vcpkg
- x64 Native Tools Command Prompt

2. 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. Загрузите 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`; обязательно правильно укажите каталоги, в которых установлены OpenSSL и PCRE2:

```
:: 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. Скомпилируйте агент Zabbix, выполнив скрипт:

```
build.bat
```

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_agent.exe` и его файл конфигурации в отдельную папку (например, `C:\Zabbix\agent`), а затем запустите агент:

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

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

Сборка агента Zabbix вручную

Attention:
Этот метод сборки агента Zabbix 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 агента Zabbix, включая установку необходимых инструментов сборки и зависимостей (Perl, OpenSSL, PCRE2), а затем компиляцию агента.

Установка инструментов сборки

1. Загрузите и установите Build Tools for Visual Studio 2022. Во время установки обязательно выберите рабочую нагрузку *Desktop development with C++*, которая включает инструменты, необходимые для сборки агента вручную:

- компилятор C (Microsoft Visual C++)
- инструмент командной строки NMake
- командная строка x64 Native Tools Command Prompt

Установка OpenSSL

> **Note:**
> Чтобы скомпилировать Zabbix агент без поддержки TLS, перейдите к разделу Установка PCRE2.

1. Загрузите и установите Strawberry Perl (доступен в виде MSI-установщика). Во время установки обязательно укажите `C:\Zabbix\Strawberry` в качестве папки установки.

2. Установите Perl-модуль `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 был скомпилирован, установите его вручную. Дополнительную информацию см. в документации NASM.

4. Загрузите 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 Zabbix агент в Windows вы выбираете для OpenSSL пользовательский каталог (например, `C:\zabbix` или `C:\openssl-64bit`), обязательно отзовите права на запись в этот каталог у пользователей без прав администратора. В противном случае агент будет загружать настройки SSL из пути, который может быть изменён непривилегированными пользователями, что создаёт потенциальную уязвимость 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 Windows без библиотек OpenSSL; однако при выпуске новой версии OpenSSL с исправлениями ошибок потребуется повторно скомпилировать Zabbix агент.
- Без параметра `no-shared` Zabbix во время выполнения использует DLL OpenSSL. Это означает, что обновления OpenSSL могут не требовать повторной компиляции Zabbix агент; однако при копировании на другие компьютеры также потребуется скопировать DLL OpenSSL.

Дополнительную информацию о других параметрах конфигурации 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 безопасности. Если некоторые тесты завершаются ошибкой, рекомендации по устранению неполадок см. в документации OpenSSL.

```
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, заголовочные файлы, но не документацию), можно использовать `nmake install_sw`.

Установка PCRE2

1. Загрузите и установите CMake (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*.

2. Загрузите архив с исходным кодом PCRE2 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:
Если возникнут 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.

5. Соберите PCRE2 с помощью NMake:

```
nmake
```

6. Установите PCRE2:

```
cmake --install .
```

Компиляция агента Zabbix

1. Загрузите архив с исходным кодом Zabbix и распакуйте его в `C:\Zabbix\zabbix-8.0.0`.

Если вам нужно создать архив с исходным кодом из необработанного репозитория исходного кода (например, чтобы применить пользовательские патчи или выполнить сборку из последней версии исходного кода), выполните следующие команды на машине с **Linux**:

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

Это создаст архив с исходным кодом, который затем можно скопировать на машину с Windows.

2. Перейдите в каталог сборки Zabbix и скомпилируйте агент Zabbix (или другие компоненты); убедитесь, что вы правильно указали каталоги, в которых установлены OpenSSL и PCRE2:

```
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
```

После компиляции двоичные файлы компонентов Zabbix будут расположены в `C:\Zabbix\zabbix-8.0.0\bin\win64`. Файл конфигурации агента Zabbix расположен в `C:\Zabbix\zabbix-8.0.0\conf`.

Чтобы запустить агент, скопируйте `zabbix_agent.exe` и его файл конфигурации в отдельную папку (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.win.conf C:\Zabbix\agent\

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

Установка агента macOS из PKG

Обзор

Агент Zabbix можно установить в macOS с помощью установочных пакетов PKG, доступных для загрузки.

Пакеты агента Zabbix доступны как с шифрованием, так 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 будет запущен автоматически после установки или перезапуска.

При необходимости вы можете отредактировать файл конфигурации `/usr/local/etc/zabbix/zabbix_agentd.conf`.

Чтобы запустить агент вручную, выполните:

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

Чтобы остановить агент вручную:

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

Во время обновления существующий файл конфигурации не перезаписывается. Вместо этого создается новый файл `zabbix_agentd.conf.NEW`, который можно использовать для проверки и обновления существующего файла конфигурации, если это необходимо. Не забудьте перезапустить агент после внесения изменений в файл конфигурации вручную.

Устранение неполадок и удаление агента

В этом разделе перечислены некоторые полезные команды, которые можно использовать для устранения неполадок и удаления установленного Zabbix агент.

Проверьте, запущен ли Zabbix агент:

```
ps aux | grep zabbix_agentd
```

Проверьте, был ли Zabbix агент установлен из пакетов:

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

Просмотрите файлы, которые были установлены из пакета установщика (обратите внимание, что начальный символ / в этом представлении не отображается):

```
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
```

Остановите Zabbix агент, если он был запущен с помощью `launchctl`:

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

Удалите файлы (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
```

Удалите сведения о том, 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:

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

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
```

4 Обновление

Перед обновлением настоятельно рекомендуется ознакомиться со всеми соответствующими примечаниями по обновлению.

Также будет полезно знать, 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 HA-кластера
- Для базы данных TimescaleDB см. Обновление схемы TimescaleDB

Компоненты для обновления

Обновление Zabbix агентов рекомендуется, но не является обязательным.

Обновление Zabbix прокси настоятельно рекомендуется. Zabbix сервер полностью поддерживает прокси той же основной версии, что и сервер. Zabbix сервер также поддерживает прокси, версия которых **не старше** предыдущего LTS-релиза Zabbix сервера, но с ограниченной функциональностью (сбор данных, выполнение удаленных команд, немедленных проверок значений элементов данных). Обновление конфигурации также отключено, и устаревшие прокси будут работать только со старой конфигурацией.

> **Attention:**
> Прокси старше предыдущей LTS-версии сервера Zabbix или новее основной версии сервера Zabbix не поддерживаются. Сервер Zabbix будет игнорировать данные от неподдерживаемых прокси-серверов, и вся связь с сервером Zabbix прервётся с предупреждением. Для получения дополнительной информации см. Совместимость версий.

Чтобы свести к минимуму время простоя и потерю данных в процессе обновления, рекомендуется останавливать, обновлять и запускать сервер Zabbix и затем останавливать, обновлять и запускать Zabbix прокси один за другим. Во время простоя сервера работающие прокси продолжают собирать данные. Как только сервер будет запущен и заработает, не обновлённые прокси отправят данные новому серверу (хотя конфигурация прокси не будет обновлена) и частично останутся функциональными. Любые уведомления о проблемах во время простоя сервера Zabbix будут сгенерированы только после запуска обновлённого сервера.

Если прокси Zabbix запускается впервые и файл базы данных SQLite отсутствует, прокси создаст его автоматически.

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 запишет ошибку в журнал 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.

> **Note:**
> Обратите внимание, что после обновления некоторые интеграции со сторонним программным обеспечением в Zabbix могут быть затронуты, если внешнее программное обеспечение несовместимо с обновлённой версией Zabbix.

Доступны следующие примечания по обновлению:

Обновление с версии	Читать полные примечания по обновлению	Наиболее важные изменения между версиями
7.4.x	Для: Zabbix 8.0	Повышены минимально требуемые версии базы данных. Минимально требуемая версия PHP повышена с 8.0.0 до 8.2.0. Символ % добавлен в список `UnsafeUserParameters` для агент Zabbix и агент Zabbix 2.
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	Минимально требуемая версия PHP повышена с 7.4.0 до 8.0.0. Асинхронные poller-процессы для проверок агент, HTTP-агент и SNMP walk[oid]. Отдельная таблица базы данных для прокси. Изменено расположение по умолчанию для файла конфигурации агента Windows. Oracle DB объявлена устаревшей. Старый числовой тип значений (float) удалён.
6.2.x	Для: Zabbix 6.4 Zabbix 7.0 Zabbix 7.2 Zabbix 7.4 Zabbix 8.0	Минимально требуемая версия MySQL повышена с 8.0.0 до 8.0.30. Для сервер/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 создаёт новую.
6.0.x LTS	Для: Zabbix 6.2 Zabbix 6.4 Zabbix 7.0 Zabbix 7.2 Zabbix 7.4 Zabbix 8.0	Минимально требуемая версия PHP повышена с 7.2.5 до 7.4.0. Мониторинг сервисов был значительно переработан. Во время обновления необходимо создать детерминированные триггеры. Если для MySQL/MariaDB включено бинарное журналирование, для этого 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 минимально требуемые версии базы данных. Сервер/прокси не запустится при использовании устаревшей базы данных. Записи журнала аудита будут потеряны 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 минимально требуемые версии базы данных. Агрегированные элементы данных удалены как отдельный тип.

Обновление с версии	Читать полные примечания по обновлению	Наиболее важные изменения между версиями
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	Минимально требуемая версия PHP повышена с 7.2.0 до 7.2.5. Алгоритм хеширования паролей изменён с MD5 на bcrypt.
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. Минимально требуемая версия PHP повышена с 5.4.0 до 7.2.0. Повышены минимально требуемые версии базы данных. Изменён каталог файлов PHP 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 медиа 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 отправлять данные на обновлённый сервер. 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 plain text прекращена, заголовок теперь обязателен. Агенты Zabbix версии ниже 1.4 больше не поддерживаются. Параметр Server в конфигурации пассивного прокси теперь обязателен.

Обновление с версии	Читать полные примечания по 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 серверной базы данных прекращена для сервер/веб-интерфейс Zabbix. Вместо расширенных POSIX теперь поддерживаются Perl Compatible Regular Expressions (PCRE). Библиотеки `libpcre` и `libevent` обязательны для сервера Zabbix. Добавлены проверки кода завершения для пользовательских параметров, удалённых команд и элементов данных system.run[] без флага `nowait`, а также для скриптов, выполняемых сервером Zabbix. Для поддержки новой функциональности необходимо обновить Zabbix Java gateway.
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 history.
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	Минимально требуемая версия PHP повышена с 5.3.0 до 5.4.0. Необходимо указать параметр агента LogFile.

Обновление с версии	Читать полные примечания по 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	Мониторинг в распределённой среде на основе узлов удалён.
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	Минимально требуемая версия PHP повышена с 5.1.6 до 5.3.0. Для корректной работы сервера требуется MySQL с учётом регистра; для корректной работы сервера Zabbix с базой данных MySQL необходимы кодировка utf8 и сопоставление utf8_bin. См. скрипты создания базы данных. Требуется PHP-расширение `mysqli` вместо `mysql`.

Обновление из пакетов

Обзор

В этом разделе приведены шаги, необходимые для успешного обновления с использованием официальных пакетов RPM и DEB, предоставляемых Zabbix, для:

- Red Hat Enterprise Linux
- Debian/Ubuntu

Пакеты Zabbix из репозиториев ОС

Некоторые дистрибутивы ОС (в частности, дистрибутивы на базе Debian) предоставляют собственные пакеты Zabbix. Эти пакеты **не поддерживаются Zabbix** и могут быть устаревшими или не содержать последних возможностей и исправлений ошибок. Рекомендуется использовать только официальные пакеты из официального репозитория Zabbix.

Если вы обновляетесь с пакетов, предоставляемых дистрибутивами ОС (или они были установлены у вас ранее), выполните следующую процедуру для перехода на официальные пакеты Zabbix:

1. Удалите старые пакеты.
2. Проверьте наличие и удалите все оставшиеся файлы от старой установки.
3. Установите официальные пакеты Zabbix, следуя инструкциям по установке, предоставленным Zabbix.

Не выполняйте прямое обновление, так как это может привести к неработоспособной установке.

Debian/Ubuntu

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 Debian/Ubuntu.

Warning:
Перед обновлением, пожалуйста, ознакомьтесь с соответствующими примечаниями к обновлению и убедитесь, что ваша система соответствует требованиям для Zabbix 8.0.

Note:
Рекомендуется использовать два параллельных SSH-сеанса во время обновления: один для выполнения шагов обновления, а другой — для мониторинга журналов сервера/прокси. Например, выполните `tail -f zabbix_server.log` или `tail -f zabbix_proxy.log` во втором сеансе, чтобы просматривать последние записи журнала и возможные ошибки в режиме реального времени. Это может быть критически важно для производственных 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/apache2/conf-enabled/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 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 повторно в процессе обновления.

Затем установите последний пакет конфигурации репозитория, чтобы обеспечить совместимость с новейшими пакетами, а также включить последние исправления безопасности и ошибок.

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 приведенную выше ссылку на правильную из репозитория Zabbix. Однако обратите внимание, что пакеты для этих версий могут включать не все компоненты Zabbix, и для обновления этих компонентов из пакетов следует рассмотреть возможность обновления ОС. Список включенных компонентов см. в разделе Пакеты Zabbix.

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 приведенную выше ссылку на правильную из репозитория Zabbix. Однако обратите внимание, что пакеты для этих версий могут включать не все компоненты Zabbix, и для обновления этих компонентов из пакетов следует рассмотреть возможность обновления ОС. Список включенных компонентов см. в разделе Пакеты Zabbix.

Вы можете увидеть запрос о конфигурации 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 конфигурации репозитория Zabbix от сопровождающего пакета.

Затем обновите информацию о репозитории:

```
apt update
```

5 Обновление компонентов Zabbix

Чтобы обновить компоненты Zabbix, выполните:

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

- Если вы используете PostgreSQL, замените `mysql` на `pgsql` в команде.
- Если вы обновляете прокси, замените `server` на `proxy` в команде.
- Если вы обновляете Zabbix агент 2, замените `zabbix-agent` на `zabbix-agent2 zabbix-agent2-plugin-*` в команде.

Attention:
Обновление Zabbix агент 2 с помощью команды `apt install zabbix-agent2` может привести к ошибке. Для получения дополнительной информации см. *Известные проблемы*.

Вы можете увидеть 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 подходит для вашей ситуации. Например, введите D, чтобы сравнить текущую и новую конфигурацию, а затем решить, хотите ли вы установить версию, предоставленную сопровождающим пакета (Y или I).

Затем, чтобы обновить веб-интерфейс Zabbix с Apache и перезапустить Apache, выполните:

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

6 Просмотрите параметры конфигурации компонентов

Просмотрите соответствующие примечания по обновлению, чтобы проверить, требуются ли какие-либо изменения параметров конфигурации.

Сведения о новых необязательных параметрах см. 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 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, пожалуйста, ознакомьтесь с соответствующими примечаниями к обновлению и убедитесь, что ваша система соответствует требованиям для Zabbix 8.0.

Note:
Рекомендуется открыть два параллельных SSH-сеанса во время обновления: один для выполнения шагов обновления, а другой — для мониторинга журналов сервера/прокси. Например, во втором сеансе выполните `tail -f zabbix_server.log` или `tail -f zabbix_proxy.log`, чтобы просматривать последние записи журнала и возможные ошибки в режиме реального времени. Это может быть критически важно для производственных 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 продолжением обновления обновите текущий пакет репозитория до последней версии, чтобы обеспечить совместимость с новейшими пакетами, а также включить последние исправления безопасности и ошибок.

На **RHEL 10** выполните:

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

На **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 него дистрибутивов замените приведенную выше ссылку на правильную из репозитория Zabbix. Однако обратите внимание, что пакеты для этих версий могут не включать все компоненты Zabbix, и для обновления этих компонентов из пакетов следует рассмотреть возможность обновления вашей ОС. Список включенных 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 очищены.

См. также: Известные проблемы для обновления пакета конфигурации репозитория в RHEL.

5 Обновление компонентов Zabbix

Чтобы обновить компоненты Zabbix, выполните:

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

- Если вы используете PostgreSQL, замените `mysql` на `pgsql` в команде.
- Если вы обновляете прокси, замените `server` на `proxy` в команде.
- Если вы обновляете агент 2, замените `zabbix-agent` на `zabbix-agent2 zabbix-agent2-plugin-*` в команде.

Attention:
Обновление Zabbix агент 2 с помощью команды `dnf install zabbix-agent2` может привести к ошибке. Для получения дополнительной информации см. *Известные проблемы*.

Затем, чтобы обновить веб-интерфейс Zabbix с Apache и перезапустить Apache, выполните:

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

6 Просмотрите параметры конфигурации компонентов

Просмотрите соответствующие примечания к обновлению, чтобы проверить, требуется ли внести какие-либо изменения в параметры конфигурации.

Сведения о новых необязательных параметрах см. 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 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 Zabbix agent 2, замените `'zabbix-*'` на `'zabbix-agent2-*'` в команде.

Обновление из исходного кода

Обзор

В этом разделе приведены шаги, необходимые для успешного обновления с Zabbix **7.4**.x до Zabbix **8.0**.x с использованием официальных исходных кодов Zabbix.

Warning:
Перед обновлением обязательно ознакомьтесь с соответствующими **примечаниями к обновлению**!

Также рекомендуется ознакомиться с требованиями для 8.0.

Note:
Во время обновления может быть удобно использовать две параллельные SSH-сессии: в одной выполнять шаги обновления, а в другой отслеживать журналы сервера/прокси. Например, во второй SSH-сессии можно выполнить `tail -f zabbix_server.log` или `tail -f zabbix_proxy.log`, чтобы в реальном времени видеть последние записи журнала и возможные ошибки. Это может быть критически важно для рабочих сред.

Процесс обновления сервера

1. Останов сервера

Остановите Zabbix сервер, чтобы быть уверенным, что в базу данных не будет происходить запись новых данных.

2. Резервное копирование существующей базы данных Zabbix

Этот шаг очень важен. Убедитесь, что у вас имеется резервная копия вашей базы данных. Копия поможет, если процедура обновления закончится неудачно (отсутствие свободного места на диске, отключение питания, любая непредвиденная проблема).

3. Резервное копирование файлов конфигурации, PHP файлов и бинарных файлов Zabbix

Выполните резервное копирование бинарных файлов Zabbix, файлов конфигурации и папки с PHP файлами.

4 Установите новые бинарные файлы сервера

Используйте эти инструкции для компиляции сервера Zabbix из исходного кода.

5 Просмотрите параметры конфигурации сервера

Обязательно ознакомьтесь с примечаниями по обновлению, чтобы проверить, требуется ли внести какие-либо изменения в параметры конфигурации.

Сведения о новых необязательных параметрах см. на странице Что нового.

6. Запуск новых бинарных файлов Zabbix

Запустите новые бинарные файлы. Проверьте файлы журналов, чтобы удостовериться, что сервер запустился успешно.

Zabbix сервер автоматически обновит базу данных. При запуске Zabbix сервер сообщает текущую (обязательную и опциональную) и требуемую версии базы данных. Если текущая обязательная версия старше требуемой версии, Zabbix сервер автоматически выполнит требуемые патчи обновления базы данных. Начало и прогресс (в процентах) обновления базы данных записываются в файл журнала Zabbix сервера. Когда обновление завершится, в файл журнала запишется сообщение «database upgrade fully completed». В случае сбоя какого-либо из патчей обновления Zabbix сервер не запустится. Zabbix сервер также не запустится, если текущая обязательная версия более новая, чем требуемая. Zabbix сервер запустится только, 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
```

До запуска сервера:

- Убедитесь, что пользователь базы данных имеет достаточно прав (create table, drop table, create index, drop index).
- Убедитесь что у вас достаточно свободного дискового пространства.

7 Установка нового веб-интерфейса Zabbix

Минимально требуемая версия PHP — 8.2.0. При необходимости обновите её и следуйте инструкциям по установке.

8. Очистка cookies и кэша в веб-браузере

После обновления вам, возможно, потребуется очистить cookies веб-браузера и кэш веб-браузера, чтобы веб-интерфейс Zabbix работал должным образом.

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 Установите новые бинарные файлы прокси

Используйте эти инструкции, чтобы скомпилировать прокси Zabbix из исходного кода.

4 Просмотрите параметры конфигурации прокси

Обязательно ознакомьтесь с примечаниями по обновлению, чтобы проверить, требуются ли какие-либо изменения параметров конфигурации.

Сведения о новых необязательных параметрах см. на странице Что нового.

6. Запуск нового Zabbix прокси

Запустите новый Zabbix прокси. Проверьте файлы журналов, чтобы удостовериться, что прокси запустился успешно.

Zabbix прокси автоматически обновит базу данных. Обновление базы данных выполняется аналогично запуску Zabbix сервера.

Процесс обновления агента

Attention:
Обновление агентов не является обязательным. Вы можете обновить агенты только, если это требуется для доступа к новому функционалу.

Процедурой обновления, описанной в этом разделе, можно воспользоваться для обновления как Zabbix агента, так и Zabbix агента 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 бинарные файлы агента

Используйте эти инструкции для компиляции агента Zabbix из исходного кода.

В качестве альтернативы вы можете загрузить предварительно скомпилированные агенты Zabbix со страницы загрузки Zabbix.

4 Просмотрите параметры конфигурации агента

Обязательно ознакомьтесь с примечаниями по обновлению, чтобы проверить, требуются ли какие-либо изменения в параметрах конфигурации.

Сведения о новых необязательных параметрах см. на странице Что нового.

5. Запуск нового Zabbix агента

Запустите новый Zabbix агент. Проверьте файлы журналов, чтобы удостовериться, что агент запустился успешно.

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 мажорными версиями. Единственное отличие заключается в том, что при обновлении между минорными версиями изменения в базу данных не вносятся.

Обновление из контейнеров

Обзор

В этом разделе описаны шаги, необходимые для успешного обновления контейнеров Zabbix **8.0**.x.

Доступны отдельные наборы инструкций для обновления образов отдельных компонентов Zabbix и Docker compose files.

Warning:
Перед обновлением обязательно ознакомьтесь с соответствующими **примечаниями к 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 старых версий в процессе обновления потребуется создать детерминированные триггеры. Если для MySQL/MariaDB включено бинарное журналирование, для этого потребуются привилегии суперпользователя или установка переменной/параметра конфигурации *log_bin_trust_function_creators = 1*. Инструкции по установке этой переменной см. в разделе Скрипты создания базы данных.
Обратите внимание, что при выполнении из консоли переменная будет установлена только временно и будет сброшена после перезапуска Docker. В этом случае оставьте SQL-сервис запущенным, остановите только сервис 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 конфигурационном файле.

В зависимости от размера базы данных обновление до версии 8.0 может занять довольно много времени.

Обновление образа Zabbix

Приведенные ниже шаги можно использовать для обновления любого компонента Zabbix. Замените `zabbix-server-mysql` на имя образа нужного компонента.

1. Проверьте текущую версию образа:

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

2. Загрузите нужную версию образа, например:

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

zabbix/zabbix-server-mysql:alpine-8.0-latest загрузит последнюю выпущенную минорную версию сервера Zabbix 8.0 с поддержкой MySQL на базе Alpine Linux. Замените его на нужное вам сочетание имени Docker-репозитория и тегов. Список доступных вариантов смотрите в разделе Установка из контейнеров.

3. Остановите контейнер:

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

4. Удалите контейнер:

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

5. Запустите обновленный контейнер, выполнив команду docker run с дополнительными аргументами для указания необходимых переменных окружения и/или точек монтирования.

Примеры конфигурации

Сервер Zabbix с 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 MYSQL_ROOT_PASSWORD="root_pwd" \
    -e ZBX_JAVAGATEWAY="zabbix-java-gateway" \
    --network=zabbix-net \
    -p 10051:10051 \
    --restart unless-stopped \
    -d zabbix/zabbix-server-mysql:alpine-8.0-latest
```

Сервер Zabbix с PostgreSQL:

```
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 \
    -d zabbix/zabbix-server-pgsql:alpine-8.0-latest
```

Сервер MySQL:

```
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 \
```

Дополнительные примеры конфигурации, включая примеры для других компонентов Zabbix, доступны на странице Установка из контейнеров.

6. Проверьте обновление:

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

Файлы compose

Следуйте инструкциям по обновлению в этом разделе, если вы установили Zabbix с использованием файла compose.

1. Проверьте текущую версию образа:

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

2. Получите последние обновления из GitHub репозитория и переключитесь на требуемую ветку:

```
git pull
git checkout 8.0
```

3. Запустите компоненты Zabbix, используя новый файл compose:

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

4. Проверьте обновление:

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

См. Установка из контейнеров для получения более подробной информации, включая списки поддерживаемых переменных окружения и точек монтирования томов.

Проблемы обновления

Экранирование и связанные с ним аспекты Превышен лимит символов для параметров функций после обновления

В Zabbix 7.0 в строковые параметры функций истории было добавлено корректное экранирование обратных слешей.

Поскольку при обновлении с версий Zabbix до 7.0 добавляются дополнительные обратные слеши, это приводит к увеличению длины параметров, что может вызвать нарушение работы функций триггеров, если длина параметра превышает максимальный размер данных в 255 символов.

Чтобы показать, какие проблемы могут возникнуть при обновлении базы данных, каково их влияние, как их распознать и исправить, рассмотрим пример конфигурации с 3 элементами данных и 2 триггерами.

Настройка

1. Версия Zabbix 6.0.
2. Файл `/tmp/ONE`, содержащий 114 обратных слэшей с символом 'a' в конце:

```
\\\
```

7. Calc_232 возвращает 0, и Trig_232 **не** срабатывает.

Обновление

1. Выполните обновление до 8.0.0 и проверьте журналы на наличие предупреждения:

```
2485502:20250228:115442.236 DBpatch_6050165(): cannot save in DB function parameter: resulting size 477 is
functionid:33792 function:'find'
used on host: 'Zabbix server'
  in trigger: 'TRIG_228'.
Current parameter value:
'$,1m:now,"regexp","\\\
```

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

Trig_232:

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

Теперь оба триггера работают так же, как и раньше, — только Trig_228 **срабатывает**.

С вычисляемыми элементами данных проблем нет, однако их также можно обновить для единообразия:

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})
```

5 Примечания по обновлению для Zabbix 8.0

Эти примечания предназначены для обновления с Zabbix 7.4.x до Zabbix 8.0.0.

Все примечания сгруппированы по следующим разделам:

- **Критические изменения** — изменения, которые могут нарушить работу существующих установок, а также другая важная информация, связанная с процессом обновления
- **Прочее** — вся остальная информация, описывающая изменения в функциональности Zabbix

См. также:

- Процедура обновления для получения всей необходимой информации об обновлении с версий до Zabbix 8.0.0;
- Обновление HA-кластера с инструкциями по обновлению серверов в кластере **high-availability** (HA).

Критические изменения Версии баз данных

Минимальные требуемые версии баз данных были повышены:

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

Плагины

Плагин Ceph для Zabbix агент 2 теперь является загружаемым плагином и требует дополнительных шагов установки. Подробности см. в readme плагина Ceph.

Схема `tcp://` объявлена устаревшей в плагине MongoDB, но сохранена для обратной совместимости с существующими конфигурациями.

Минимально требуемая версия PHP

Минимально требуемая версия PHP была повышена с 8.0.0 до 8.2.0.

Расширенный список UnsafeUserParameters

Символ % был добавлен в список `UnsafeUserParameters` для агента Zabbix и агента Zabbix 2.

Другое Удалены устаревшие макросы

Поддержка следующих встроенных макросов теперь прекращена:

Удалено	Используйте вместо этого
{ACK.DATE}	{EVENT.UPDATE.DATE}
{ACK.MESSAGE}	{EVENT.UPDATE.MESSAGE}
{ACK.TIME}	{EVENT.UPDATE.TIME}
{EVENT.ACK.HISTORY}	{EVENT.UPDATE.HISTORY}
{HOSTNAME<1-9>}	{HOST.HOST}
{IPADDRESS<1-9>}	{HOST.IP}

Удалено	Используйте вместо этого
{PROFILE.*}	{INVENTORY.*}
{TRIGGER.COMMENT}	{TRIGGER.DESCRIPTION}
{TRIGGER.KEY}	{ITEM.KEY}
{STATUS}	{TRIGGER.STATUS}
{USER.ALIAS}	{USER.USERNAME}

Удалена проверка JSON и XML в элементах данных HTTP агент

Проверка JSON и XML была удалена из поля *Тело запроса* в элементах данных HTTP агент. Это поле больше не проверяет, является ли значение корректным XML или JSON, а выбор *XML data* в качестве *Типа тела запроса* больше не требует библиотеку libxml2.

Скрытие отключенных ссылок на контролируемых узлах сети

Отключенные ссылки *Graphs*, *Dashboards* и *Web* больше не отображаются в списке контролируемых узлов сети.

Тип данных JSON

Теперь Zabbix поддерживает JSON в качестве типа данных для значений элементов данных. Если вы используете TimescaleDB, новую гипертаблицу `history_json` (используемую для хранения значений JSON) необходимо настроить вручную. Если вы используете Elasticsearch, значение по умолчанию параметра конфигурации сервера Zabbix `HistoryStorageTypes` теперь включает `json`.

Удалена радиокнопка в форме нового фильтра тегов

Радиокнопка, требовавшая выбора между *Все теги* и *Список тегов* при создании нового фильтра тегов, была удалена.

Значения элементов данных, отображаемые как HTML с использованием iframe

Данные элемента данных в виджете История элементов данных, при отображении в виде текста в формате HTML, теперь изолируются в iframe.

Template compatibility with Host Wizard

All out-of-the-box templates are now compatible with the Host Wizard. To upgrade them, see Template upgrade.

Изменения шаблонов

На этой странице перечислены все изменения стандартных шаблонов, поставляемых с Zabbix.

При обновлении Zabbix существующие шаблоны не обновляются автоматически, чтобы избежать перезаписи пользовательских изменений. Чтобы обновить шаблоны или добавить новые, смотрите Обновление шаблонов.

> **Note:**
> Обратите внимание, что начиная с Zabbix 6.0 все шаблоны используют обновленный формат, что может повлиять на импорт шаблонов версий до 6.0. Дополнительную информацию смотрите в разделе Изменения шаблонов в 6.0.

Изменения в 8.0.0

Примечания по обновлению для Zabbix 8.0.x

На этой странице собраны примечания по обновлению для минорных выпусков основной версии Zabbix.

См. также примечания по обновлению для основной версии.

Примечания к обновлению для 8.0.1 Эта версия еще не выпущена.

Известные проблемы

См. также: Проблемы компиляции.

Обновление

Настройка SQL mode для успешного обновления

В MySQL/MariaDB в параметре `sql_mode` должен быть установлен режим "STRICT_TRANS_TABLES". Если он отсутствует, обновление базы данных Zabbix завершится ошибкой (см. также ZBX-19435).

Обновление с MariaDB 10.2.1 и более ранними версиями

Обновление Zabbix может завершиться неудачей, если таблицы базы данных были созданы в MariaDB 10.2.1 и более ранних версиях, поскольку в этих версиях формат строк по умолчанию — compact. Это можно исправить, изменив формат строк на dynamic (см. также ZBX-17690).

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 включён, при установке пакетов Zabbix могут быть установлены версии из 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 содержат слово `release` в строке 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 в наличии доступа 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, что 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 регулируется подписками, которые в случае сред UBI передаются путем монтирования каталогов конфигурации репозиториев и секретов узла RHEL в пространство имен файловой системы контейнера.

Для получения дополнительной информации см. 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.

Timescale DB: высокое потребление памяти при большом количестве партиций

Версии PostgreSQL 9.6-12 используют слишком много памяти при обновлении таблиц с большим количеством партиций. Эта проблема проявляется, когда Zabbix обновляет тренды в системах с TimescaleDB, если тренды разделены на относительно небольшие чанки (например, по 1 дню). Это приводит к наличию сотен чанков в таблицах трендов при настройках housekeeper по умолчанию — в условиях, когда PostgreSQL с высокой вероятностью может исчерпать память.

Проблема была решена начиная с Zabbix 5.0.1 для новых установок с TimescaleDB, однако если TimescaleDB был настроен с Zabbix до этого, ознакомьтесь с примечаниями по миграции в ZBX-16347.

Timescale DB 2.5.0: политика сжатия может завершаться с ошибкой на таблицах, содержащих целые числа

Эта проблема проявляется при использовании TimescaleDB 2.5.0/2.5.1. Она была решена начиная с TimescaleDB 2.5.2.

Для получения дополнительной информации см. TimescaleDB Issue #3773.

TLS-подключение к базе данных с MariaDB

TLS-подключение к базе данных не поддерживается с параметром 'verify_ca' для параметра DBTLSConnect parameter, если используется MariaDB.

Возможные взаимные блокировки с MySQL/MariaDB

При работе под высокой нагрузкой и при участии более чем одного LLD worker возможно возникновение взаимной блокировки, вызванной ошибкой InnoDB, связанной со стратегией блокировки строк (см. ошибку в исходном проекте). Эта ошибка была исправлена в MySQL, начиная с версии 8.0.29, но не в MariaDB. Подробнее см. ZBX-21506.

Глобальная корреляция событий

События могут коррелироваться некорректно, если интервал времени между первым и вторым событием очень мал, то есть полсекунды и меньше.

NetBSD 8.0 и новее

Различные процессы Zabbix могут случайным образом аварийно завершаться при запуске в версиях NetBSD 8.X и 9.X. Это связано со слишком маленьким размером стека по умолчанию (4 МБ), который необходимо увеличить, выполнив:

```
ulimit -s 10240
```

Для получения дополнительной информации см. соответствующий отчёт о проблеме: ZBX-18275.

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.

Проверки IPMI

Проверки IPMI не будут работать со стандартным пакетом библиотеки OpenIPMI в Debian до версии 9 (stretch) и Ubuntu до версии 16.04 (xenial). Чтобы исправить это, перекомпилируйте библиотеку OpenIPMI с включенной поддержкой OpenSSL, как описано в ZBX-6139.

IPMI — недоверенные узлы сети могут привести к сбою OpenIPMI

В библиотеке OpenIPMI, используемой Zabbix для опроса данных IPMI, существует ошибка, которая может быть вызвана специально сформированными ответами от недоверенного устройства.
Недоверенное IPMI-устройство может отправить специально сформированные данные, которые приведут к сбою библиотеки OpenIPMI, что, в свою очередь, может вызвать завершение процесса сервера Zabbix, выполняющего опрос IPMI.

SSH-проверки

- Некоторые дистрибутивы Linux, такие как Debian и Ubuntu, не поддерживают зашифрованные закрытые ключи (с парольной фразой), если библиотека libssh2 установлена из пакетов. Подробнее см. ZBX-4850.
- При использовании libssh 0.9.x в некоторых дистрибутивах Linux с OpenSSH 8 SSH-проверки могут время от времени сообщать об ошибке "Cannot read data from SSH server". Это вызвано проблемой в libssh (более подробный отчет). Ожидается, что ошибка была исправлена в стабильном выпуске libssh 0.9.5. Подробнее см. ZBX-17756.
- Использование символа канала "|" в SSH-скрипте может привести к ошибке "Cannot read data from SSH server". В этом случае рекомендуется обновить версию библиотеки libssh. Подробнее см. ZBX-21337.

Проверки ODBC

- Драйвер MySQL unixODBC не следует использовать с сервером Zabbix или прокси Zabbix, скомпилированными с библиотекой коннектора MariaDB, и наоборот; по возможности также лучше избегать использования того же коннектора в качестве драйвера из-за ошибки в исходном проекте. Рекомендуемая конфигурация:

Коннектор PostgreSQL, SQLite или Oracle → драйвер MariaDB или MySQL unixODBC
Коннектор MariaDB → драйвер MariaDB unixODBC
Коннектор MySQL → драйвер MySQL unixODBC

Дополнительную информацию и доступные обходные решения смотрите в ZBX-7665.

- XML-данные, запрашиваемые из Microsoft SQL Server, могут усекаться 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 сбою сервера Zabbix. См. также: ZBX-18402, ZBX-20803.
- При использовании драйвера FreeTDS UnixODBC необходимо добавлять оператор 'SET NOCOUNT ON' в начало SQL-запроса (например, `SET NOCOUNT ON DECLARE @strsql NVARCHAR(max) SET @strsql = ....`). В противном случае элемент данных монитора базы данных в Zabbix не сможет получить информацию и вернет ошибку "SQL query returned empty result".
  Дополнительную информацию смотрите в ZBX-19917.

Некорректный параметр метода запроса в элементах данных

Параметр метода запроса, используемый только в HTTP-проверках, может быть некорректно установлен в значение '1', которое является нестандартным для всех элементов данных, в результате обновления с версии Zabbix до 4.0. Подробную информацию о том, как исправить эту ситуацию, см. в ZBX-19308.

Веб-мониторинг и HTTP агент

Сервер Zabbix допускает утечку памяти в некоторых дистрибутивах Linux из-за ошибки в вышестоящем проекте, когда в веб-сценариях или HTTP агенте включена опция "SSL verify peer". Дополнительную информацию и доступные обходные пути смотрите в ZBX-10486.

Простые проверки

В версиях **fping** ранее v3.10 присутствует ошибка, из-за которой некорректно обрабатываются дублирующиеся пакеты echo reply. Это может приводить к неожиданным результатам для элементов данных `icmpping`, `icmppingloss`, `icmppingsec`. Рекомендуется использовать последнюю версию **fping**. Дополнительные сведения см. в ZBX-11726.

Ошибки при выполнении fping в rootless-контейнерах

Когда контейнеры работают в rootless-режиме или в среде с определёнными ограничениями, при выполнении ICMP-проверок вы можете столкнуться с ошибками, связанными с выполнением fping, например `fping: Operation not permitted`, либо со всеми потерянными пакетами для всех ресурсов.

Чтобы исправить эту проблему, добавьте `--cap-add=net_raw` к командам "docker run" или "podman run".

Кроме того, выполнение fping в средах без root-прав может потребовать изменения sysctl, например:

```
sudo sysctl -w "net.ipv4.ping_group_range=0 1995"
```

где "1995" — это GID zabbix. Подробнее см. ZBX-22833.

SNMP-проверки

Если используется операционная система OpenBSD, ошибка use-after-free в библиотеке Net-SNMP до версии 5.7.3 может привести к сбою сервера Zabbix, если в файле конфигурации сервера Zabbix задан параметр SourceIP. В качестве обходного решения не задавайте параметр SourceIP. Та же проблема также относится к Linux, однако она не приводит к остановке работы сервера Zabbix. Локальный патч для пакета net-snmp в OpenBSD был применён и будет выпущен вместе с OpenBSD 6.3.

Всплески данных SNMP

Наблюдались всплески данных SNMP, которые могут быть связаны с определёнными физическими факторами, такими как скачки напряжения в электросети. Подробнее см. ZBX-14318.

SNMP traps

Пакет "net-snmp-perl", необходимый для SNMP traps, был удален в RHEL 8.0-8.2; повторно добавлен в RHEL 8.3.

Поэтому, если вы используете RHEL 8.0-8.2, наилучшим решением будет обновление до RHEL 8.3.

Дополнительную информацию см. в ZBX-17192.

Сбой процесса alerter в RHEL 7

В RHEL 7 были зафиксированы случаи сбоя процесса alerter сервера Zabbix. Подробности см. в ZBX-10461.

Обновление Zabbix агент 2 (6.0.5 или более ранней версии)

При обновлении Zabbix агент 2 (версии 6.0.5 или более ранней) из пакетов может возникнуть ошибка конфликта файлов, связанная с плагином. Чтобы исправить ошибку, создайте резервную копию конфигурации агент 2 (при необходимости), удалите агент 2 и установите его заново.

В системах на базе RHEL выполните:

```
dnf remove zabbix-agent2
dnf install zabbix-agent2
```

В системах на базе Debian выполните:

```
apt remove zabbix-agent2
apt install zabbix-agent2
```

Для получения дополнительной информации см. ZBX-23250.

Переключение локалей веб-интерфейса

Было замечено, что локали веб-интерфейса могут переключаться без видимой логики, то есть некоторые страницы (или части страниц) отображаются на одном языке, тогда как другие страницы (или части страниц) — на другом языке. Обычно проблема может возникать, когда есть несколько пользователей, часть из которых использует одну локаль, а другие — другую.

Известный обходной путь — отключить многопоточность в PHP и Apache.

Проблема связана с тем, как работает установка локали в PHP: информация о локали хранится на уровне процесса, а не потока. Поэтому в многопоточной среде, когда несколько проектов выполняются в рамках одного и того же процесса Apache, возможно, что локаль будет изменена в другом потоке, и это повлияет на обработку данных в потоке Zabbix.

Для получения дополнительной информации см. связанные отчёты о проблемах:

- ZBX-10911 (Проблема с переключением локалей веб-интерфейса)
- ZBX-16297 (Проблема с обработкой чисел на графиках при использовании функции `bcdiv` из набора математических функций BC Math)

Графики

Проблемы с графиками (classic)

Если вы сталкиваетесь с проблемами в классических графиках, 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.

Переход на летнее время

Изменения, связанные с переходом на летнее время (DST), приводят к некорректному отображению подписей оси X (дублирование дат, отсутствие дат и т. д.).

Агрегация sum

При использовании агрегации sum в графике для периода менее одного часа графики отображают некорректные (умноженные) значения, когда данные поступают из trends.

Наложение текста

Для некоторых языков веб-интерфейса (например, японского) локальные шрифты могут вызывать наложение текста в легенде графика. Чтобы избежать этого, используйте PHP GD extension версии 2.3.0 (или более поздней).

Мониторинг лог-файлов

Элементы данных `log[]` и `logrt[]` повторно перечитывают лог-файл с начала, если файловая система заполнена на 100% и в лог-файл продолжается запись (подробнее см. ZBX-10884).

Медленные запросы MySQL

Сервер Zabbix генерирует медленные запросы `SELECT` в случае отсутствия значений для элементов данных. Эта проблема известна для версий MySQL 5.6/5.7 (подробное обсуждение см. в ZBX-10652) и в некоторых случаях может также возникать в более поздних версиях MySQL. В качестве обходного решения можно отключить оптимизатор `index_condition_pushdown` или `prefer_ordering_index` в MySQL. Однако обратите внимание, что это обходное решение может не устранить все проблемы, связанные с медленными запросами.

Постоянные настройки фильтра из ссылок

При открытии ссылки на страницу веб-интерфейса Zabbix, содержащей настройки фильтра, включая селектор времени, фильтр автоматически сохраняется в базе данных для пользователя, заменяя ранее сохранённый фильтр и/или настройки селектора времени для этой страницы. Эти настройки остаются активными, пока пользователь не обновит или не сбросит их вручную.

Проблема с IPv6-адресом в SNMPv3 трапах

Из-за ошибки в net-snmp IPv6-адрес может отображаться некорректно при использовании SNMPv3 в SNMP трапах. Для получения более подробной информации и возможного обходного решения см. ZBX-14541.

Усечённый длинный IPv6 IP-адрес в информации о неудачном входе

Сообщение о неудачной попытке входа будет отображать только первые 39 символов сохранённого IP-адреса, так как это ограничение количества символов в поле базы данных. Это означает, что IPv6 IP-адреса длиннее 39 символов будут показаны не полностью.

Проверки агента Zabbix в Windows

Несуществующие DNS-записи в параметре `Server` файла конфигурации агента Zabbix (zabbix_agentd.conf) могут увеличить время отклика агента Zabbix в Windows. Это происходит потому, что служба 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

Существует несколько известных проблем с экспортом/импортом YAML:

- Сообщения об ошибках не переводятся;
- Корректный JSON с расширением файла .yaml иногда не удаётся импортировать;
- Даты в удобочитаемом формате без кавычек автоматически преобразуются в метки времени Unix.

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 настройкой в юните /usr/lib/systemd/system/php-fpm.service, которая не позволяет Zabbix записывать данные в /etc. (введено в PHP 7.4).

Доступны два варианта обходного решения:

- Установите для параметра ProtectSystem значение 'true' вместо 'full' в systemd-юните php-fpm.
- 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 достигал Zabbix. Это может вызывать проблемы с аутентификацией при использовании Zabbix API или служб единого входа (SSO), таких как SAML с Okta.

Чтобы решить эту проблему, обновите конфигурацию вашего веб-сервера.

Для **Apache**, если вы используете его как обратный прокси (настройка без CGI), 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 использованием 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.

Дополнительные сведения см. в:

- ZBX-22952
- Apache 2.4 + PHP-FPM and Authorization headers
- Директивы SetEnvIfNoCase и CGIPassAuth
- NGINX Reverse Proxy

Chromium для веб-службы Zabbix в Ubuntu 20

Хотя в большинстве случаев веб-служба Zabbix может работать с Chromium, в Ubuntu 20.04 использование Chromium вызывает следующую ошибку:

```
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, потому что `/var/lib/zabbix` используется как домашний каталог пользователя 'zabbix'.

Пользовательские коды ошибок MySQL

Когда Zabbix обнаруживает, что внутренняя база данных недоступна, он отправляет уведомление и продолжает попытки подключения. Для некоторых СУБД распознаются определённые коды ошибок. В MySQL к таким распознаваемым кодам ошибок относятся:

- 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, при использовании Zabbix с установленной в Azure MySQL в журналах Zabbix может появляться общее сообщение об ошибке *[9002] Some errors occurred*. Это сообщение отправляется базой данных на сервер или прокси Zabbix. 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 (всех версий), 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 затрагивает шаблон *^[\w-\.]*. Чтобы снова сделать это регулярное выражение допустимым без изменения семантики, измените выражение на *^[-\w\.]* . Это происходит потому, 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 со старой версии Zabbix с NGINX и во время обновления не переключились на новый файл конфигурации NGINX.

Чтобы устранить проблему, вы можете удалить старый файл конфигурации, использовать файл конфигурации из пакета текущей версии и заново настроить его, как описано в инструкциях по загрузке в разделе *e. Configure PHP for Zabbix frontend*.

Кроме того, 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, однако присваивания необъявленным идентификаторам (например, `secret = value`) создают неявные глобальные переменные, которые могут сохраняться после завершения текущего выполнения. Хранение конфиденциальных данных (токенов, паролей и т. д.) в неявных глобальных переменных повышает риск их 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 секретные данные к глобальным объектам (например, `globalThis` или `window`). Поддерживаемого способа переопределить встроенные глобальные объекты из предобработки не существует.

Безопасный пример:

```
var apiToken = payload.token;
var count = 1;
return JSON.stringify({ token: apiToken, calls: count });
```

Группы процессоров в Windows

Согласно документации Microsoft, системы с менее чем 64 логическими процессорами всегда имеют одну группу процессоров — Group 0. Однако пользователи Zabbix сообщали о редкой ошибке ZBX-20260, когда в системах с 64 или менее логическими процессорами присутствуют две группы процессоров. Это приводило к тому, что счётчики производительности "\Processor(n)" были доступны только для одной из двух групп процессоров. Фактическая первопричина этой ошибки неизвестна. Однако похожий случай описан на stackoverflow.com, и его первопричина заключалась во взаимодействии между BIOS и Windows.

Ограничения фильтрации с сопоставлениями utf8mb4

Фильтры (например, в *Сбор данных* > *Обслуживание*) могут работать некорректно при применении к сущностям, содержащим определенные символы Unicode (например, 𝓀, 𝔞). 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 устранить это ограничение, пользователи могут изменить сопоставление столбцов базы данных на альтернативные варианты, такие как utf8mb4_0900_bin, utf8mb4_0900_ai_ci или utf8mb4_unicode_520_ci. Однако обратите внимание, что изменение сопоставления может вызвать неожиданное поведение при обработке пробелов, а также при сортировке и фильтрации других символов.

Дополнительную информацию об изменении сопоставлений см. в документации MySQL или документации MariaDB. Подробные сведения о различиях между сопоставлениями см. в разделе Unicode Character Sets документации MySQL.

Доступ к элементам пользовательского интерфейса с MariaDB 10.5.1–10.5.9

При доступе к веб-интерфейсу Zabbix с ролью, отличной от Super Admin, может появиться сообщение: «Произошла системная ошибка. Пожалуйста, свяжитесь с администратором Zabbix.». Эта проблема затрагивает установки, использующие версии MariaDB с 10.5.1 по 10.5.9.

Чтобы избежать этой проблемы, обновите MariaDB до версии выше 10.5.9. Подробнее см. ZBX-25746.

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 исходного кода укажите 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 указывают tcmalloc, как отслеживать использование памяти и формировать отчёты:

```
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. Вызовите дамп профиля, отправив сигнал 5 целевому процессу. Замените 1234 на фактический идентификатор процесса (PID):

```
kill -5 1234
```

4. Выведите созданный 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 для связанных отчётов о проблемах.
- GCC Option Summary о параметрах компиляции (`-std=gnu99`, `-g`, `-O0` и т. д.).
- Документацию Gperftools Heap Profiler о переменных окружения для профилирования tcmalloc.

MySQL 8.0 Group Replication в режиме multi-primary

При использовании MySQL 8.0 Group Replication в режиме multi-primary вы можете столкнуться с ошибкой во время фиксации транзакций, похожей 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, что эта ошибка вызвана проблемами с операциями отката, связанными с ограничениями внешних ключей.

См. также:

- ZBX-26060 для связанного отчёта о проблеме.
- MySQL Bug #96758 "Rollbacks with Foreign Keys on single node" для информации о проблеме в исходном проекте.

Требования

Оборудование

Память

Zabbix требуется как физическая, так и дисковая память. Объем необходимой дисковой памяти, очевидно, зависит от количества узлов сети и параметров, которые отслеживаются. Если вы планируете хранить длительную историю отслеживаемых параметров, вам следует предусмотреть как минимум несколько гигабайт, чтобы иметь достаточно места для хранения истории в базе данных. Каждому процессу демона Zabbix требуется несколько подключений к серверу базы данных. Объем памяти, выделяемой для подключения, зависит от конфигурации механизма базы данных.

> **Note:**
> Чем больше у вас физической памяти, тем быстрее работает база данных (и, следовательно, Zabbix).

ЦП

Zabbix, и в особенности база данных Zabbix, могут требовать значительных ресурсов ЦП в зависимости от количества контролируемых параметров и выбранного механизма базы данных.

Другое оборудование

Для использования поддержки SMS-уведомлений в Zabbix требуются порт последовательной связи и последовательный GSM-модем. Также подойдет преобразователь USB-to-serial.

Примеры аппаратной конфигурации

В таблице приведены примеры аппаратной конфигурации для платформы **Linux/BSD/Unix**.

Это примеры размеров и аппаратной конфигурации, с которых можно начать. Каждая установка Zabbix уникальна. Обязательно выполните тестирование производительности вашей системы Zabbix в промежуточной или тестовой среде, чтобы полностью понять свои требования перед развертыванием установки Zabbix в промышленной среде.

Размер установки	Отслеживаемые метрики[1]	Ядра CPU/vCPU	Память (GiB)	База данных	Amazon EC2[2]
Маленькая	1 000	2	8	MySQL Server, Percona Server, MariaDB Server, PostgreSQL	m6i.large/m6g.large
Средняя	10 000	4	16	MySQL Server, Percona Server, MariaDB Server, PostgreSQL	m6i.xlarge/m6g.xlarge
Большая	100 000	16	64	MySQL Server, Percona Server, MariaDB Server, PostgreSQL	m6i.4xlarge/m6g.4xlarge

Размер установки	Отслеживаемые метрики[1]	Ядра CPU/vCPU	Память (GiB)	База данных	Amazon EC2[2]
Очень большая	1 000 000	32	96	MySQL Server, Percona Server, MariaDB Server, PostgreSQL	m6i.8xlarge/m6g.8xlarge

[1] 1 метрика = 1 элемент данных + 1 триггер + 1 график
 [2] Пример с инстансами Amazon EC2 общего назначения с использованием архитектуры ARM64 или x86_64; при оценке и тестировании установки Zabbix перед развертыванием в промышленной среде следует выбрать подходящий тип инстанса, например, оптимизированный для вычислений/памяти/хранилища

Note:
Фактическая конфигурация в значительной степени зависит от количества активных элементов данных и интервалов обновления (подробности см. в разделе размер базы данных на этой странице). Для крупных установок настоятельно рекомендуется запускать базу данных на отдельном сервере.

Поддерживаемые платформы

Из-за требований безопасности и критически важного характера сервера мониторинга UNIX является единственной операционной системой, которая может стабильно обеспечивать необходимую производительность, отказоустойчивость и устойчивость к сбоям. Zabbix работает на ведущих версиях, представленных на рынке.

Компоненты Zabbix доступны и протестированы для следующих платформ:

Платформа	Сервер	Агент	Агент 2	Комментарии
Linux	x	x	x	
Windows	-	x	x	Агент Zabbix поддерживается на всех настольных и серверных версиях, начиная с Windows XP (64-bit)/Server 2003. Агент Zabbix 2 поддерживается на всех настольных и серверных версиях, начиная с Windows 10 (32-bit)/Server 2016, так как он компилируется только с поддерживаемой версией Go для предотвращения критических уязвимостей безопасности. Начиная с Go 1.21, минимально требуемые версии Windows были повышены, поэтому Windows 10/Server 2016 является минимальной версией для агента Zabbix 2.
macOS	x	x	-	
IBM AIX	x	x	-	Агент Zabbix не работает 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:
Сервер/агент Zabbix может работать и на других Unix-подобных операционных системах.

Attention:
Zabbix отключает core dump, если скомпилирован с поддержкой шифрования, и не запускается, если система не позволяет отключить core dump.

Необходимое программное обеспечение

Zabbix построен на основе современных веб-серверов, ведущих систем управления базами данных и языка сценариев PHP.

Стороннее внешнее программное обеспечение

Если указано, что программное обеспечение/библиотека является обязательным, оно строго необходимо. Необязательные компоненты требуются для поддержки некоторых специфических функций.

Программное обеспечение	Статус обязательности	Поддерживаемые версии	Комментарии
MySQL/Percona	Одно из	8.4.0-9.5.X	Требуется, если MySQL (или Percona) используется в качестве внутренней базы данных Zabbix. Требуется движок InnoDB. Мы рекомендуем использовать библиотеку C API (libmysqlclient) для сборки сервера/прокси.
MariaDB		10.11.00-12.0.X	Требуется движок InnoDB. Рекомендуемая версия — 11.4. Мы рекомендуем использовать библиотеку MariaDB Connector/C для сборки сервера/прокси. См. также: Возможные взаимные блокировки с MariaDB и Доступ к элементам пользовательского интерфейса с MariaDB 10.5.1–10.5.9.
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* (по умолчанию 4 МБ), чтобы объём памяти, используемый базой данных для конкретной операции, был достаточным, а выполнение запросов не занимало слишком много времени.
TimescaleDB for PostgreSQL		2.20.X-2.25.X	Требуется, если TimescaleDB используется как расширение базы данных PostgreSQL. Убедитесь, что установлена редакция TimescaleDB Community Edition, поддерживающая сжатие. Обратите внимание, что PostgreSQL 15 поддерживается, начиная с TimescaleDB 2.10. Вы также можете обратиться к документации TimescaleDB за подробной информацией о совместимости версий PostgreSQL и TimescaleDB.
SQLite	Необязательно	3.3.5-3.34.X	SQLite поддерживается только с прокси Zabbix. Требуется, если SQLite используется в качестве базы данных прокси Zabbix.
Elasticsearch		7.X	Elasticsearch поддерживается только с серверами Zabbix и исключительно для хранения исторических данных. Поддержка Elasticsearch в настоящее время является экспериментальной. См. также требуемое программное обеспечение для сервера/прокси.
smartmontools		7.1 или новее	Требуется для Zabbix агента 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` как часть q applets требуются для плагина system.sw.packages в Gentoo Linux.

Note:
Хотя Zabbix может работать с базами данных, доступными в операционных системах, для наилучшего результата мы рекомендуем использовать базы данных, установленные из официальных репозиториев разработчиков баз данных.

Веб-интерфейс

Минимальная поддерживаемая ширина экрана для веб-интерфейса Zabbix составляет 1200px.

Если указано, что компонент обязателен, то соответствующее программное обеспечение/библиотека строго необходимы. Необязательные компоненты требуются для поддержки некоторых специфических функций.

Software	Mandatory status	Supported versions	Comments
PHP	Yes	8.2.0 - 8.4.X	
Apache	One of	2.4 или новее	
Nginx		1.20 или новее	
MySQL	One of	См. Стороннее внешнее сопутствующее программное обеспечение	
PostgreSQL			
PHP extensions			
mysqli	Yes		Требуется, если MySQL используется в качестве внутренней базы данных Zabbix.
pgsql			Требуется, если PostgreSQL используется в качестве внутренней базы данных 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 новее	php-gd (если поставляется дистрибьютором как отдельный пакет); расширение PHP GD должно поддерживать изображения PNG (*--with-png-dir*), изображения JPEG (*--with-jpeg-dir*) и FreeType 2 (*--with-freetype-dir*). Версия 2.3.0 или новее может потребоваться, чтобы избежать возможного наложения текста на графиках для некоторых языков веб-интерфейса.
libxml		2.6.15 или новее	php-xml (если поставляется дистрибьютором как отдельный пакет)
xmlwriter			php-xmlwriter (если поставляется дистрибьютором как отдельный пакет)
xmlreader			php-xmlreader (если поставляется дистрибьютором как отдельный пакет)
ctype			php-ctype (*--enable-ctype*)
session			php-session (если поставляется дистрибьютором как отдельный пакет)
ldap	No		php-ldap; требуется для LDAP-аутентификации.
openssl			php-openssl; требуется для SAML-аутентификации.
gettext			php-gettext (*--with-gettext*); требуется для переводов.
cURL		7.19.4 или новее	php-curl; требуется для Duo Universal Prompt MFA и SMTP-аутентификации.

Сторонние библиотеки веб-интерфейса, поставляемые вместе с Zabbix:

Library	Mandatory status	Supplied version	Comments
jQuery JavaScript Library	Yes	3.6.0	Библиотека JavaScript, упрощающая процесс кросс-браузерной разработки.
jQuery UI		1.12.1	Набор взаимодействий пользовательского интерфейса, эффектов, виджетов и тем, построенный поверх jQuery.
SAML PHP Toolkit		4.3.1	Набор инструментов PHP, добавляющий поддержку аутентификации SAML 2.0 для входа в Zabbix.
Symfony Yaml Component		5.1.0	Добавляет поддержку экспорта и импорта элементов конфигурации Zabbix в формате YAML.

Note:
Zabbix также может работать с предыдущими версиями Apache, MySQL и PostgreSQL.

Attention:
Для шрифтов, отличных от стандартного DejaVu, может потребоваться PHP-функция imagerotate. Если она отсутствует, эти шрифты могут отображаться некорректно при показе графика. Эта функция доступна только в том случае, если PHP скомпилирован со встроенным GD, что не относится к Debian и другим дистрибутивам.

Сторонние библиотеки, используемые для написания и отладки кода веб-интерфейса Zabbix:

Library	Mandatory status	Minimum version	Description
Composer	No	2.4.1	Менеджер пакетов уровня приложения для PHP, предоставляющий стандартный формат для управления зависимостями PHP-программного обеспечения и требуемых библиотек.
PHPUnit		8.5.29	Фреймворк модульного тестирования PHP для тестирования веб-интерфейса Zabbix.
SASS		3.4.22	Язык сценариев препроцессора, который интерпретируется и компилируется в Cascading Style Sheets (CSS).

Веб-браузер на стороне клиента

Файлы cookie и JavaScript должны быть включены.

Поддерживаются последние стабильные версии Google Chrome, Mozilla Firefox, Microsoft Edge, Apple Safari и Opera.

Warning:
Реализована политика одинакового источника для IFrame, что означает, что Zabbix нельзя размещать во фреймах на другом домене.

 Тем не менее страницы, размещенные во фрейме Zabbix, будут иметь доступ к веб-интерфейсу Zabbix (через JavaScript), если страница, размещенная во фрейме, и веб-интерфейс Zabbix находятся на одном домене. Страница вида `http://secure-zabbix.com/cms/page.html`, если она размещена на панелях `http://secure-zabbix.com/zabbix/`, будет иметь полный доступ к Zabbix через JS.

Сервер/прокси

Если указано, что компонент обязателен, то требуемое программное обеспечение/библиотека строго необходимы. Необязательные компоненты нужны для поддержки некоторых специфических функций.

Требование	Статус обязательности	Описание
libpcre2	Да	Библиотека PCRE2 требуется для поддержки Perl Compatible Regular Expression (PCRE). Поддерживается PCRE2 v10.x.
libevent		Требуется для межпроцессного взаимодействия. Версия 2.0.10 или выше.
libevent-pthreads		Требуется для межпроцессного взаимодействия.
libpthread		Требуется для поддержки mutex и блокировок чтения-записи (может быть частью libc).
libresolv		Требуется для разрешения DNS-имен (может быть частью libc).
libiconv		Требуется для преобразования кодировки/формата текста (может быть частью libc). Обязательна для сервера Zabbix в Linux.
libz		Требуется для поддержки сжатия.
libm		Математическая библиотека. Требуется только серверу Zabbix.
libmysqlclient	Один из	Требуется, если используется MySQL.
libmariadb		Требуется, если используется MariaDB.
libpq5		Требуется, если используется PostgreSQL; версия *libpq5* должна соответствовать версии используемой базы данных PostgreSQL или быть выше.
libsqlite3		Требуется, если используется SQLite. Требуется только для прокси Zabbix.
libOpenIPMI	Нет	Требуется для поддержки IPMI. Требуется только серверу Zabbix.

Требование	Статус обязательности	Описание
libssh2 или *libssh*		Требуется для SSH-проверок. Версия 1.8.0 или выше (libssh2); 0.9.0 или выше (libssh).
libcurl		Требуется для следующих возможностей: - элементов данных веб-мониторинга, мониторинга VMware и HTTP agent (для всех: версия 7.19.1 или выше); - элементов данных агента Zabbix web.page.* (версия 7.19.1 или выше; см. также требования к агент); - SMTP-аутентификации (Basic: версия 7.20.0 или выше; OAuth: версия 7.33 или выше; см. также требования к веб-интерфейс); - Elasticsearch (версия 7.28.0 или выше). Для всех возможностей рекомендуется версия 7.28.0 или выше. Чтобы использовать обновленные возможности cURL для элементов данных `web.page.*`, перезапустите сервер/прокси Zabbix. Для SMTP-аутентификации во время выполнения используйте пакет `libcurl-full`.
libxml2		Требуется для мониторинга VMware и предобработки XML XPath.
net-snmp		Требуется для поддержки SNMP. Версия 5.3.0 или выше. Поддержка протоколов сильного шифрования (AES192/AES192C, AES256/AES256C) доступна начиная с библиотеки net-snmp 5.8; в системах на базе RHEL 8+ рекомендуется использовать net-snmp 5.8.15 или новее.
libunixodbc		Требуется для мониторинга баз данных.
libgnutls или *libopenssl*		Требуется при использовании шифрования. Минимальные версии: *libgnutls* — 3.1.18, *libopenssl* — 1.0.1
libldap		Требуется для поддержки LDAP.
fping		Требуется для элементов данных ICMP ping.
c-ares		Требуется для асинхронного разрешения DNS-имен, если Zabbix собран с опцией `--with-ares`. В противном случае будет использоваться *libevent*. Минимальная версия: 1.16.0. Кэширование DNS-запросов (кэш запросов включен по умолчанию) требует c-ares 1.26.0 или новее.

Агент

Требование	Обязательный статус	Описание
libpcre2	Да	Библиотека PCRE2 требуется 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 блокировок чтения-записи (может быть частью libc). Не требуется в Windows.
libresolv		Требуется для разрешения DNS-имен (может быть частью libc). Не требуется в Windows.
libiconv		Требуется для кодирования/преобразования текстового формата в UTF-8 в элементах данных журналов, содержимого файлов, file regex и regmatch (может быть частью libc). Не требуется в Windows.
libgnutls or *libopenssl*	Нет	Требуется при использовании шифрования. Минимальные версии: *libgnutls* - 3.1.18, *libopenssl* - 1.0.1 В Microsoft Windows требуется OpenSSL 1.1.1 или новее.
libldap		Требуется, если используется LDAP. Не поддерживается в Windows.
libcurl		Требуется для расширенной поддержки элементов данных web.page.* агента Zabbix. 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.

Требование	Обязательный статус	Описание
libmodbus		Требуется только при использовании мониторинга Modbus. Версия 3.0 или выше.

Агент 2

Требование	Обязательность	Описание
Go	Да	Требуется для сборки Zabbix агента 2 и его плагинов из исходного кода. Поддерживается Go 1.24.10 или новее. Инструкции по установке см. на go.dev. Библиотеки Go, используемые Zabbix агентом 2 и его плагинами, перечислены в Git-репозитории Zabbix (библиотеки, помеченные в репозитории как `indirect`, являются зависимостями других требуемых библиотек): - Zabbix агент 2 - Поддержка плагинов - PostgreSQL - MongoDB - MSSQL - Ember+ - NVIDIA GPU - Пример плагина
libpcre2	Да	Библиотека PCRE2 требуется для поддержки Perl Compatible Regular Expression (PCRE). Поддерживается PCRE2 v10.x. Требуется для мониторинга журналов. Также требуется в Windows.
libopenssl	Нет	Требуется при использовании шифрования. На платформах UNIX требуется OpenSSL 1.0.1 или новее. В библиотеке OpenSSL должна быть включена поддержка PSK. LibreSSL не поддерживается. В системах Microsoft Windows требуется OpenSSL 1.1.1 или новее.

Веб-служба

Для создания отчётов по расписанию с использованием веб-службы Zabbix поддерживается последняя стабильная версия Google Chrome.

Требуемая версия Go для сборки веб-службы соответствует версии, используемой для Zabbix агент 2.

Java gateway

Если вы получили Zabbix из репозитория исходного кода или архива, то необходимые зависимости уже включены в дерево исходного кода.

Если вы получили Zabbix из пакета вашего дистрибутива, то необходимые зависимости уже предоставляются системой управления пакетами.

В обоих приведённых выше случаях программное обеспечение готово к использованию, и никаких дополнительных загрузок не требуется.

Однако если вы хотите использовать собственные версии этих зависимостей (например, если вы подготавливаете пакет для какого-либо дистрибутива Linux), ниже приведён список версий библиотек, с которыми Java gateway, как известно, работает. Zabbix также может работать и с другими версиями этих библиотек.

В следующей таблице перечислены JAR-файлы, которые в настоящее время входят в состав Java gateway в исходном коде:

Библиотека	Обязательный статус	Включённая версия	Комментарии
android-json	Да	4.3r1	JSON (JavaScript Object Notation) — это облегчённый формат обмена данными. Это совместимая с org.json реализация Android, извлечённая из Android SDK.
logback-classic		1.5.16	
logback-core		1.5.16	

Библиотека	Обязательный статус	Включённая версия	Комментарии
slf4j-api		2.0.16	

Java gateway может быть собран с использованием как Oracle Java, так и OpenJDK с открытым исходным кодом (версии 1.6 или новее). Пакеты, предоставляемые Zabbix, компилируются с использованием OpenJDK. В следующей таблице перечислены пакеты OpenJDK, используемые для сборки пакетов Zabbix в различных дистрибутивах:

Дистрибутив	Пакет OpenJDK
AlmaLinux 9	`java-11-openjdk-devel` (amd64: 11.0.19.0.7-4; arm64: 11.0.20.0.8-3)
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 9	`java-11-openjdk-devel` (amd64, arm64: 11.0.18.0.10-3)
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 12	`default-jdk-headless` (amd64, arm64: 2:1.17-74)
Debian 11	`default-jdk-headless` (amd64: 2:1.11-72)
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 9	`java-11-openjdk-devel` (amd64: 11.0.19.0.7-4.0.1; arm64: 11.0.20.0.8-2.0.1)
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 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 9	`java-11-openjdk-devel` (amd64: 11.0.19.0.7-4; arm64: 11.0.20.0.8-3)
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 9	`java-11-openjdk-devel` (amd64: 11.0.19.0.7-4; arm64: 11.0.20.0.8-3)
Rocky Linux 8	`java-1.8.0-openjdk-devel` (amd64: 1.8.0.372.b07-4; arm64: 1.8.0.382.b05-2)
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 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)

Номера портов по умолчанию

Следующий список открытых портов для каждого компонента применим для конфигурации по умолчанию:

Компонент Zabbix	Номер порта	Протокол	Тип соединения
Агент Zabbix	10050	TCP	по запросу
Агент Zabbix 2	10050	TCP	по запросу
Сервер Zabbix	10051	TCP	по запросу
Прокси Zabbix	10051	TCP	по запросу
Zabbix Java gateway	10052	TCP	по запросу
Веб-служба Zabbix	10053	TCP	по запросу
Веб-интерфейс Zabbix	80	HTTP	по запросу
	443	HTTPS	по запросу
Zabbix trapper	10051	TCP	по запросу

Note:
Номера портов должны быть открыты в межсетевом экране, чтобы обеспечить обмен данными Zabbix. Исходящие TCP-соединения обычно не требуют явной настройки межсетевого экрана.

Размер базы данных

Данные конфигурации Zabbix требуют фиксированного объема дискового пространства и почти не 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 в базу данных Zabbix каждую секунду 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, файлы журналов и т. д.) объем дискового пространства 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/лог-элементов данных невозможно точно предсказать, но можно ожидать около 500 байт на значение.

- Настройка housekeeper для трендов

Zabbix хранит 1-часовой набор значений max/min/avg/count для каждого элемента данных в таблице **trends**. Эти данные используются для трендов и графиков за длительный период. Период в один час нельзя настроить.

База данных Zabbix, в зависимости от типа базы данных, требует около 90 байт на каждый такой набор. Предположим, мы хотим хранить данные трендов 5 лет. Значения для 3000 элементов данных потребуют 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 будут восстановлены, 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. Таким образом, на каждое событие потребуется дополнительно #tags * 100 байт дискового пространства.

Это означает, что если мы хотим хранить события 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] Приблизительные размеры основаны на MySQL и могут отличаться для других баз данных.

Таблица содержит формулы, которые можно использовать для расчета дискового пространства, необходимого для системы Zabbix:

Parameter	Formula for required disk space (in bytes)
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.

Parameter	Formula for required disk space (in bytes)
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. Одно (1) событие в секунду в худшем случае. 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 использовано сразу после установки Zabbix. Размер базы данных будет расти, а затем в какой-то момент перестанет увеличиваться, в зависимости от настроек housekeeper.

Синхронизация времени

Очень важно, чтобы на сервере, где работает Zabbix, было точно настроено системное время. ntpd — это наиболее популярный демон, который синхронизирует время узла сети со временем других машин. Настоятельно рекомендуется поддерживать синхронизированное системное время на всех системах, на которых работают компоненты Zabbix.

Требования к сети

Следующий список открытых портов для каждого компонента применим для конфигурации по умолчанию.

Компоненты	Порт
Веб-интерфейс	http на 80, https на 443
Сервер	10051 (для использования с активным прокси/агентами)
Активный прокси	10051
Пассивный прокси	10051
Agent2	10050
Trapper	
JavaGateway	10052
WebService	10053

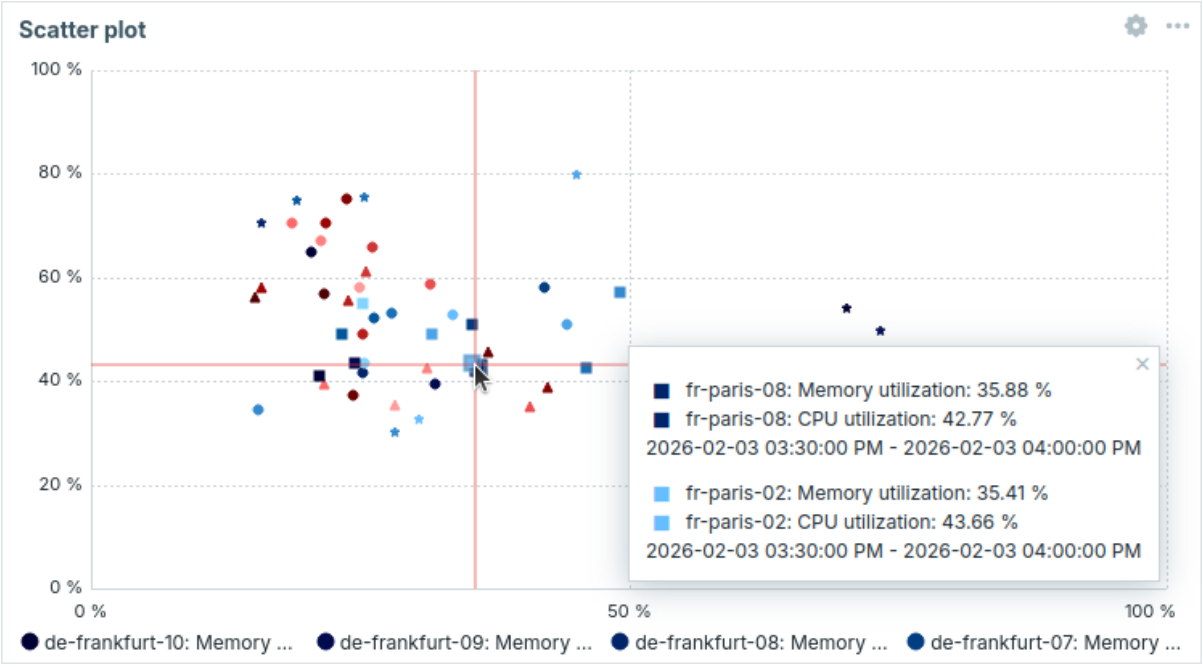
Note:
Номера портов должны быть открыты в межсетевом экране, чтобы обеспечить внешние соединения с Zabbix. Исходящие TCP-соединения обычно не требуют явной настройки межсетевого экрана.

2 Что нового в Zabbix 8.0

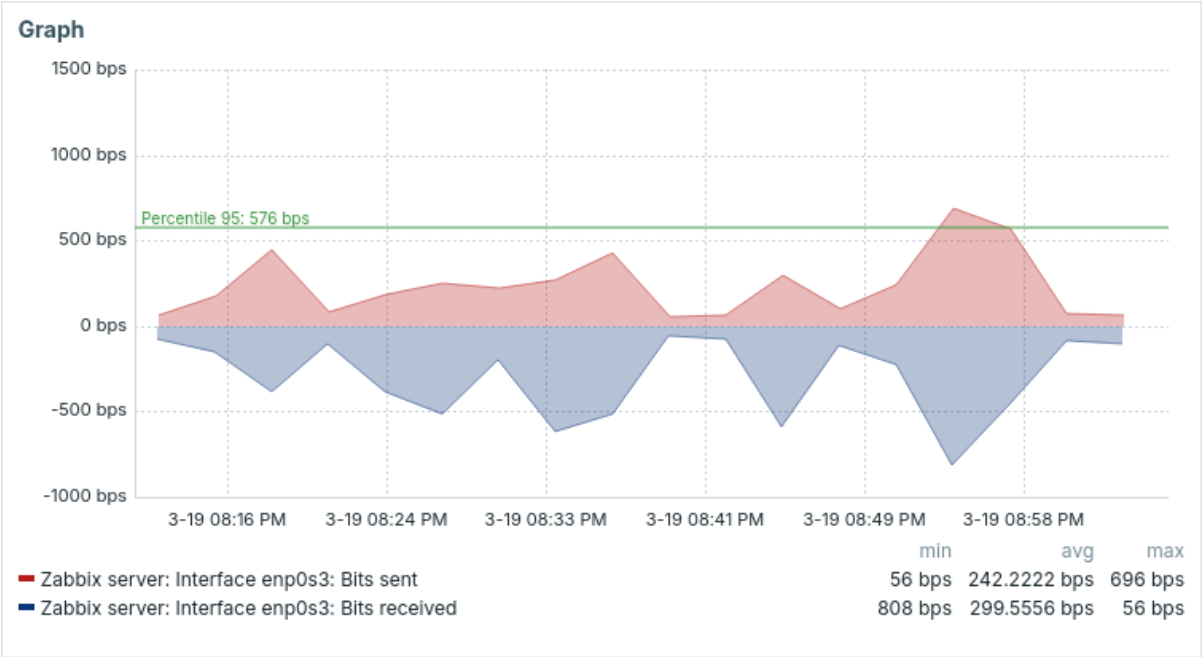
Zabbix 8.0.0 продолжает развитие Zabbix 7.4.0, добавляя новые функции и улучшения.

См. несовместимые изменения для этой версии.

Виджет диаграммы рассеяния Виджет Scatter plot был добавлен в виджеты панели мониторинга. Он отображает взаимосвязь между двумя метриками, нанося отдельные точки данных по осям X и Y. Это помогает выявлять закономерности, кластеры, корреляции и выбросы в наборе данных.



Инверсия оси Y для виджета Graph Графики в виджете Graph теперь могут отображаться с инвертированными значениями оси Y. Новый параметр *Invert values* позволяет умножать значения оси Y на −1 без изменения исходных данных.



См. также: Улучшения виджета Graph.

Шаблоны

Новые шаблоны

- Aruba CX 8300s by SNMP — шаблон, предоставляющий мониторинг серии коммутаторов Aruba CX 8300 на основе SNMP.
- Набор шаблонов AWS by HTTP был дополнен шаблоном *AWS Backup Vault by HTTP*.
- Набор шаблонов Azure by HTTP был дополнен шаблоном *Azure Sentinel by HTTP*.
- Ciena 3906 by SNMP — шаблон для мониторинга устройств Ciena 3906.
- Cisco Secure Firewall Threat Defense by HTTP — шаблон, предоставляющий возможности мониторинга устройств Cisco Secure Firewall Threat Defense с использованием REST API.
- Cradlepoint NCM v2 by HTTP — набор шаблонов для мониторинга Cradlepoint NCM v2 и его устройств через HTTP.
- Domain RDAP by HTTP — шаблон для мониторинга данных регистрации доменов по протоколу RDAP.
- IBM Maximo Service Request — шаблон на основе вебхука, который позволяет связать Zabbix с IBM Maximo.
- MariaDB by ODBC — шаблон для мониторинга баз данных MariaDB через ODBC.

- Microsoft Hyper-V Failover Cluster by SSH и Microsoft Hyper-V Standalone by SSH — шаблоны для мониторинга кластеров Microsoft Hyper-V и автономных узлов сети через SSH.
- OpenAI Platform by HTTP — шаблон для мониторинга платформы разработчика OpenAI.
- Percona by ODBC — шаблон для мониторинга баз данных Percona через ODBC.
- Ribbon SBC Edge by HTTP — шаблон для мониторинга устройств Ribbon SBC Edge (ранее SWe Lite) через HTTP.
- Ribbon SBC SWe Core by HTTP — шаблон для мониторинга устройств SBC SWe Core через HTTP, а Ribbon SBC SWe CE by HTTP — шаблон для мониторинга экземпляров Ribbon SBC SWe Call Engine (CE) через HTTP.
- Stormshield SNS by SNMP — шаблон для мониторинга устройств Stormshield Network Security (SNS) через SNMP.
- VeloCloud SD-WAN Edge by HTTP — шаблон для мониторинга устройств VeloCloud SD-WAN Edge через HTTP.
- Vyatta Virtual Router by SNMP — шаблон для мониторинга виртуального маршрутизатора Vyatta 1908e.

Обновленные шаблоны

- Ciena 3906 by SNMP был обновлен и теперь включает элементы данных файловой системы и загрузки CPU.
- Набор шаблонов GCP by HTTP был дополнен шаблоном *GCP Cloud Run Service by HTTP*.
- GitHub organization by HTTP был обновлен и теперь включает элементы данных для мониторинга Microsoft Copilot.
- Microsoft 365 reports by HTTP был обновлен и теперь включает элементы данных для мониторинга Microsoft Copilot.
- MySQL by Zabbix агент, MySQL by Zabbix агент 2, MySQL by Zabbix агент active, MySQL by Zabbix агент 2 active и MySQL by ODBC были обновлены для поддержки как `SHOW SLAVE STATUS` (старый синтаксис), так и `SHOW REPLICA STATUS` (новый синтаксис).
- MySQL by ODBC также был обновлен новыми метриками, правилами обнаружения для таблиц и реплик, а также улучшенной панелью мониторинга.
- PostgreSQL by ODBC был обновлен новыми метриками, правилами обнаружения и панелями мониторинга, а также общими улучшениями производительности и наблюдаемости.
- Proxmox VE by HTTP был обновлен с поддержкой функциональности вложенного LLD. Кроме того, для наглядности был скорректирован формат единиц измерения для элементов данных, отображающих проценты.
- VeloCloud SD-WAN by HTTP, ранее *VMWare SD-WAN VeloCloud by HTTP*, был переименован и обновлен в соответствии с последней версией платформы VeloCloud SD-WAN.
- Vyatta Virtual Router by SNMP был обновлен новыми элементами данных OID и улучшениями панели мониторинга.

Элементы данных

Тип данных JSON Теперь Zabbix поддерживает JSON как тип данных для значений элементов данных.

Ранее значения JSON собирались текстовыми элементами данных и хранились как строки с ограничением 64 КБ. Теперь Zabbix может хранить значения JSON в собственном формате с ограничением 128 МиБ, а также отклонять некорректные значения JSON (например, содержащие ключи без кавычек, завершающие запятые, несоответствующие скобки).

Тип данных JSON поддерживается всеми типами элементов данных и прототипами элементов данных (кроме *Calculated*), а также доступен в экспорте данных в реальном времени и коннекторах. Значения JSON могут храниться во всех поддерживаемых базах данных и Elasticsearch. Если вы используете TimescaleDB, см. примечания по обновлению.

Обратите внимание, что элементы данных JSON нельзя использовать в триггерах; однако вы можете извлекать поля JSON с помощью зависимых элементов данных с типом данных, отличным от JSON, и использовать их в триггерах.

Элементы данных, которые возвращают строку JSON (`net.if.discovery`, `vfs.file.get` и т. д.), по-прежнему остаются текстовыми элементами данных; однако при необходимости вы можете изменить их тип на JSON.

Подробнее, включая ограничения для данных JSON, см. в разделе настройка элемента данных.

Параметр type для обнаружения дисков S.M.A.R.T. Элемент данных smart.disk.discovery (плагин S.M.A.R.T. агента Zabbix 2) теперь принимает необязательный параметр **type** для указания значения, по которому выполнять поиск дисков.

Плагины

Плагин Ceph Теперь этот плагин работает в двух режимах:

- **native** — в этом режиме для прямого взаимодействия с кластером Ceph через собственный API Ceph (протокол msgr2) используется библиотека go-ceph. Это рекомендуемый режим для современных установок Ceph, однако он поддерживается только **в Linux** и начиная с Ceph 16.
- **restful** (устаревший) — в этом режиме для взаимодействия используется Ceph RESTful API. Это режим по умолчанию для обратной совместимости, однако он не будет работать с Ceph версии 20 (Tentacle) и новее из-за удаления модуля mgr/restful.

Используемый режим определяется значением параметра mode (native/restful):

- `Plugins.Ceph.Default.Mode=native` — установить для плагина режим native
- `Plugins.Ceph.Sessions.<SessionName>.Mode=native` — установить режим native для именованной сессии

Обратите внимание, что набор учетных данных пользователя различается для каждого режима, и они несовместимы друг с другом. Параметр `Plugins.Ceph.InsecureSkipVerify` игнорируется в режиме native, поскольку безопасность соединения определяется на стороне кластера Ceph протоколом msgr2 (по умолчанию безопасным).

Обратите внимание, что плагин Ceph для Zabbix агент 2 теперь является загружаемым плагином и требует дополнительных шагов установки. Это связано с требованием наличия пакета librados (для режима native). Подробности см. в readme плагина Ceph.

Плагин MongoDB Плагин MongoDB для Zabbix агент 2 предлагает несколько улучшений: - Полная поддержка стандартного разбора MongoDB URI, включая поддержку схем `mongodb://` и `mongodb+srv://`. - Автоматическое обнаружение узлов набора реплик MongoDB, что расширяет возможности мониторинга кластеров MongoDB. - Поддержка аутентификации x509, обеспечивающая безопасные подключения к MongoDB с использованием клиентских сертификатов.

Плагин Oracle Oracle plugin для Zabbix агент 2 теперь поддерживает зашифрованные подключения к базам данных Oracle с использованием протокола TCPS (TLS). Это позволяет выполнять мониторинг экземпляров Oracle через сокеты, защищённые TLS, повышая безопасность удалённого мониторинга.

Плагин Redis — поддержка TLS и проверка при запуске В плагин Redis для Zabbix агент 2 была добавлена поддержка TLS.

Реализована проверка конфигурации TLS плагина при запуске, а также улучшены сообщения проверки/об ошибках. Некорректная логика конфигурации (например, использование типа соединения `verify_full` без указания `TLSCAFile`) может помешать запуску Zabbix агент 2.

Режим тестового выполнения для загружаемых плагинов Загружаемые плагины теперь можно запускать в тестовом режиме с помощью флага `-t` (`--test`), передавая в качестве аргумента ключ элемента данных. В этом режиме плагин запускается для целей отладки и разработки, а файлы конфигурации плагина игнорируются.

Обнаружение низкого уровня

Флажок Convert to JSON в формах обнаружения В форму правила обнаружения и форму прототипа обнаружения добавлен новый флажок *Convert to JSON*, который отображается, если в выпадающем списке *Type* выбран "HTTP агент". Этот параметр позволяет пометить полученные данные для автоматического преобразования в JSON перед дальнейшей обработкой.

Поддержка макросов во вложенном обнаружении низкого уровня Макросы обнаружения низкого уровня теперь поддерживаются во вложенных правилах обнаружения низкого уровня в:

- параметрах предобработки JSONPath parameters
- поле JSONPath для пользовательских LLD-макросов

Редактируемые теги для триггеров, созданных с помощью обнаружения низкого уровня Для триггеров, созданных из прототипов триггеров, теперь можно добавлять теги вручную. Теги, унаследованные от прототипов триггеров, по-прежнему применяются автоматически. Теги, добавленные вручную, можно изменять в обнаруженных триггерах, и они будут включены в массивы тегов событий и доступны для функций, учитывающих теги, таких как фильтры, виджеты панели мониторинга и макросы уведомлений.

Процессы

Кэширование и повторное использование EngineID SNMPv3 Теперь Zabbix кэширует сопоставления SNMPv3 EngineID → IP и пытается повторно использовать кэшированные EngineID для последующих проверок SNMPv3, уменьшая объём пробного трафика и повышая производительность опрашивателя. Если повторно используемый EngineID не отвечает, опрашиватель возвращается к запросу EngineID и может удалять устаревшие записи после изменений интерфейса или при постоянных сбоях.

Уточнена логика ограничения прокси во время восстановления кэша истории Логика ограничения прокси была уточнена для повышения стабильности сервера во время восстановления кэша истории. Когда использование кэша истории достигает порога ограничения, сервер, как и раньше, перестает принимать данные от прокси. Когда использование кэша снижается до 60%, сервер начинает обрабатывать список ограничения, но все еще может отклонять загрузки от прокси, содержащие очень большие пакеты данных (примерно более 10 тыс. записей), пока нагрузка на кэш не уменьшится еще больше. Это изменение снижает риск повторных перегрузок кэша во время восстановления сервера.

Увеличено максимальное значение тайм-аута для zabbix_get и zabbix_js Максимальное значение параметра `timeout` для утилит командной строки zabbix_get и zabbix_js было увеличено до 600 секунд.

Оптимизированная очистка истории Очистка истории была оптимизирована за счет переноса создания задач очистки в триггеры базы данных. При удалении элементов данных (включая правила низкоуровневого обнаружения), триггеров, сервисов или правил сетевого обнаружения триггеры базы данных (вместо явных вызовов на уровне приложения) теперь заполняют таблицу `housekeeper` задачами очистки. Подробности см. в разделе процедура очистки истории.

Теперь housekeeper также удаляет события сетевого обнаружения, созданные на основе обнаруженных узлов сети или сервисов, которые впоследствии были удалены, а также все события, связанные с проблемами триггеров, которые впоследствии были удалены (ранее удалялись только сами проблемы; связанные события удалялись только после истечения периода хранения данных триггеров housekeeper *Trigger data storage period*).

Ручное закрытие — теги триггера, наследуемые событиями восстановления События восстановления, созданные после ручного закрытия, наследуют теги триггера в дополнение к тегам элемента данных и узла сети. Эти теги присутствуют в массиве тегов события и доступны для макросов уведомлений, таких как `{EVENT.RECOVERY.TAGS}` и `{EVENT.RECOVERY.TAGSJSON}`.

Кэширование DNS-запросов для демонов Zabbix Сервер Zabbix, прокси Zabbix и агент Zabbix получили поддержку использования резолвера c-ares для всех DNS-запросов, обеспечивая кэширование DNS-запросов и улучшенное переключение на резервный резолвер при сбое, если Zabbix собран с `--with-ares`. Для кэширования DNS-запросов требуется c-ares 1.26.0 или более поздней версии.

Сборка поддержки c-ares в Windows Теперь агент Zabbix можно собрать с использованием резолвера c-ares в Microsoft Windows. c-ares можно установить через vcpkg, а сборка агента поддерживает либо `ARES=<vcpkg prefix>`, либо отдельные пути `ARESINCDIR`/`ARESLIBDIR` для каталогов include и библиотек.

Аутентификация

Импортируемые сертификаты SAML для единого входа Пользователи с правами Super admin теперь могут импортировать сертификаты и закрытые ключи напрямую в веб-интерфейс для настройки SAML. В *Administration* > *Authentication* > *SAML* были добавлены три новых поля:

- *Сертификат IdP* — сертификат X.509, предоставляемый поставщиком удостоверений
- *Сертификат SP* — сертификат поставщика услуг, используемый для обмена данными SAML
- *Закрытый ключ SP* — закрытый ключ, соответствующий сертификату SP

Эти элементы управления позволяют добавлять новые значения или изменять существующие на вкладке настроек SAML. Перед сохранением в выбранное хранилище сертификаты и закрытые ключи проходят проверку; недопустимые или некорректно сформированные значения отклоняются с пояснением ошибки.

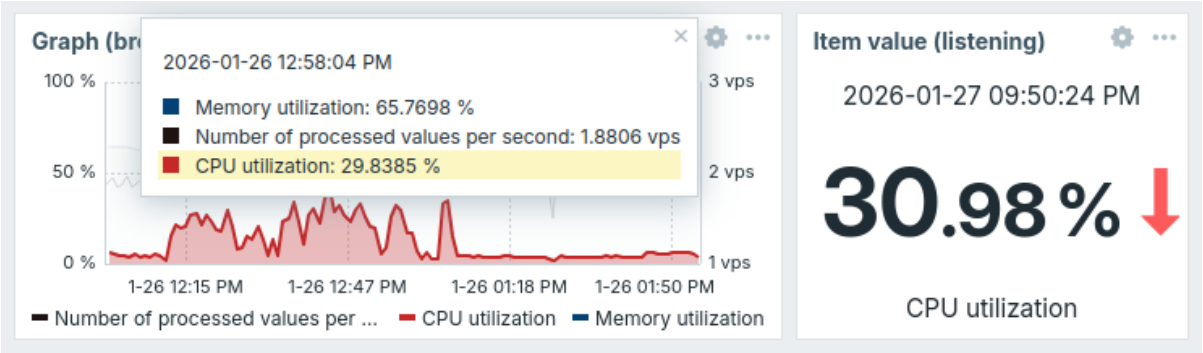
Виджеты

Кластеризация маркеров узлов сети на геокарте Виджет Geomap теперь поддерживает параметр *Clustering* для управления тем, как близко расположенные маркеры узлов сети объединяются в один маркер со счетчиком. Задав уровень масштабирования карты для кластеризации, вы сможете сохранить читаемость больших карт и поддерживать точную видимость при увеличении масштаба.

Агрегированные столбцы в Top items Виджет Top items теперь может группировать шаблоны элементов данных в один агрегированный столбец или строку, что упрощает отображение объединённых метрик (например: общий входящий трафик по всем сетевым интерфейсам узла сети). При настройке столбцов элементов данных были добавлены три новые опции: *Агрегированные столбцы*, *Функция агрегации столбца* и *Имя объединённого столбца*.

Улучшения виджета Graph Виджет Graph предлагает несколько улучшений удобства использования:

- Элементы данных во всплывающей подсказке графика, которая появляется при наведении курсора на график, сортируются по значению в порядке убывания.
- При наведении курсора на элемент данных во всплывающей подсказке его график выделяется, а остальные приглушаются.
- При выборе элемента данных во всплывающей подсказке его данные передаются другим виджетам, которые поддерживают их получение.
- Ограничение на количество элементов данных, отображаемых во всплывающей подсказке, удалено.
- Новый параметр *Host names in labels* позволяет выбрать, будут ли имена узлов сети отображаться во всплывающей подсказке и в легенде графика.



- Новый параметр *Invert values* позволяет инвертировать график.

Веб-интерфейс

Новые и встроенные шрифты Теперь Zabbix включает новые встроенные шрифты, которые улучшают читаемость, загружаются быстрее и отображают дополнительные языки веб-интерфейса с минимальным влиянием на макет. Поскольку эти шрифты встроены, они также обеспечивают единообразный внешний вид в разных системах.

Новые шрифты:

Linux by Zabbix agent: Available memory in %: Memory utilization	Triggers 1	vm.memory.utilization
Linux by Zabbix agent: Total memory	Triggers 1	vm.memory.size[total]
Linux by Zabbix agent: Total swap space	Triggers 1	system.swap.size[,total]

Устаревшие шрифты:

Linux by Zabbix agent: Available memory in %: Memory utilization	Triggers 1	vm.memory.utilization
Linux by Zabbix agent: Total memory	Triggers 1	vm.memory.size[total]
Linux by Zabbix agent: Total swap space	Triggers 1	system.swap.size[,total]

Новые шрифты используются почти во всех темах. При необходимости доступны недавно добавленные темы *Blue (classic)* и *Dark (classic)* с устаревшими шрифтами.

Семейство шрифтов для моноширинного содержимого и графиков остаётся без изменений.

Встроенная валидация Следующие формы в веб-интерфейсе были добавлены в набор форм, поддерживающих встроенную валидацию:

- API-токен
- Аутентификация
- Авторегистрация
- Коннектор
- Географические карты
- Группа узлов сети
- Прототипы узлов сети
- Сопоставление иконок
- Изображения
- Обслуживание
- Способ оповещения
- Уведомления
- Профиль
- Прокси
- Регулярные выражения
- Сервис
- SLA
- Группа шаблонов
- Обновление проблемы

- Глобальные пользовательские макросы
- Роли пользователей
- Пользователи

Ошибки ввода отображаются сразу после заполнения полей, что повышает удобство использования и снижает количество ошибок конфигурации.

Модальные формы При настройке низкоуровневого обнаружения форма конфигурации прототипа узла сети теперь открывается в модальном (всплывающем) окне.

Унаследованные теги, видимые в узлах сети, шаблонах, элементах данных и триггерах Унаследованные теги теперь отображаются и возвращаются единообразно для шаблонов, узлов сети, элементов данных, веб-сценариев и триггеров. Теги, унаследованные из цепочек шаблонов/узлов сети, видны в *Мониторинг > Последние данные*. Фильтрация по унаследованным тегам доступна везде, где поддерживается фильтрация по тегам, включая разделы *Мониторинг > Последние данные* и *Сбор данных*, а также все виджеты панелей мониторинга, в которых доступна фильтрация по тегам узлов сети, элементов данных, триггеров или веб-сценариев, — поэтому выборка и дополнительная фильтрация на основе тегов работают одинаково независимо от того, где был определён тег.

Вкладка *Теги* в формах настройки шаблона, узла сети и прототипа узла сети теперь содержит переключатель для выбора способа отображения тегов: в шаблонах показываются *Теги шаблона* / *Унаследованные теги и теги шаблона*, а в узлах сети и прототипах узлов сети — *Теги узла сети* / *Унаследованные теги и теги узла сети*. Унаследованные теги визуально выделяются новым значком документа с контуром рядом с меткой тега.

Виджеты Graph и Круговая диаграмма включают новый параметр *Теги элементов данных*.

Позиционирование всплывающих подсказок Теперь всплывающие подсказки можно перемещать в новое место, перетаскивая их. Это относится, например, к всплывающей подсказке Графика, а также к всплывающим подсказкам с описаниями в разделах Последние данные и Проблемы.

Документация

Сводные страницы документации для минорных релизов Документация по релизам для минорных версий основного релиза Zabbix теперь будет собираться на отдельных сводных страницах для новых возможностей и примечаний по обновлению соответственно.

Что нового в Zabbix 8.0.x

На этой странице представлена сводная информация о новых возможностях, включённых в минорные выпуски основной версии Zabbix.

См. также Что нового для основной версии.

Информацию об изменениях в существующих шаблонах и о новых шаблонах для различных устройств, сервисов и партнёрских решений см. в разделе Изменения шаблонов.

Что нового в Zabbix 8.0.1 Эта версия ещё не выпущена.

3 Zabbix appliance

Обзор Устройство Zabbix предоставляет возможность мгновенно развернуть сервер Zabbix и веб-интерфейс, вместо того чтобы настраивать их вручную или повторно использовать существующий сервер для Zabbix.

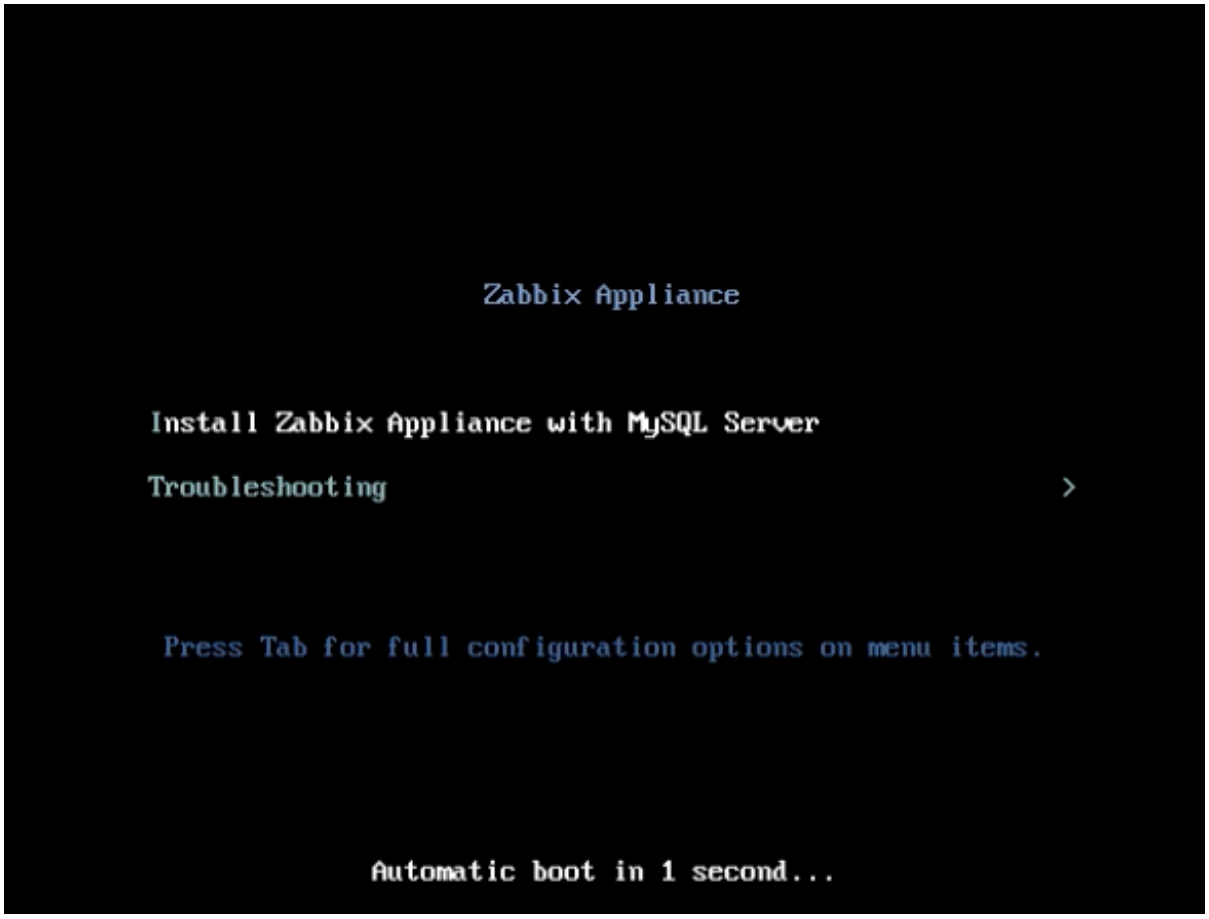
Устройство основано на AlmaLinux 8 (x86_64) и содержит предварительно настроенный сервер Zabbix, работающий на MySQL, и веб-интерфейс, работающий на веб-сервере Nginx.

> **Attention:**
> Это устройство предназначено для оценки Zabbix. Не рекомендуется использовать его в серьёзных производственных средах.

Образы устройства доступны для загрузки в следующих форматах:

- Установочный CD/DVD (.iso)
- VMware (.vmx) — см. примечания
- Open Virtualization Format (.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 из загруженного образа.
2. Настройте параметры сети виртуальной машины так, чтобы обеспечить доступ к ней из браузера на хост-машине. Этого можно добиться, включив *режим моста*.
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 хост-машине и перейдите по IP-адресу, 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 вы можете 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 в процессе установки. Для базы данных определены следующие пользователи:

Root:

- Имя пользователя: root
- Пароль: пароль хранится в файле `/root/.my.cnf`. Для учетной записи root ввод пароля не требуется.

Сервер Zabbix:

- Имя пользователя: zabbix_srv
- Пароль: пароль хранится в `/etc/zabbix/zabbix\_server.conf`

Веб-интерфейс Zabbix:

- Имя пользователя: zabbix_web
- Пароль: пароль хранится в `/etc/zabbix/web/zabbix.conf.php`

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 виртуальной машины>*.

По умолчанию доступ разрешён отовсюду. Чтобы ограничить доступ, измените файл `/etc/nginx/conf.d/zabbix.conf`. После сохранения изменённого файла перезапустите Nginx, войдя через SSH как пользователь root и выполнив:

```
systemctl restart nginx
```

Статический IP-адрес

По умолчанию appliance использует DHCP для получения IP-адреса. Чтобы задать статический IP-адрес:

- Войдите как пользователь root.
- Выполните следующие команды, заменив значения на ваши собственные IP-адреса:

```
nmcli connection modify eth0 ipv4.addresses 192.168.1.10/24    # Appliance IP address/CIDR prefix
nmcli connection modify eth0 ipv4.gateway 192.168.1.1     # Gateway IP address
nmcli connection modify eth0 ipv4.dns 8.8.8.8    # DNS server IP address
nmcli connection modify eth0 ipv4.method manual
systemctl restart network
```

В качестве альтернативы можно открыть файл `/etc/NetworkManager/system-connections/eth0.nmconnection` и добавить следующие строки:

```
[ipv4]
address1=192.168.1.10/24,192.168.1.1
dns=8.8.8.8
method=manual
```

После сохранения измененного файла выполните команду `systemctl restart network`.

Настройка межсетевого экрана

Для управления настройками межсетевого экрана appliance использует iptables с предопределёнными правилами:

- Открыт порт SSH (22 TCP)

- Открыты порты агента Zabbix (10050 TCP) и Zabbix trapper (10051 TCP)
- Открыты порты HTTP (80 TCP) и HTTPS (443 TCP)
- Открыт порт SNMP trap (162 UDP)
- Открыты исходящие соединения к порту NTP (123 UDP)
- ICMP-пакеты ограничены до 5 пакетов в секунду
- Все остальные входящие соединения отбрасываются

Чтобы открыть дополнительные порты, измените файл `/etc/sysconfig/iptables` и перезагрузите правила межсетевого экрана:

```
systemctl reload iptables
```

Репозитории

Устройство Zabbix Appliance использует 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 настраивается отдельно и может быть изменен в настройках веб-интерфейса. Часовой пояс по умолчанию для веб-интерфейса Zabbix — Europe/Riga.

Расположение файлов

- Файлы конфигурации расположены в `/etc/zabbix`
- Файлы журналов сервера Zabbix, прокси и агента расположены в `/var/log/zabbix`
- Веб-интерфейс Zabbix расположен в `/usr/share/zabbix`
- Домашний каталог пользователя *zabbix* — `/var/lib/zabbix`

Системные службы

Службы systemd доступны. Чтобы увидеть список служб Zabbix, выполните следующую команду на виртуальной машине:

```
systemctl list-units zabbix*
```

Примечания по форматам изображений VMware

Образы в формате *vmdk* можно использовать напрямую 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). Если вы используете VMware vCenter Converter, вы можете столкнуться с проблемами, связанными с гибридным сетевым адаптером. В таком случае можно попробовать указать адаптер E1000 в процессе преобразования. Либо после завершения преобразования можно удалить существующий адаптер и добавить адаптер E1000.

Образ HDD/flash (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 Flash/HDD диска.

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 войти в веб-интерфейс вы видите сообщение об ошибке *Access denied for user 'replace_user'@'localhost' (using password: YES)*, это может указывать на то, что установка все еще выполняется.



Если ошибка сохраняется спустя несколько минут ожидания или вы замечаете любое другое непредвиденное поведение, это, вероятно, означает, что процесс установки не был успешно завершен. В этом случае мы рекомендуем удалить текущее appliance и развернуть его заново, следуя тем же инструкциям по установке. Обычно это решает проблему.

Обратите внимание, что пытаться вручную исправить поврежденную установку не рекомендуется, так как это может привести к дальнейшим осложнениям.

4 Процессы Zabbix

Пожалуйста, используйте боковую панель для доступа к содержимому раздела процессов Zabbix.

1. 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 самостоятельно удаленно проверять сетевые службы (такие как веб-серверы и почтовые серверы), используя простые проверки служб.

Сервер является центральным хранилищем, в котором сохраняются все данные конфигурации, статистики и эксплуатации, а также именно он в Zabbix активно оповещает администраторов при возникновении проблем в любой из контролируемых систем.

Функционирование базового сервера Zabbix разделено на три отдельных компонента: сервер Zabbix, веб-интерфейс и хранилище базы данных.

Вся информация о конфигурации Zabbix хранится в базе данных, с которой взаимодействуют как сервер, так и веб-интерфейс. Например, когда вы создаете новый элемент данных с помощью веб-интерфейса (или API), он добавляется в таблицу items базы 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 изменения, внесенные в веб-интерфейсе Zabbix, могут появиться в разделе последних данных с задержкой до двух минут.

Запуск сервера

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 вышеописанное не работает, необходимо запустить его вручную. Найдите путь к исполняемому файлу zabbix_server и выполните:

```
zabbix_server
```

С сервером Zabbix можно использовать следующие параметры командной строки:

```
-c --config <file>                              (        /usr/local/etc/zabbix_server.conf)
-f --foreground                              Zabbix
-R --runtime-control <option>
-T --test-config
-h --help
-V --version
```

Примеры запуска сервера Zabbix 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_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 сервера.	**historycache** - статистика кэша истории **valuecache** - статистика кэша значений **preprocessing** - статистика менеджера предобработки **alerting** - статистика менеджера оповещений **lld** - статистика менеджера LLD **locks** - список мьютексов (пуст на системах *BSD*) **connector** - статистика для коннекторов с наибольшей очередью
ha_status	Записать состояние кластера высокой доступности (HA) в журнал.	
ha_remove_node=target	Удалить узел высокой доступности (HA), указанный по имени или ID. Обратите внимание, что активные/резервные узлы не могут быть удалены.	**target** - имя или ID узла (можно получить, выполнив ha_status)
ha_set_failover_delay=delay	Установить задержку переключения при отказе для высокой доступности (HA). Поддерживаются суффиксы времени, например 10s, 1m.	
proxy_config_cache_reload[=**target**]	Перезагрузить кэш конфигурации прокси.	**target** - список имен прокси, разделенных запятыми Если цель не указана, конфигурация будет перезагружена для всех прокси
secrets_reload	Перезагрузить секреты из Vault.	
service_cache_reload	Перезагрузить кэш менеджера сервисов.	
snmp_cache_reload	Перезагрузить кэш SNMP — очистить свойства SNMP engine (engine time, engine boots, engine id, credentials) для всех узлов сети. Используйте для принудительной глобальной очистки кэша при устранении проблем SNMP.	
housekeeper_execute	Запустить процедуру очистки истории. Игнорируется, если процедура очистки истории уже выполняется.	

Параметр	Описание	Цель
trigger_housekeeper_execute	Запустить процедуру очистки истории триггеров. Игнорируется, если процедура очистки истории триггеров уже выполняется.	
log_level_increase[=<**target**>]	Повысить уровень журналирования; если цель не указана, влияет на все процессы. Не поддерживается в системах *BSD*.	**process type** - Все процессы указанного типа (например, poller) См. все типы процессов сервера. **process type,N** - Тип процесса и номер (например, poller,3) **pid** - Идентификатор процесса (от 1 до 65535). Для больших значений укажите цель как 'process type,N'.
log_level_decrease[=<**target**>]	Понизить уровень журналирования; если цель не указана, влияет на все процессы. Не поддерживается в системах *BSD*.	
prof_enable[=<**target**>]	Включить профилирование. Если цель не указана, влияет на все процессы. Включенное профилирование предоставляет сведения обо всех rwlocks/mutexes по имени функции.	**process type** - Все процессы указанного типа (например, history syncer) Поддерживаемые типы процессов как цели 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, preprocessing worker, proxy poller, self-monitoring, service manager, snmp trapper, task manager, timer, trapper, unreachable poller, vmware collector **process type,N** - Тип процесса и номер (например, history syncer,1) **pid** - Идентификатор процесса (от 1 до 65535). Для больших значений укажите цель 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. Если цель не указана, влияет на все процессы.	**process type** - Все процессы указанного типа (например, history syncer) Поддерживаемые типы процессов как цели профилирования: см. `prof_enable` **process type,N** - Тип процесса и номер (например, history syncer,1) **pid** - Идентификатор процесса (от 1 до 65535). Для больших значений укажите цель как 'process type,N'.

Пример использования runtime control для перезагрузки кэша конфигурации сервера:

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

Примеры использования runtime control для перезагрузки конфигурации прокси:

```
###                        :
zabbix_server -R proxy_config_cache_reload

###                   Proxy1   Proxy2:
zabbix_server -R proxy_config_cache_reload=Proxy1,Proxy2
```

Пример использования runtime control для очистки кэша истории для элемента данных:

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

Примеры использования runtime control для сбора диагностической информации:

```
###                                    :
zabbix_server -R diaginfo

###                                :
zabbix_server -R diaginfo=historycache
```

Пример использования runtime control 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 завершается неудачей после изменения учетных данных (например, из-за несоответствий engineBoots/engineID или устройств, не соответствующих RFC), либо когда необходимо принудительно выполнить глобальную очистку кэша SNMP для диагностики.

Пример использования runtime control для запуска выполнения housekeeper:

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

Примеры использования runtime control для изменения уровня логирования:

```
###                          :
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 HA на минимальное значение 10 секунд:

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

Пользователь процесса

Сервер Zabbix разработан для запуска от имени пользователя, не являющегося root. Он будет работать от имени любого пользователя, не являющегося root, от которого был запущен. Поэтому вы можете без каких-либо проблем запускать сервер от имени любого пользователя, не являющегося root.

Если вы попытаетесь запустить его от имени 'root', он переключится на жестко заданного пользователя 'zabbix', который должен присутствовать в вашей системе. Вы можете запускать сервер от имени 'root' только в том случае, если соответствующим образом измените параметр 'AllowRoot' в файле конфигурации сервера.

Если сервер Zabbix и агент запущены на одной и той же машине, рекомендуется использовать для запуска сервера пользователя, отличного от пользователя, от имени которого запущен агент. В противном случае, если оба процесса запущены от имени одного и того же пользователя, агент сможет получить доступ к файлу конфигурации сервера, и любой пользователь уровня Admin в Zabbix сможет довольно легко получить, например, пароль к базе данных.

Файл конфигурации

Подробные сведения о настройке zabbix_server см. в разделе параметры файла конфигурации.

Скрипты запуска

Скрипты используются для автоматического запуска/остановки процессов Zabbix во время запуска/выключения системы. Скрипты расположены в каталоге misc/init.d.

Типы процессов и потоков сервера

- `agent poller` — асинхронный процесс опроса для пассивных проверок агента с рабочим потоком
- `alert manager` — менеджер очереди оповещений
- `alert syncer` — процесс записи оповещений в БД
- `alerter` — процесс для отправки уведомлений

- availability manager — процесс для обновления доступности узлов сети
- browser poller — процесс опроса для проверок браузерных элементов данных
- configuration syncer — процесс для управления кэшем данных конфигурации в памяти
- configuration syncer worker — процесс для разрешения и синхронизации значений пользовательских макросов в именах элементов данных
- connector manager — управляющий процесс для коннекторов
- connector worker — процесс для обработки запросов от connector manager
- discovery manager — управляющий процесс для обнаружения устройств
- discovery worker — процесс для обработки задач обнаружения от discovery manager
- escalator — процесс для эскалации действий
- ha manager — процесс для управления высокой доступностью
- history poller — процесс для обработки вычисляемых проверок, требующих подключения к базе данных
- history syncer — процесс записи истории в БД
- housekeeper — процесс для удаления устаревших данных (истории и трендов элементов данных, пользовательских сессий, 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 процессов опроса IPMI
- ipmi poller — процесс опроса для проверок IPMI
- java poller — процесс опроса для проверок Java
- lld manager — управляющий процесс задач 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 предобработки с рабочими потоками предобработки
- preprocessing worker — поток для предобработки данных
- proxy poller — процесс опроса для пассивных прокси
- proxy group manager — менеджер балансировки нагрузки прокси и высокой доступности
- report manager- менеджер задач генерации отчётов по расписанию
- report writer — процесс для генерации отчётов по расписанию
- self-monitoring — процесс для сбора внутренней статистики сервера
- service manager — процесс для управления сервисами путём получения информации о проблемах, тегах проблем и восстановлении проблем от history syncer, task manager и alert manager
- snmp poller — асинхронный процесс опроса для проверок SNMP с рабочим потоком (только элементы данных walk[OID] и get[OID])
- snmp trapper — trapper для SNMP-ловушек
- task manager — процесс для удалённого выполнения задач, запрошенных другими компонентами (например, закрытие проблемы, подтверждение проблемы, немедленная проверка значения элемента данных, функциональность удалённых команд)
- timer — таймер для обработки периодов обслуживания
- trapper — trapper для активных проверок, ловушек, взаимодействия с прокси
- trigger housekeeper — процесс для удаления проблем и событий, созданных триггерами, которые впоследствии были удалены
- unreachable poller — процесс опроса для недоступных устройств
- vmware collector — сборщик данных VMware, отвечающий за сбор данных от сервисов VMware

Для наблюдения за этими типами процессов можно использовать файл журнала сервера.

Файл журнала сервера создаётся с правами на чтение и запись только для владельца файла. Кроме того, файл доступен для чтения группе владельца. Все остальные права запрещены.

Различные типы процессов сервера Zabbix можно отслеживать с помощью внутреннего **zabbix[process,<type>,<mode>,<state>]** элемента данных.

Статистика транзакций синхронизатора истории

Заголовок процесса синхронизатора истории отображает подробную статистику о транзакциях синхронизатора истории:

```
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 элементов данных (состояние, ошибки, инвентарные данные узла сети и т. д.);
- Время, затраченное на сброс трендов в базу данных;
- Время, затраченное на вычисление триггеров;
- Время, затраченное на обработку событий и действий.

Процедура очистки устаревших данных

Процесс housekeeper периодически удаляет устаревшие данные (историю и тренды элементов данных, пользовательские сессии, события и т. д.), а также данные, оставшиеся после удаления объектов. Он работает циклически, при этом частота запуска и лимит удаления за один цикл определяются параметрами HousekeepingFrequency и MaxHousekeeperDelete. Любые данные, не удалённые за один цикл, переносятся на следующий. Автоматическую очистку устаревших данных можно включить и настроить в разделе *Администрирование* > *Очистка устаревших данных*.

Для удаления данных, оставшихся после удаления объектов, процесс housekeeper использует задачи из таблицы housekeeper, которая заполняется каждый раз при удалении объекта. Например, при удалении узла сети Zabbix также 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 - сервис; 3 - обнаруженный узел сети; 4 - обнаруженный сервис)
- objectid - ID объекта (помогает housekeeper находить данные, связанные с объектом)

Например, удаление узла сети с двумя элементами данных и одним триггером заполняет таблицу housekeeper следующим образом:

```
+---------------+--------+----------+
| housekeeperid | object | objectid |
+---------------+--------+----------+
|             1 |      1 |    28724 |
|             2 |      0 |    59396 |
|             3 |      0 |    59397 |
+---------------+--------+----------+
```

Триггеры базы данных заполняют таблицу housekeeper без проверки наличия данных, связанных с объектом; эта проверка выполняется процессом housekeeper.

Каждая задача приводит к одной или нескольким операциям housekeeper, которые зависят от типа объекта:

- Для элементов данных (включая правила LLD) — удаляются данные из всех таблиц истории и трендов (history, history_str, history_log, history_uint, history_text, history_bin, history_json, trends, trends_uint), содержащих значения для этих элементов данных. Также проверяется таблица problems и удаляются устаревшие внутренние события, связанные с этими элементами данных.
- Для триггеров — проверяются таблицы, связанные с событиями (problem, event_symptom, event_recovery, events), и удаляются устаревшие события, связанные с этими триггерами, а также процесс service manager уведомляется об удалённых событиях.

Note:
Отдельный процесс trigger housekeeper выполняет более узкую задачу — удаление проблем и событий, для которых неизвестен исходный триггер. Частота его запуска управляется параметром ProblemHousekeepingFrequency.

 До запуска процедуры очистки триггеров проблемы, вызванные триггерами, которые к этому моменту уже были удалены, всё ещё могут создавать проблемы сервисов и назначать их сервисам. Если в вашей среде используется много правил расчёта статуса сервисов, основанных на часто обнаруживаемых/пропадающих триггерах, рассмотрите возможность увеличения частоты процедуры очистки, изменив параметр конфигурации сервера ProblemHousekeepingFrequency.

- Для сервисов — проверяется таблица problems и удаляются устаревшие события сервисов, а также устаревшие проблемы сервисов, тем самым они закрываются в момент выполнения очистки.
- Для сетевого обнаружения — удаляются устаревшие события обнаружения из таблицы problems.

Housekeeper удаляет только те события, которые не связаны с проблемами. Например, устаревшее событие проблемы/восстановлени 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` (секционированные таблицы TimescaleDB), пропускаются; обрабатываются только таблицы, использующие режим `regular`.

Поддерживаемые платформы

Из-за требований безопасности и критически важного характера работы сервера только UNIX является операционной системой, которая может стабильно обеспечивать необходимую производительность, отказоустойчивость и устойчивость к сбоям. Zabbix работает на ведущих версиях, представленных на рынке.

Сервер Zabbix протестирован на следующих платформах:

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

Note:
Zabbix также может работать и на других Unix-подобных операционных системах.

Локаль

Обратите внимание, что серверу требуется локаль UTF-8, чтобы некоторые текстовые элементы данных могли интерпретироваться корректно. В большинстве современных Unix-подобных систем локаль UTF-8 используется по умолчанию, однако существуют системы, где её необходимо задать явно.

1. Кластер высокой доступности

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 как узла кластера

Для запуска сервера Zabbix как узла кластера в configuration 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`), по которому сервер будет использоваться в конфигурациях агента и прокси. Если не указать HANodeName, 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.

После внесения изменений в файлы конфигурации перезапустите все серверы Zabbix. Теперь они будут запущены как узлы кластера. Новый статус серверов можно увидеть в *Reports* → *System information* а также с помощью команды:

```
zabbix_server -R ha_status
```

Эта runtime-команда запишет текущий статус HA-кластера в журнал сервера Zabbix (и 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 адрес:порт сервера Zabbix **не указан** в конфигурации веб-интерфейса (находится в `conf/zabbix.conf.php` в каталоге файлов веб-интерфейса).

```
// 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, считывая настройки из таблицы nodes в базе данных Zabbix. Адрес узла активного узла будет использоваться как адрес сервера Zabbix.

Настройка прокси

Ноды (серверы) HA кластера должны быть перечислены в конфигурации как пассивных, так и активных Zabbix прокси.

В случае пассивных прокси имена нод необходимо перечислить в параметре Server этих прокси, разделяя имена **запятой**.

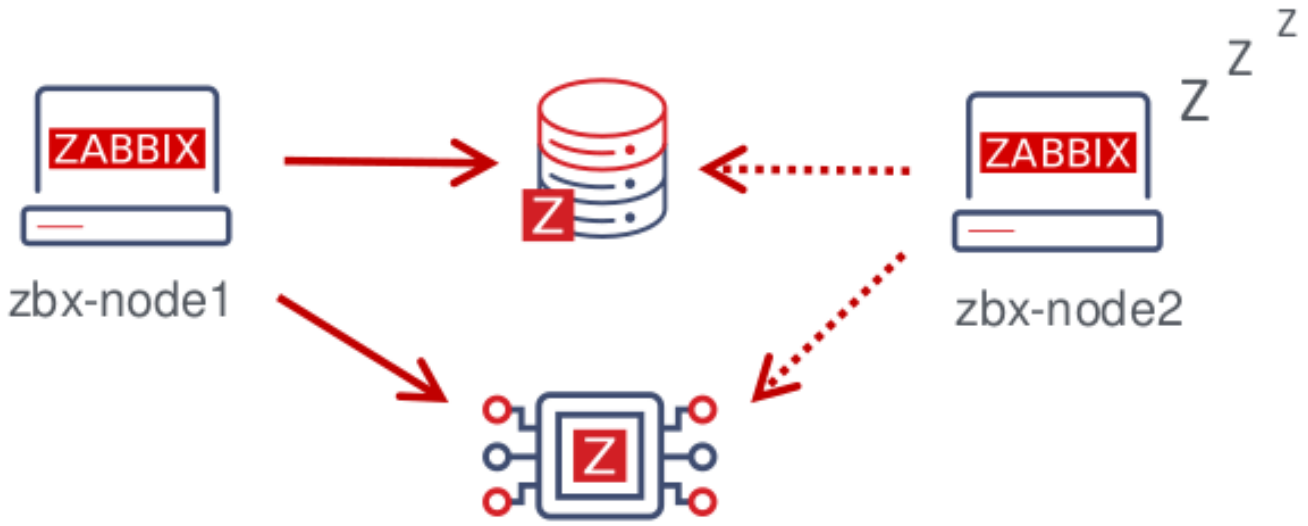
```
Server=zabbix-node-01,zabbix-node-02
```

В случае активных прокси имена нод необходимо перечислить в параметре Server этих прокси, разделяя имена **точкой с запятой**.

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

Настройка агента

Ноды (серверы) HA кластера необходимо перечислить в конфигурации Zabbix агента или Zabbix агента 2.



Чтобы активировать пассивные проверки, имена нод необходимо перечислить в параметре Server, разделяя имена нод **запятой**.

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

Чтобы активировать активные проверки, имена нод необходимо перечислить в параметре ServerActive. Обратите внимание, что для работы активных проверок ноды необходимо отделить от любых других серверов запятой, тогда как сами ноды разделяются (друг от друга) **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 HA-кластером

Текущий статус HA-кластера можно контролировать с помощью специальных параметров runtime control:

- `ha_status` — записать статус HA-кластера в журнал сервера Zabbix (и в stdout)
- `ha_remove_node=target` — удалить узел HA, идентифицируемый по <target> — имени или ID узла (имя/ID можно получить из вывода команды ha_status), например:

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

Обратите внимание, что активные/резервные узлы нельзя удалить.

- `ha_set_failover_delay=delay` — установить задержку переключения HA (от 10 секунд до 15 минут; поддерживаются суффиксы времени, например 10s, 1m)

Статус узла можно отслеживать:

- в *Отчёты* → *Системная информация*

- в виджете панели мониторинга *Системная информация*
- с помощью параметра runtime control `ha_status` сервера (см. выше).

Внутренний элемент данных `zabbix[cluster,discovery,nodes]` можно использовать для обнаружения узлов, так как он возвращает JSON с информацией об узлах высокой доступности.

Отключение HA-кластера

Чтобы отключить кластер высокой доступности:

- создайте резервные копии файлов конфигурации
- остановите резервные узлы
- удалите параметр HANodeName на активном основном сервере
- перезапустите основной сервер (он запустится в автономном режиме)

Обновление кластера высокой доступности

Чтобы выполнить обновление основной (major) версии нод высокой доступности:

- остановите все ноды;
- создайте полную резервную копию базы данных;
- если база данных использует репликацию, убедитесь, что все узлы синхронизированы и не имеют проблем. Не обновляйтесь, если репликация нарушена.
- выберите один узел, который будет выполнять обновление базы данных, измените его конфигурацию на автономный режим, закомментировав HANodeName и обновите его;
- убедитесь, что обновление базы данных полностью завершено (*Информация о системе* должна отображать, что сервер Zabbix работает);
- перезапустите ноду в режиме HA;
- 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.

2. Агент

Обзор

Агент Zabbix разворачивается на целевом объекте мониторинга для активного мониторинга локальных ресурсов и приложений (жесткие диски, память, статистика процессора и т. д.).

Агент собирает рабочую информацию локально и передает данные на сервер Zabbix для дальнейшей обработки. В случае сбоев (например, при переполнении жесткого диска или аварийном завершении процесса службы) сервер Zabbix может активно оповещать администраторов конкретной машины, сообщившей о сбое.

Агенты Zabbix отличаются высокой эффективностью благодаря использованию собственных системных вызовов для сбора статистической информации.

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 с сервера список элементов мониторинга (загрузка CPU, доступная память и т. д.), затем собирает необходимые данные и периодически отправляет ему новые значения.

Тип проверки агента настраивается выбором соответствующего типа элемента данных. Агент Zabbix обрабатывает элементы данных типа «Агент Zabbix» или «Агент Zabbix (активный)».

Поддерживаемые платформы

Список поддерживаемых платформ приведён на странице Требования.

Агент в UNIX-подобных системах

Агент Zabbix в UNIX-подобных системах запускается на контролируемом узле сети.

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 приведенным инструкциям.
- Исходный код Zabbix — загрузите исходные файлы и скомпилируйте агент Zabbix, настроив исходный код с параметром `--enable-agent`.

Attention:
В целом, 32-разрядные агенты Zabbix будут работать в 64-разрядных системах, но в некоторых случаях это может не сработать.

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.

Если установлен из пакета

Агент Zabbix работает как процесс-демон. Агент можно запустить, выполнив:

```
systemctl start zabbix-agent
```

Это будет работать в большинстве систем GNU/Linux. В других системах может потребоваться выполнить:

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

Чтобы остановить, перезапустить или проверить состояние агента Zabbix, используйте следующие команды:

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

Запуск вручную

Вы можете запустить агент Zabbix, найдя исполняемый файл `zabbix_agentd` и запустив его напрямую; например:

```
zabbix_agentd
```

Агент на Windows системах

Zabbix агент на Windows работает как служба Windows.

Установка

Агент Zabbix можно установить в Windows одним из следующих способов:

- Предварительно скомпилированные бинарные файлы агента Zabbix — загрузите пакет установщика MSI агента Zabbix и следуйте инструкциям на странице Установка агента Windows из MSI.
- Исходный код Zabbix — загрузите исходные файлы и следуйте инструкциям на странице Сборка агента Zabbix в Windows.

Дополнительные сведения об установке агента Zabbix (из ZIP-архива) как службы Windows см. на странице Агент Zabbix в Microsoft Windows.

Параметры

Можно запускать несколько экземпляров агента на узле сети. Один экземпляр 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).

Следующие параметры командной строки можно использовать с Zabbix агентом:

Parameter	Description
Агент для UNIX и Windows	
-c --config <config-file>	Путь к файлу конфигурации. Этот параметр можно использовать для указания файла конфигурации, отличного от используемого по умолчанию. В UNIX по умолчанию используется /usr/local/etc/zabbix_agentd.conf или путь, заданный переменными compile-time *--sysconfdir* или *--prefix* В Windows по умолчанию используется C:\Program Files\Zabbix Agent\zabbix_agentd.conf
-f --foreground	Запустить Zabbix агент на переднем плане (по умолчанию: true).
-p --print	Вывести известные элементы данных и завершить работу. *Примечание*: чтобы также возвращались результаты пользовательских параметров, необходимо указать файл конфигурации (если он не находится в расположении по умолчанию).
-t --test <item key>	Проверить указанный элемент данных и завершить работу. *Примечание*: чтобы также возвращались результаты пользовательских параметров, необходимо указать файл конфигурации (если он не находится в расположении по умолчанию).
-T --test-config	Проверить файл конфигурации и завершить работу.
-h --help	Показать справочную информацию.
-V --version	Показать номер версии.
Только UNIX агент	
-R --runtime-control <option>	Выполнить административные функции. См. runtime control.
Только Windows агент	
-m --multiple-agents	Использовать несколько экземпляров агента (с параметрами -i, -d, -s, -x). Чтобы различать имена служб экземпляров, в имя каждой службы будет включено значение Hostname из указанного файла конфигурации.
-S --startup-type <value>	Установить тип запуска службы Zabbix Windows агента. Допустимые значения: `automatic` - *(по умолчанию)* автоматически запускать службу при старте Windows; `delayed` - отложить запуск службы до завершения запуска служб, запускаемых автоматически (доступно в Windows Server 2008/Vista и более поздних версиях); `manual` - запускать службу вручную (пользователем или приложением); `disabled` - отключить службу, чтобы она не могла быть запущена пользователем или приложением. Этот параметр можно использовать вместе с параметром -i или отдельно для изменения типа запуска уже установленной службы.
-i --install	Установить Zabbix Windows агент как службу.
-d --uninstall	Удалить службу Zabbix Windows агента.
-s --start	Запустить службу Zabbix Windows агента.
-x --stop	Остановить службу Zabbix Windows агента.

Конкретные **примеры** использования параметров командной строки:

- вывод всех встроенных элементов данных агента со значениями
- проверка пользовательского параметра с ключом "mysql.ping", определённого в указанном файле конфигурации
- установка службы "Zabbix Agent" для Windows с использованием пути по умолчанию к файлу конфигурации C:\Program Files\Zabbix Agent\zabbix_agentd.conf
- установка службы "Zabbix Agent [Hostname]" для Windows с использованием файла конфигурации zabbix_agentd.conf, расположенного в той же папке, что и исполняемый файл агента, и обеспечение уникальности имени службы путём добавления значения Hostname из файла конфигурации
- изменение типа запуска установленной службы "Zabbix Agent" для Windows с использованием файла конфигурации zabbix_agentd.conf, 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 вы можете изменить уровень журналирования процессов агента.

Параметр	Описание	Цель
log_level_increase[=<target>]	Увеличить уровень журналирования. Если цель не указана, будут затронуты все процессы.	Цель может быть указана как: **тип процесса** — все процессы указанного типа (например, listener) См. все типы процессов агента. **тип процесса,N** — тип процесса и номер (например, listener,3) **pid** — идентификатор процесса (от 1 до 65535). Для больших значений укажите цель как 'process-type,N'.
log_level_decrease[=<target>]	Уменьшить уровень журналирования. Если цель не указана, будут затронуты все процессы.	
userparameter_reload	Перезагрузить значения параметров *UserParameter* и *Include* из текущего файла конфигурации.	

Примеры:

- увеличение уровня журналирования всех процессов
- увеличение уровня журналирования третьего процесса listener
- увеличение уровня журналирования процесса с PID 1234
- уменьшение уровня журналирования 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

- `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 создаётся с правами на чтение и запись только для владельца файла. Дополнительно файл доступен для чтения группе владельца. Все остальные права запрещены.

Пользователь процесса

Агент Zabbix в UNIX предназначен для запуска от имени пользователя без прав root. Он будет работать от имени того пользователя без прав root, от которого был запущен. Поэтому вы можете без каких-либо проблем запускать агент от имени любого пользователя без прав root.

Если вы попытаетесь запустить его от имени 'root', он переключится на жестко заданного пользователя 'zabbix', который должен присутствовать в вашей системе. Вы можете запускать агент от имени 'root' только в том случае, если соответствующим образом измените параметр 'AllowRoot' в конфигурационном файле агента.

Файл конфигурации

Подробную информацию о настройке Zabbix агент см. в параметрах файла конфигурации для zabbix_agentd или агент для Windows.

Локаль

Обратите внимание, что агенту требуется локаль UTF-8, чтобы некоторые текстовые элементы данных агента могли возвращать ожидаемое содержимое. В большинстве современных Unix-подобных систем локаль UTF-8 используется по умолчанию, однако существуют системы, где её необходимо задать явно.

Код завершения процесса

Zabbix агент возвращает 0 в случае корректного завершения процесса и 1 в случае отказа/аварийного завершения процесса.

3. Агент 2

Обзор

Zabbix агент 2 — это новое поколение Zabbix агента, написанное на Go (с повторным использованием части кода на C из Zabbix агента). Он предназначен для того, чтобы:

- Сократить 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 скриптов и автономного сбора данных с периодической отправкой отчетов.
- Работать в качестве замены Zabbix агента, поддерживая все предыдущие возможности.

Пассивные и активные проверки

Zabbix агент 2 поддерживает пассивные и активные проверки, аналогично Zabbix агенту. Кроме того, активные проверки Zabbix агента 2 поддерживают гибкие интервалы/интервалы по расписанию и параллельное выполнение проверок в рамках одного активного сервера.

Note:
По умолчанию после перезапуска Zabbix агент 2 планирует первый сбор данных для активных проверок на условно случайное время в пределах интервала обновления элемента данных, чтобы предотвратить всплески использования ресурсов. Чтобы выполнять активные проверки, для которых *Scheduling* интервал обновления сразу после перезапуска агента, задайте параметр `ForceActiveChecksOnStart` (на глобальном уровне) или `Plugins.<Plugin name>.System.ForceActiveChecksOnStart` (влияет только на проверки конкретного плагина) в файле конфигурации. Параметр уровня плагина, если он задан, 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 параметрах конфигурации *плагина*.

Поддерживаемые платформы

Список поддерживаемых платформ приведён на странице Требования.

Агент 2 в UNIX-подобных системах

Zabbix агент 2 в UNIX-подобных системах запускается на контролируемом узле сети.

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.
- Исходный код Zabbix — загрузите исходные файлы и скомпилируйте агент, настроив его с параметром `--enable-agent2`.

Note:
Возможности мониторинга Zabbix агента 2 можно расширить с помощью загружаемых плагинов, которые доступны отдельно. Подробнее см. в разделе Загружаемые плагины.

Если установлен как пакет

Zabbix агент 2 работает как процесс переднего плана и использует внешний менеджер служб (например, systemd) для выполнения в фоновом режиме; Zabbix агент 2 не имеет встроенной поддержки демонизации в Linux.

Агент можно запустить, выполнив:

```
systemctl start zabbix-agent2
```

Чтобы остановить, перезапустить или проверить состояние Zabbix агента 2, используйте следующие команды:

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

Запуск вручную

Вы можете запустить агент Zabbix, найдя исполняемый 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 агент 2 можно установить в Windows одним из следующих способов:

- Предварительно скомпилированные бинарные файлы Zabbix агента — загрузите пакет MSI-установщика агента и следуйте инструкциям на странице Установка Windows агента из MSI.
- Исходный код Zabbix — загрузите исходные файлы и следуйте инструкциям на странице Сборка Zabbix агента 2 в Windows.

Note:
Возможности мониторинга Zabbix агента 2 можно расширить с помощью загружаемых плагинов, которые доступны отдельно. Подробнее см. на странице Загружаемые плагины.

Дополнительные сведения об установке Zabbix агента 2 (из ZIP-архива) как службы Windows см. на странице Zabbix агент в Microsoft Windows.

Параметры

Следующие параметры командной строки можно использовать с Zabbix агент 2:

Параметр	Описание
UNIX и Windows агент	
-c --config <config-file>	Путь к файлу конфигурации. Вы можете использовать этот параметр, чтобы указать файл конфигурации, отличный от используемого по умолчанию. В UNIX по умолчанию используется /usr/local/etc/zabbix_agent2.conf или путь, заданный переменными compile-time *--sysconfdir* или *--prefix* В Windows по умолчанию используется C:\Program Files\Zabbix Agent 2\zabbix_agent2.conf
-f --foreground	Запустить Zabbix агент в режиме переднего плана (по умолчанию: true).

Параметр	Описание
-p --print	Вывести известные элементы данных и завершить работу. *Примечание*: чтобы также возвращать результаты пользовательских параметров, необходимо указать файл конфигурации (если он не находится в расположении по умолчанию).
-t --test <item key>	Проверить указанный элемент данных и завершить работу. *Примечание*: чтобы также возвращать результаты пользовательских параметров, необходимо указать файл конфигурации (если он не находится 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 версию агента и информацию о лицензии.
-R --runtime-control <option>	Выполнить административные функции. См. runtime control.
Только Windows агент	
-m --multiple-agents	Использовать несколько экземпляров агента (с параметрами -i, -d, -s, -x). Чтобы различать имена служб экземпляров, в каждое имя службы будет включено значение Hostname из указанного файла конфигурации.
-S --startup-type <value>	Установить тип запуска службы Zabbix Windows agent. Допустимые значения: `automatic` - *(по умолчанию)* запускать службу автоматически при запуске Windows; `delayed` - отложить запуск службы до завершения запуска служб, запускаемых автоматически; `manual` - запускать службу вручную (пользователем или приложением); `disabled` - отключить службу, чтобы ее нельзя было запустить пользователем или приложением. Вы можете использовать этот параметр вместе с параметром -i или отдельно, чтобы изменить тип запуска уже установленной службы.
-i --install	Установить Zabbix Windows agent как службу.
-d --uninstall	Удалить службу Zabbix Windows agent.
-s --start	Запустить службу Zabbix Windows agent.
-x --stop	Остановить службу Zabbix Windows agent.

Конкретные **примеры** использования параметров командной строки:

- вывести все встроенные элементы данных агента со значениями
- проверить пользовательский параметр с ключом "mysql.ping", определенный в указанном файле конфигурации
- установить службу "Zabbix Agent" для Windows, используя путь по умолчанию к файлу конфигурации C:\Program Files\Zabbix Agent 2\zabbix_agent2.conf
- изменить тип запуска установленной службы "Zabbix Agent" для Windows, используя файл конфигурации zabbix_agent2.conf, расположенный в той же папке, что и исполняемый файл агента

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

Управление работой

Управление работой агента предоставляет некоторые опции для удалённого управления агентом.

Опция	Описание
log_level_increase	Увеличение уровня журналирования.
log_level_decrease	Уменьшение уровня журналирования.
metrics	Список доступных метрик.
version	Отображение версии агента.
userparameter_reload	Загрузка заново значений параметров *UserParameter* и *Include* из текущего файла конфигурации.
help	Отображение справочной информации о функции управления работой.

Примеры:

- увеличение уровня журналирования агента 2
- вывод опций управления работой

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

Файл конфигурации

Параметры конфигурации агент 2 в основном совместимы с Zabbix агентом, за некоторыми исключениями.

Новые параметры	Описание
ControlSocket	Путь к управляющему сокету времени выполнения. Агент 2 использует управляющий сокет для команд времени выполнения.
EnablePersistentBuffer, *PersistentBufferFile*, *PersistentBufferPeriod*	Эти параметры используются для настройки постоянного хранилища в агент 2 для активных элементов данных.
ForceActiveChecksOnStart	Определяет, должен ли агент выполнять активные проверки сразу после перезапуска или равномерно распределять их во времени.
Plugins	У плагинов могут быть собственные параметры в формате `Plugins.<Plugin name>.<Parameter>=<value>`. Общим параметром плагинов является *System.Capacity*, задающий предел проверок, которые могут выполняться одновременно.
StatusPort	Порт, на котором агент 2 будет ожидать HTTP-запрос статуса и отображать список настроенных плагинов и некоторые внутренние параметры
Удалённые параметры	**Описание**
AllowRoot, *User*	Не поддерживаются, поскольку поддержка демонизации отсутствует.
LoadModule, *LoadModulePath*	Загружаемые модули не поддерживаются.
StartAgents	Этот параметр использовался в Zabbix агенте для увеличения параллелизма пассивных проверок или их отключения. В Агент 2 параллелизм настраивается на уровне плагина и может быть ограничен параметром capacity.

Для получения более подробной информации смотрите параметры файла конфигурации для zabbix_agent2.

Коды завершения

Zabbix агент 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 не удалась, то в стандартный поток ошибок (STDERR) выводится сообщение об ошибке, и агент 2 завершает работу, возвращая код 2 или 3 соответственно.

Agent vs agent 2 comparison

This section describes the differences between the Zabbix agent and the Zabbix agent 2.

Parameter	Zabbix agent	Zabbix agent 2
Programming language	C	Go with some parts in C
Daemonization	yes	by systemd only (yes on Windows)
Supported extensions	Custom loadable modules in C.	Custom plugins in Go.
Requirements		
Supported platforms	Linux, IBM AIX, FreeBSD, NetBSD, OpenBSD, HP-UX, Mac OS X, Solaris: 9, 10, 11, Windows: all desktop and server versions since XP	Linux, Windows: all desktop and server versions, on which a supported Go version can be installed.
Supported crypto libraries	GnuTLS 3.1.18 and newer OpenSSL 1.0.1, 1.0.2, 1.1.0, 1.1.1, 3.0.x LibreSSL - tested with versions 2.7.4, 2.8.2 (certain limitations apply, see the Encryption page for details).	Linux: OpenSSL 1.0.1 and later. MS Windows: OpenSSL 1.1.1 or later. The OpenSSL library must have PSK support enabled. LibreSSL is not supported.
Monitoring processes		

Parameter	Zabbix agent	Zabbix agent 2
Processes	A separate active check process for each server/proxy record.	Single process with automatically created threads. The maximum number of threads is determined by the GOMAXPROCS environment variable.
Metrics	**UNIX**: see a list of supported items. **Windows**: see a list of additional Windows-specific items.	**UNIX**: All metrics supported by Zabbix agent. Additionally, the agent 2 provides Zabbix-native monitoring solution for: Docker, Memcached, MySQL, PostgreSQL, Redis, systemd, and other monitoring targets - see a full list of agent 2 specific items. **Windows**: All metrics supported by Zabbix agent, and also net.tcp.service* checks of HTTPS, LDAP. Additionally, the agent 2 provides Zabbix-native monitoring solution for: PostgreSQL, Redis.
Concurrency	Active checks for single server are executed sequentially.	Checks from different plugins or multiple checks within one plugin can be executed concurrently.
Third-party traps	no	yes
Additional features		
Persistent storage	no	yes
Persistent files for log*[] metrics	yes (only on Unix)	no
Log data upload	Can be performed during log gathering to free the buffer.	Log gathering is stopped when the buffer is full, therefore the BufferSize parameter must be at least MaxLinesPerSecond x 2.
Changes user at runtime	yes (Unix-like systems only)	no (controlled by systemd)
User-configurable ciphersuites	yes	no

See also:

- *Zabbix processes description*: Zabbix agent, Zabbix agent 2
- *Configuration parameters*: Zabbix agent UNIX / Windows, Zabbix agent 2 UNIX / Windows

Built-in plugins

These plugins are built into Zabbix agent 2 and are available out-of-the-box. Click on the plugin name to go to the plugin repository with additional information.

Plugin name	Description	Supported item keys	Comments
Agent	Metrics of the Zabbix agent being used.	agent.hostname, agent.ping, agent.version	Supported keys have the same parameters as Zabbix agent keys.
CPU	System CPU monitoring (number of CPUs/CPU cores, discovered CPUs, utilization percentage).	system.cpu.discovery, system.cpu.num, system.cpu.util	Supported keys have the same parameters as Zabbix agent keys.

Plugin name	Description	Supported item keys	Comments
Docker	Monitoring of Docker containers.	docker.container_info, docker.container_stats, docker.containers, docker.containers.discovery, docker.data_usage, docker.images, docker.images.discovery, docker.info, docker.ping	See also: Configuration parameters
File	File metrics collection.	vfs.file.cksum, vfs.file.contents, vfs.file.exists, vfs.file.md5sum, vfs.file.regexp, vfs.file.regmatch, vfs.file.size, vfs.file.time	Supported keys have the same parameters as Zabbix agent keys.
Kernel	Kernel monitoring.	kernel.maxfiles, kernel.maxproc	Supported keys have the same parameters as Zabbix agent keys.
Log	Log file monitoring.	log, log.count, logrt, logrt.count	Supported keys have the same parameters as Zabbix agent keys. See also: Plugin configuration parameters (Unix/Windows)
Memcached	Memcached server monitoring.	memcached.ping, memcached.stats	
Modbus	Reads Modbus data.	modbus.get	Supported keys have the same parameters as Zabbix agent keys.
MQTT	Receives published values of MQTT topics.	mqtt.get	To configure an encrypted connection to the MQTT broker, specify TLS parameters in the agent configuration file as a named session or default parameters. Currently TLS parameters cannot be passed as item key parameters.
MySQL	Monitoring of MySQL and its forks.	mysql.custom.query, mysql.db.discovery, mysql.db.size, mysql.get_status_variables, mysql.ping, mysql.replication.discovery, mysql.replication.get_slave_status, mysql.version	To configure an encrypted connection to the database, specify TLS parameters in the agent configuration file as a named session or default parameters. Currently TLS parameters cannot be passed as item key parameters.
NetIf	Monitoring of network interfaces.	net.if.collisions, net.if.discovery, net.if.in, net.if.out, net.if.total	Supported keys have the same parameters as Zabbix agent keys.

Plugin name	Description	Supported item keys	Comments
Oracle	Oracle Database monitoring.	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	Install the Oracle Instant Client before using the plugin.
Proc	Process CPU utilization percentage.	proc.cpu.util	Supported key has the same parameters as Zabbix agent key.
Redis	Redis server monitoring.	redis.config, redis.info, redis.ping, redis.slowlog.count	To configure an encrypted connection to Redis, specify TLS parameters in the agent configuration file as a named session or default parameters. TLS parameters cannot be passed as item key parameters. Note that an incorrect or otherwise invalid TLS configuration can prevent Zabbix agent 2 from starting, so verify certificate files, permissions and paths before enabling TLS.

Smart	S.M.A.R.T. monitoring.	smart.attribute.discovery, smart.disk.discovery, smart.disk.get	The minimum required smartctl version is 7.1.
			Sudo/root access rights to smartctl are required for the user executing Zabbix agent 2. The plugin uses only the following commands:
			`/usr/sbin/smartctl -a *`
			`/usr/sbin/smartctl --scan *`
			`/usr/sbin/smartctl -j -V`
			
			Supported keys can be used with Zabbix agent 2 only on Linux/Windows, both as a passive and active check.
			See also:
			Configuration parameters

SW	Listing of installed packages.	system.sw.packages, system.sw.packages.get	The supported keys have the same parameters as Zabbix agent key.
Swap	Swap space size in bytes/percentage.	system.swap.size	Supported key has the same parameters as Zabbix agent key.
SystemRun	Runs specified command.	system.run	Supported key has the same parameters as Zabbix agent key. See also: Plugin configuration parameters (Unix/Windows)

Plugin name	Description	Supported item keys	Comments
Systemd	Monitoring of systemd services.	systemd.unit.discovery, systemd.unit.get, systemd.unit.info	
TCP	TCP connection availability check.	net.tcp.port	Supported key has the same parameters as Zabbix agent key.
UDP	Monitoring of the UDP services availability and performance.	net.udp.service, net.udp.service.perf	Supported keys have the same parameters as Zabbix agent keys.
Uname	Retrieval of information about the system.	system.hostname, system.sw.arch, system.uname	Supported keys have the same parameters as Zabbix agent keys.
Uptime	System uptime metrics collection.	system.uptime	Supported key has the same parameters as Zabbix agent key.
VFSDev	VFS metrics collection.	vfs.dev.discovery, vfs.dev.read, vfs.dev.write	Supported keys have the same parameters as Zabbix agent keys.
WebCertificate	Monitoring of TLS/SSL website certificates.	web.certificate.get	
WebPage	Web page monitoring.	web.page.get, web.page.perf, web.page.regexp	Supported keys have the same parameters as Zabbix agent keys.
ZabbixAsync	Asynchronous metrics collection.	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	Supported keys have the same parameters as Zabbix agent keys.
ZabbixStats	Zabbix server/proxy internal metrics or number of delayed items in a queue.	zabbix.stats	Supported keys have the same parameters as Zabbix agent keys.
ZabbixSync	Synchronous metrics collection.	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.	Supported keys have the same parameters as Zabbix agent keys.

Loadable plugins

Loadable plugins are not available out-of-the-box for Zabbix agent 2 and need to be installed separately:

- On Linux, you can use packages (e.g., install Ember+ on Ubuntu with `apt install zabbix-agent2-plugin-ember-plus`) or build plugins
- On Windows, you can install from MSI or build plugins

Attention:
Before installing a plugin, please check its README file. It may contain specific requirements and installation instructions.

Click on the plugin name to go to the plugin repository, which contains the plugin README file with additional information.

Plugin name	Description	Supported item keys	Comments
Ceph	Ceph monitoring.	ceph.df.details, ceph.osd.stats, ceph.osd.discovery, ceph.osd.dump, ceph.ping, ceph.pool.discovery, ceph.status	See also Ceph plugin configuration parameters.
Ember+	Monitoring of Ember+.	ember.get	See also Ember+ plugin configuration parameters.
MongoDB	Monitoring of MongoDB servers and clusters (document-based, distributed database).	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	To configure encrypted connections to the database, specify TLS parameters in the agent configuration file as a named session parameters. Currently TLS parameters cannot be passed as item key parameters. See also MongoDB plugin configuration parameters.
MSSQL	Monitoring of MSSQL database.	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	To configure an encrypted connection to the database, specify TLS parameters in the agent configuration file as a named session or default parameters. Currently TLS parameters cannot be passed as item key parameters. See also MSSQL plugin configuration parameters.

Plugin name	Description	Supported item keys	Comments
NVIDIA GPU	Monitoring of NVIDIA GPU.	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	See also NVIDIA GPU plugin configuration parameters.
PostgreSQL	Monitoring of PostgreSQL and its forks.	pgsql.autovacuum.count, pgsql.archive, pgsql.bgwriter, pgsql.cache.hit, pgsql.connections, pgsql.custom.query, pgsql.dbstat, pgsql.dbstat.sum, pgsql.db.age, pgsql.db.bloating_tables, pgsql.db.discovery, pgsql.db.size, pgsql.locks, pgsql.oldest.xid, pgsql.ping, pgsql.queries, pgsql.replication.count, pgsql.replication.process, pgsql.replication.process.discovery, pgsql.replication.recovery_role, pgsql.replication.status, pgsql.replication_lag.b, pgsql.replication_lag.sec, pgsql.uptime, pgsql.version, pgsql.wal.stat	To configure encrypted connections to the database, specify TLS parameters in the agent configuration file as a named session or default parameters. Currently TLS parameters cannot be passed as item key parameters. See also PostgreSQL plugin configuration parameters.

Note:
Loadable plugins, when launched with:
 - `-V --version` - print plugin version and license information;
 - `-h --help` - print help information.
 - `-t, --test <item key>` — launch plugin for testing (plugin config ignored).

4. Прокси

Обзор

Прокси Zabbix — это процесс, который может собирать данные мониторинга с одного или нескольких контролируемых устройств и отправлять эту информацию на сервер Zabbix, фактически работая от имени сервера. Все собранные данные буферизуются локально, а затем передаются на сервер Zabbix, к которому относится прокси.

Развертывание прокси не является обязательным, однако может быть очень полезным для распределения нагрузки одного сервера Zabbix. Если данные собирают только прокси, обработка на сервере требует меньше ресурсов CPU и дискового ввода-вывода.

Прокси Zabbix — это идеальное решение для централизованного мониторинга удаленных площадок, филиалов и сетей без локальных администраторов.

Прокси Zabbix требует отдельную базу данных.

> **Attention:**
> Обратите внимание, что базы данных, поддерживаемые прокси Zabbix, — это SQLite, MySQL и PostgreSQL.

См. также: Использование прокси в распределенной среде

Запуск прокси

Если установлен из пакета

Прокси Zabbix работает как процесс-демон. Прокси можно запустить, выполнив:

```
systemctl start zabbix-proxy
```

Это будет работать в большинстве систем GNU/Linux. В других системах может потребоваться выполнить:

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

Аналогично, для остановки/перезапуска/просмотра состояния прокси Zabbix используйте следующие команды:

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

Запуск вручную

Если описанное выше не работает, необходимо запустить его вручную. Найдите путь к исполняемому файлу zabbix_proxy и выполните:

```
zabbix_proxy
```

С прокси Zabbix можно использовать следующие параметры командной строки:

```
-c --config <file>
-f --foreground                         Zabbix
-R --runtime-control <option>
-T --test-config
-h --help
-V --version
```

Примеры запуска прокси Zabbix 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:

Параметр	Описание	Цель
config_cache_reload	Перезагрузить кэш конфигурации. Игнорируется, если кэш загружается в данный момент. 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 предобработки **locks** - список mutex'ов (пуст на системах *BSD*)
snmp_cache_reload	Перезагрузить кэш SNMP — очистить свойства SNMP engine (engine time, engine boots, engine id, credentials) для всех узлов сети. Используйте для принудительной глобальной очистки кэша при устранении проблем с SNMP.	
housekeeper_execute	Запустить процедуру очистки устаревших данных. Игнорируется, если процедура очистки устаревших данных уже выполняется.	
log_level_increase[=<**target**>]	Увеличить уровень журналирования; если target не указан, затрагиваются все процессы. 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 (от 1 до 65535). Для больших значений указывайте target как 'process type,N'.
log_level_decrease[=<**target**>]	Уменьшить уровень журналирования; если target не указан, затрагиваются все процессы. Не поддерживается в системах *BSD*.	
prof_enable[=<**target**>]	Включить профилирование. Если target не указан, затрагиваются все процессы. При включенном профилировании предоставляются 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 (от 1 до 65535). Для больших значений указывайте 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. Если target не указан, затрагиваются все процессы.	**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 (от 1 до 65535). Для больших значений указывайте target как 'process type,N'.

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 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 завершается ошибкой после изменения учетных данных (например, из-за несоответствий engineBoots/engineID или устройств, не соответствующих RFC), либо когда необходимо принудительно выполнить глобальную очистку кэша 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"
```

Пользователь процесса

Прокси Zabbix разработан для запуска от имени непривилегированного пользователя. Он будет работать от имени того непривилегированного пользователя, от которого был запущен. Таким образом, вы можете без каких-либо проблем запускать прокси от имени любого непривилегированного пользователя.

Если вы попытаетесь запустить его от имени 'root', он переключится на жестко заданного пользователя 'zabbix', который должен присутствовать в вашей системе. Вы можете запускать прокси от имени 'root' только в том случае, если соответствующим образом измените параметр 'AllowRoot' в файле конфигурации прокси.

Файл конфигурации

Подробные сведения о настройке zabbix_proxy см. в разделе configuration file.

Типы процессов и потоков прокси

- `agent poller` — асинхронный процесс опроса для пассивных проверок с рабочим потоком
- `availability manager` — процесс обновления доступности узла сети
- `browser poller` — процесс опроса для проверок browser-элементов данных
- `configuration syncer` — процесс управления кэшем данных конфигурации в памяти
- `data sender` — отправитель данных прокси
- `discovery manager` — управляющий процесс обнаружения устройств
- `discovery worker` — процесс обработки задач обнаружения от discovery manager
- `history syncer` — процесс записи истории в БД
- `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` — управляющий процесс опроса 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` — управляющий процесс задач предобработки с рабочими потоками предобработки
- `preprocessing worker` — поток предобработки 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` — trapper для SNMP-ловушек
- `task manager` — процесс удаленного выполнения задач, запрошенных другими компонентами (например, закрытие проблемы, подтверждение проблемы, немедленная проверка значения элемента данных, функциональность удаленных команд)
- `trapper` — trapper для активных проверок, ловушек, взаимодействия с прокси
- `unreachable poller` — процесс опроса для недоступных устройств
- `vmware collector` — сборщик данных VMware, отвечающий за получение данных из сервисов VMware

Файл журнала прокси можно использовать для наблюдения за этими типами процессов.

Файл журнала прокси создается с правами на чтение и запись только для владельца файла. Дополнительно файл доступен для чтения группе владельца. Все остальные права запрещены.

Различные типы процессов Zabbix proxy можно отслеживать с помощью внутреннего элемента данных **zabbix[process,<type>,<mo**

Статистика транзакций history syncer

Заголовок процесса history syncer отображает подробную статистику о транзакциях 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 показатели в строке "processed...in N (<timings>) sec" означают:

- Время, затраченное на запись значений элементов данных в базу данных;
- Время, затраченное на обновление данных элементов данных (состояние, ошибки).

Процедура очистки устаревших данных

Прокси Zabbix имеет процесс `housekeeper`, который удаляет устаревшую историю и тренды элементов данных. Он работает циклически; частота определяется параметром `HousekeepingFrequency`, а лимит удаления за один цикл — параметрами `ProxyLocalBuffer` и `ProxyOfflineBuffer`. В отличие от процедуры очистки устаревших данных на сервере Zabbix, процесс `housekeeper` на 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, обнаружение сети, авторегистрацию узлов сети) в буфере и отправлять их на сервер Zabbix без обращения к базе данных. Буфер памяти был добавлен для прокси начиная с Zabbix 7.0.

В установках до Zabbix 7.0 собранные данные сохранялись в базе данных перед отправкой на сервер Zabbix. Для таких установок это поведение остается используемым по умолчанию после обновления до Zabbix 7.0.

Для оптимальной 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 режиме hybrid буфер защищен от потери данных за счет сброса неотправленных данных в базу данных, если прокси остановлен, буфер заполнен или данные слишком старые. Когда все значения будут сброшены в базу данных, прокси возвращается к использованию буфера памяти.

В режиме memory будет использоваться буфер памяти, однако защита от потери данных отсутствует. Если прокси будет остановлен или память переполнится, неотправленные данные будут отброшены.

Режим hybrid (ProxyBufferMode=hybrid) применяется ко всем новым установкам начиная с Zabbix 7.0.

Дополнительные параметры, такие как ProxyMemoryBufferSize и ProxyMemoryBufferAge, определяют соответственно размер буфера памяти и максимальный возраст данных в буфере.

Обратите внимание, что при конфликтующей конфигурации прокси выведет ошибку и не сможет запуститься, например, если:

- ProxyBufferMode установлено в ”hybrid” или ”memory”, а ProxyMemoryBufferSize — в ”0”;
- ProxyBufferMode установлено в ”hybrid” или ”memory”, а ProxyLocalBuffer — не в ”0”.

Локаль

Обратите внимание, что прокси требует локаль UTF-8, чтобы некоторые текстовые элементы данных могли интерпретироваться корректно. В большинстве современных Unix-подобных систем локаль UTF-8 используется по умолчанию, однако существуют системы, где её необходимо задать явно.

Расчет очередей во время обслуживания

Attention:
Прокси Zabbix не знает о периодах обслуживания; подробности см. в разделе Расчет очередей во время обслуживания.

5. Java gateway

Обзор

Zabbix Java gateway можно установить из исходного кода или из пакетов.

Встроенная поддержка мониторинга JMX-приложений 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, Zabbix сервер обращается к Zabbix Java gateway, который использует API управления JMX для удаленного запроса к интересующему приложению. Для приложения не требуется установка какого-либо дополнительного программного обеспечения, оно просто должно быть запущено с параметром `-Dcom.sun.management.jmxremote` в командной строке.

Java gateway принимает входящие соединения от Zabbix сервера или прокси и может использоваться только как ”пассивный прокси”. В отличие от Zabbix proxy, он также может использоваться из Zabbix proxy (цепочки из Zabbix proxy недопустимы). Доступ к каждому Java gateway настраивается непосредственно в конфигурационном файле Zabbix сервера или прокси, поэтому для каждого Zabbix сервера или Zabbix proxy может быть настроен только один Java gateway. Если у узла сети есть элементы данных типа **JMX agent** и элементы данных других типов, только элементы данных типа **JMX agent** будут передаваться в Java gateway для получения данных.

Когда элемент данных должен быть обновлён через Java gateway, Zabbix сервер или прокси подключается к Java gateway и запрашивает значение, которое Java gateway в свою очередь извлекает и возвращает серверу или прокси. Таким образом, Java gateway никакие значения не кэширует.

У Zabbix сервера и прокси есть специальный тип процессов, которые подключаются к Java gateway, их количество настраивается опцией **StartJavaPollers**. Внутренне, Java gateway запускается в несколько потоков, что настраивается опцией **START_POLLERS**. На стороне сервера, если соединение занимает более чем **Timeout** секунд, оно будет завершено, но Java gateway может оставаться занят получением значения счетчика JMX. Чтобы решить эту проблему, в Java gateway имеется опция **TIMEOUT**, позволяющая указать время ожидания сетевых операций JMX.

Сервер Zabbix или прокси будет пытаться по возможности объединять запросы к одной цели JMX (в зависимости от интервалов элементов данных) и отправлять их в Java gateway по одному соединению для повышения производительности.

Рекомендуется, чтобы **StartJavaPollers** было меньше или равно **START_POLLERS**, иначе возможны ситуации, когда в Java gateway не будет доступных потоков для обслуживания входящих запросов; в таком случае Java gateway использует ThreadPoolExecutor.CallerRunsPolicy, что означает, что основной поток будет обслуживать входящий запрос и временно не будет принимать новые запросы.

Если вы пытаетесь выполнять мониторинг Java-приложений на базе Wildfly с помощью Zabbix Java gateway, установите последнюю версию jboss-client.jar, доступную на странице загрузки Wildfly.

2. Установка из пакетов RHEL

Обзор

Если установлено из пакетов, следующая информация поможет вам настроить Zabbix Java gateway.

Настройка и запуск Java gateway

Параметры конфигурации Zabbix Java gateway можно настроить в следующем файле:

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

Для получения более подробных сведений смотрите параметры настройки 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 более поздние:

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

RHEL до версии 7:

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

Настройка сервера для использования с Java gateway

Когда Java gateway запущен и работает, вы должны указать Zabbix серверу, где найти Zabbix Java gateway. Чтобы это сделать, укажите параметры JavaGateway и JavaGatewayPort в файле конфигурации сервера. Если же узел сети, на котором работает JMX приложение, наблюдается через Zabbix прокси, то параметры соединения указываются в файле конфигурации прокси.

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

По умолчанию, сервер не запускает процессы, связанные с мониторингом JMX. Если же вы хотите использовать этот тип мониторинга, то вам нужно указать количество экземпляров Java поллеров. Это делается таким же образом, как и для обычных поллеров и трапперов.

```
StartJavaPollers=5
```

Не забудьте перезапустить сервер или прокси после того как закончите изменение настроек.

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 `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.

3. Установка из пакетов Debian/Ubuntu

Обзор

Если установлено из пакетов, следующая информация поможет вам настроить Zabbix Java gateway.

Настройка и запуск Java gateway

Параметры конфигурации Zabbix Java gateway можно настроить в следующем файле:

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

Для получении более подробных сведений смотрите 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 с Java gateway

Когда Java gateway запущен и работает, вы должны указать Zabbix серверу, где найти Zabbix Java gateway. Чтобы это сделать, задайте параметры JavaGateway и JavaGatewayPort в файле конфигурации сервера. Если же узел сети, на котором работает приложение JMX, наблюдается через Zabbix прокси, то параметры соединения вместо этого указываются в файле конфигурации прокси.

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

По умолчанию, сервер не запускает процессы, связанные с мониторингом JMX. Если же вы хотите использовать этот тип мониторинга, то вам нужно указать количество экземпляров Java поллеров. Это делается таким же образом, как и для обычных поллеров и трапперов.

```
StartJavaPollers=5
```

Не забудьте перезапустить сервер или прокси после того как закончите изменение настроек.

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 `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 исходного кода

Обзор

Если установлен из исходного кода, следующая информация поможет вам настроить Zabbix Java gateway.

Обзор файлов

Если вы получили Java gateway из исходных кодов, у вас должен быть набор shell-скриптов, JAR- и конфигурационных файлов в каталоге $PREFIX/sbin/zabbix_java. Назначение этих файлов кратко описано ниже.

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

Собственно JAR-файл Java gateway.

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

Зависимости Java gateway: библиотека Logback, SLF4J и Android JSON.

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

Конфигурационные файлы для Logback.

`shutdown.sh`
`startup.sh`

Вспомогательные скрипты для запуска и остановки Java gateway.

`settings.sh`

Конфигурационный файл, который подключается скриптами запуска и остановки, указанными выше.

Настройка и запуск Java gateway

По умолчанию, Java gateway слушает порт 10052. Если вы планируете работу Java gateway на другом порту, то вы можете указать его в скрипте settings.sh. Смотрите описание файла конфигурации Java gateway для получения сведений о том, как указать эту и другие опции.

Warning:
Порт 10052 не зарегистрирован в IANA.

Выполнив настройки, вы можете запустить Java gateway, выполнив скрипт запуска:

```
./startup.sh
```

Точно так же, если вам более не требуется Java gateway, выполните скрипт завершения работы для остановки Java gateway:

```
./shutdown.sh
```

Обратите внимание, что в отличие от сервера или прокси, Java-шлюз является легковесным и не требует базы данных.

Настройка сервера для использования с Java gateway

Когда Java gateway запущен и работает, вы должны указать Zabbix серверу, где найти Zabbix Java gateway. Чтобы это сделать, укажите параметры JavaGateway и JavaGatewayPort в файле конфигурации сервера. Если же узел сети, на котором работает JMX приложение, наблюдается через Zabbix прокси, то параметры соединения указываются в файле конфигурации прокси.

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

По умолчанию, сервер не запускает процессы, связанные с мониторингом JMX. Если же вы хотите использовать этот тип мониторинга, то вам нужно указать количество экземпляров Java поллеров. Это делается таким же образом, как и для обычных поллеров и трапперов.

```
StartJavaPollers=5
```

Не забудьте перезапустить сервер или прокси после того как закончите изменение настроек.

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 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 для получения более подробных сведений.

Мониторинг JMX

См. страницу Мониторинг JMX для получения более подробной информации.

6. Sender

Обзор

Zabbix sender — это утилита командной строки, которая может использоваться для отправки данных производительности на сервер Zabbix для обработки.

Эта утилита обычно используется в длительно работающих пользовательских скриптах для периодической отправки данных о доступности и производительности.

Для отправки результатов напрямую на сервер Zabbix или прокси должен быть настроен элемент данных типа trapper.

См. также библиотеку Python для Zabbix, которая имеет встроенную функциональность для работы в качестве Zabbix sender.

Выполнение Zabbix sender

Пример выполнения Zabbix UNIX sender:

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

где:

- z — хост Zabbix сервера (также можно использовать IP адрес)
- s — техническое имя наблюдаемого узла сети (как указано в веб-интерфейсе Zabbix)
- k — ключ элемента данных
- o — отправляемое значение

Attention:
Если значения опций содержат пробелы, то эти значения необходимо заключить в двойные кавычки.

Zabbix sender можно использовать для отправки нескольких значений из входного файла. Обратитесь к странице помощи Zabbix sender для получения более подробной информации.

Если указывается файл конфигурации, для отправки данных Zabbix sender использует все адреса, которые указаны в параметре ServerActive конфигурации агента. Если отправка на один адрес завершится ошибкой, sender попытается отправить на другие адреса. Если отправка партии данных на один из адресов завершается ошибкой, тогда следующие партии данных отправляться на этот адрес не будут.

Zabbix sender принимает строки в кодировке UTF-8 (как в UNIX-подобных системах, так и в Windows) без начального маркера последовательности байтов (BOM) в файле.

Zabbix sender в Windows можно выполнить аналогичным образом:

```
zabbix_sender.exe [   ]
```

В сценариях отправки данных утилитой zabbix_sender в реальном времени несколько значений, переданных ей за короткий промежуток времени, будут накапливаться и затем отправляться серверу за одно соединение. Значения, пришедшие через менее, чем 0.2 секунды с момента получения предыдущего значения, накапливаются в одном стеке, но максимальное время их накопления до отправки — всё же 1 секунда.

Note:
Zabbix sender завершится с ошибкой, если в указанном файле конфигурации имеется ошибочный (не соответствующий определению *параметр=значение*) параметр.

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`).

7. Get

Обзор

Zabbix get — это утилита командной строки, которую можно использовать для взаимодействия с агентом Zabbix и получения необходимой информации от агента.

Эта утилита обычно используется для устранения неполадок агентов Zabbix.

См. также библиотеку Python для Zabbix, которая имеет встроенную функциональность для работы аналогично 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]
```

Другой пример запуска Zabbix get для получения 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 обозначения ключа элемента данных в оболочке используются кавычки. Кавычки не являются частью ключа элемента данных; они будут удалены оболочкой и не будут переданы агенту Zabbix.

Если ключ элемента данных не поддерживается, Zabbix get вернет код завершения 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 -                       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
```

См. также man-страницу Zabbix get для получения дополнительной информации.

Zabbix get в Windows можно запускать аналогичным образом:

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

8. JS

Обзор

zabbix_js — это утилита командной строки, которую можно использовать для тестирования встроенных скриптов.

Эта утилита выполняет пользовательский скрипт со строковым параметром и выводит результат. Скрипты выполняются с использованием встроенного механизма сценариев Zabbix.

В случае ошибок компиляции или выполнения zabbix_js выведет сообщение об ошибке в stderr и завершит работу с кодом 1.

Использование

```
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
```

zabbix_js принимает следующие параметры командной строки:

```
-s, --script script-file                                    .                        "-",                stdi
-i, --input input-file                                     .                             .
-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 входную информацию

Файл со скриптом (`script-file.js`):

```
return value;
```

Файл с входной информацией (`example.txt`):

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

Использование:

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

Вывод: `Example of input information from the file`

Пример 3: Выполнение скрипта с входными данными, прочитанными из stdin

Файл со скриптом (`script-file.js`):

```
return value;
```

Использование:

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

Ввод (stdin):

```
Example of input from stdin
```

Вывод: `Example of input from stdin`

9. Веб-сервис

Обзор

Веб-служба Zabbix — это процесс, используемый для взаимодействия с внешними веб-службами. В настоящее время веб-служба Zabbix используется для создания и отправки запланированных отчетов, при этом в будущем планируется добавить дополнительную функциональность.

Сервер Zabbix подключается к веб-службе по HTTP(S). Для работы веб-службы 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-службу Zabbix из исходного кода, укажите параметр configure `--enable-webservice`.

Чтобы настроить веб-службу Zabbix, обновите параметры файла конфигурации `zabbix_web_service.conf`.

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.

5 Конфигурация

Пожалуйста, используйте боковую панель для доступа к содержимому раздела «Конфигурация».

1 Настройка шаблона

Обзор

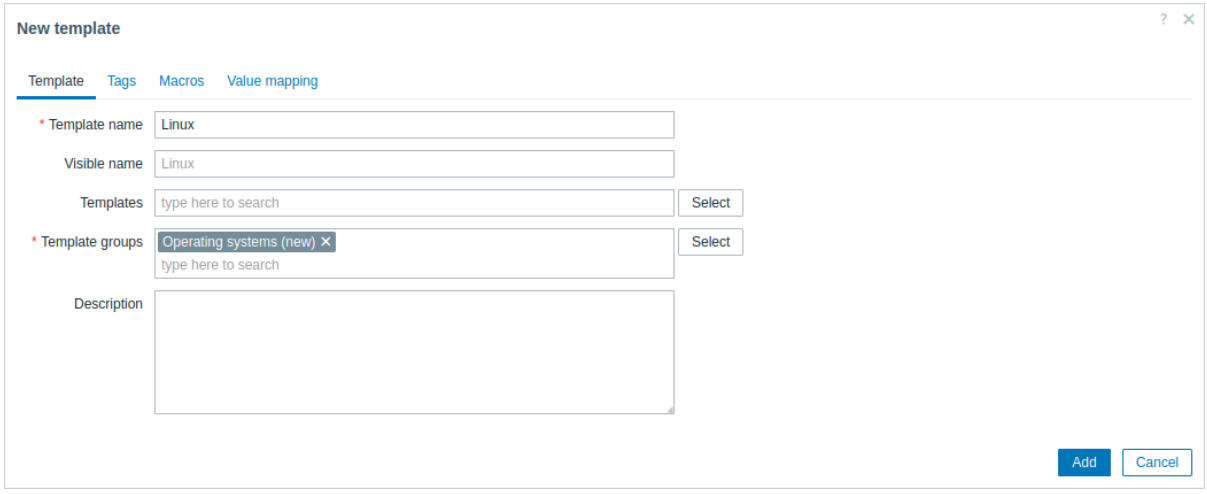
Настройка шаблона требует, чтобы вы сначала создали шаблон, определив его общие параметры, а затем добавили в него сущности (элементы данных, триггеры, графики и т. д.).

Создание шаблона

Чтобы создать шаблон, выполните следующие действия:

1. Перейдите в *Сбор данных* > *Шаблоны*.
2. Нажмите *Создать шаблон*.
3. Отредактируйте атрибуты шаблона.

Вкладка **Шаблон** содержит общие атрибуты шаблона.

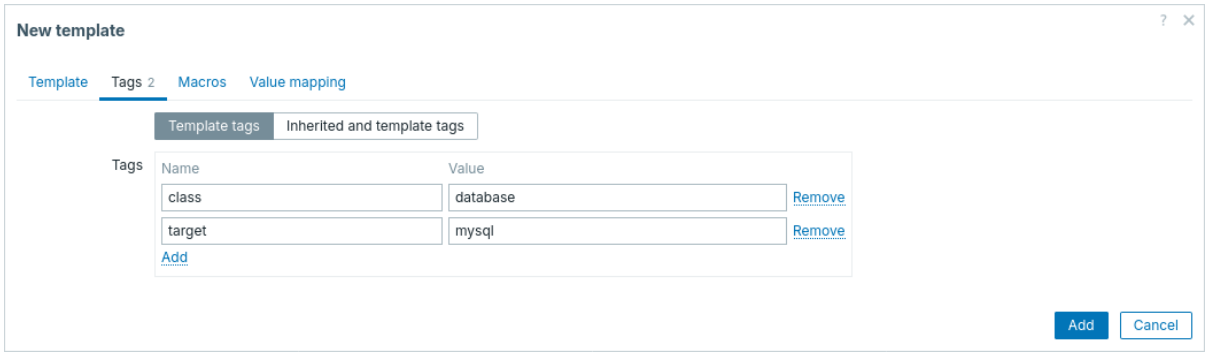


Все обязательные поля ввода отмечены красной звездочкой.

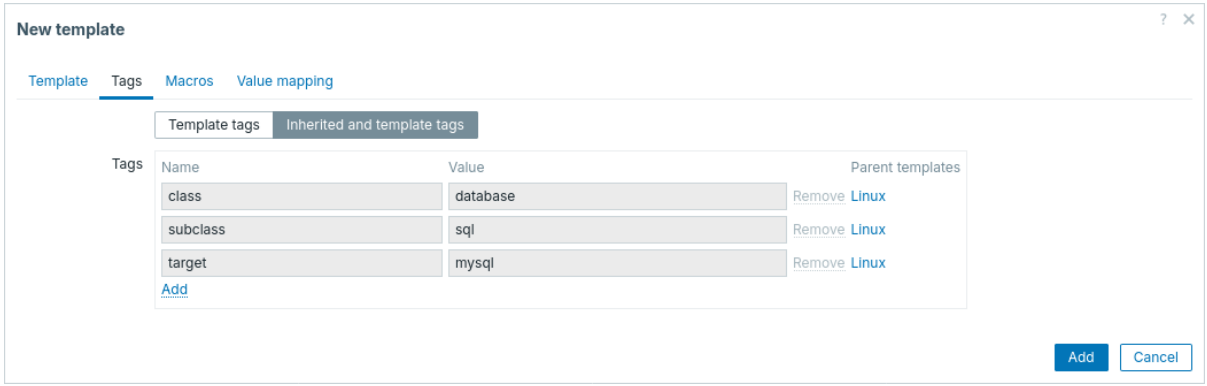
Атрибуты шаблона:

Параметр	Описание
Имя шаблона	Уникальное имя шаблона. Допускаются буквенно-цифровые символы, пробелы, точки, дефисы и символы подчеркивания. Начальные и конечные пробелы не допускаются.
Видимое имя	Если вы зададите это имя, оно будет отображаться в списках, картах и т. д.
Шаблоны	Свяжите с этим шаблоном один или несколько шаблонов. Все сущности (элементы данных, триггеры и т. д.) будут унаследованы из связанных шаблонов. Чтобы связать новый шаблон, введите его имя в поле *Шаблоны* — отобразится раскрывающийся список подходящих шаблонов. Либо нажмите *Выбрать*, чтобы открыть всплывающее окно *Шаблоны*, в котором изначально шаблоны не отображаются. Чтобы заполнить список, либо введите имя группы шаблонов в поле *Группа шаблонов* (и выберите одну из предложенных), либо нажмите кнопку *Выбрать* рядом с полем *Группа шаблонов*, чтобы открыть всплывающее окно *Группы шаблонов*. После выбора группы шаблонов всплывающее окно *Шаблоны* обновится и покажет шаблоны, принадлежащие этой группе. Выбранные шаблоны будут связаны при сохранении или обновлении конфигурации. Чтобы отвязать шаблон, используйте один из двух вариантов в блоке *Шаблоны*: *Отвязать* - отвязать шаблон, но сохранить его сущности (элементы данных, триггеры и т. д.); *Отвязать и очистить* - отвязать шаблон и удалить все его сущности (элементы данных, триггеры и т. д.).
Группы шаблонов	Группы шаблонов, к которым принадлежит шаблон.
Описание	Описание шаблона.
Поставщик и версия	Поставщик и версия шаблона; отображаются только при обновлении существующих шаблонов (готовых шаблонов, предоставляемых Zabbix, импортированных шаблонов или шаблонов, измененных через Template API), 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 новой основной версией Zabbix). Например, 7.0-0, 7.0-5, 8.0-0, 8.0-3.

Вкладка **Теги** позволяет определять теги на уровне шаблона. Все проблемы узлов сети, связанных с этим шаблоном, будут помечены указанными здесь значениями.



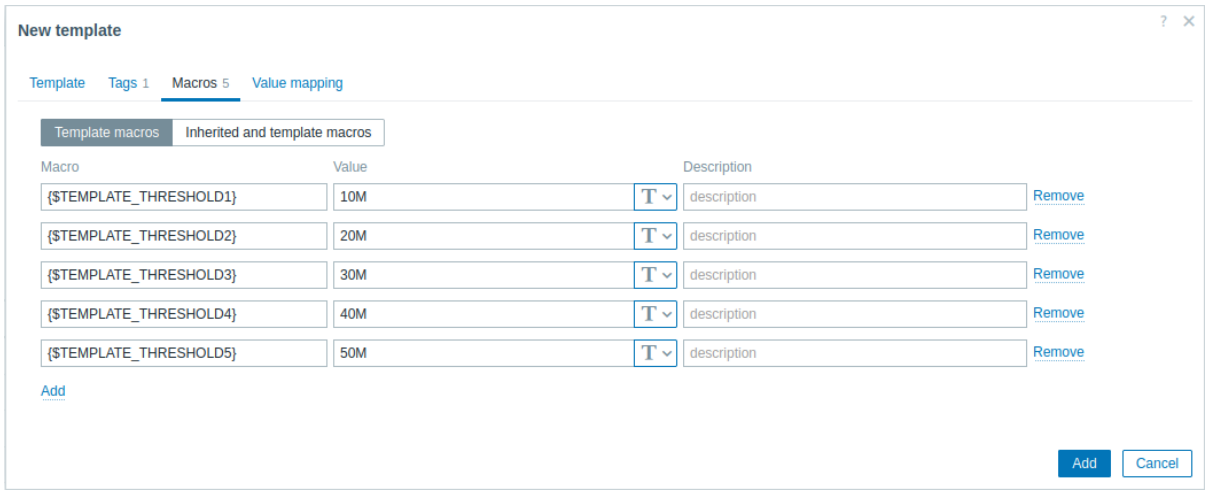
Если выбрать опцию *Унаследованные теги и теги шаблона*, можно также просматривать теги из связанных шаблонов, которые будут унаследованы этим шаблоном.



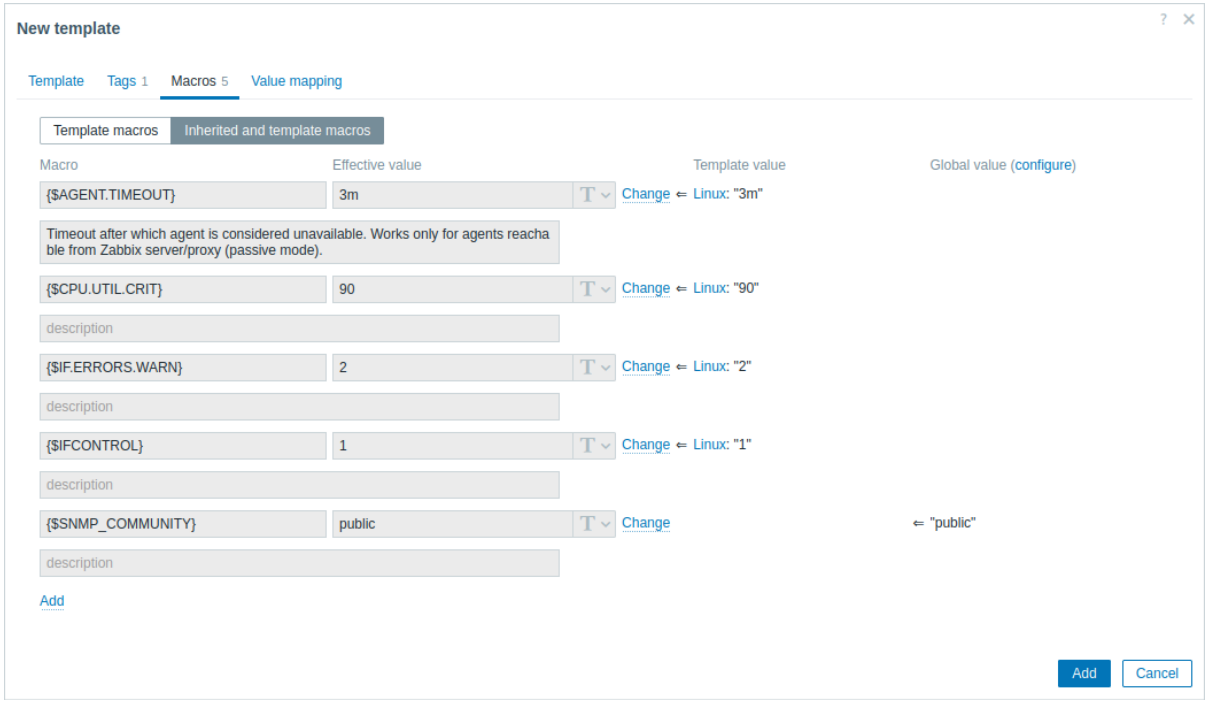
Для удобства предусмотрены ссылки на соответствующие шаблоны.

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 определять пользовательские макросы на уровне шаблона как пары имя-значение. Обратите внимание, что значения макросов могут храниться как обычный текст, секретный текст или секрет Vault. Также поддерживается добавление описания.



Если выбрать опцию *Унаследованные макросы и макросы шаблона*, можно также просматривать макросы из связанных шаблонов и глобальные макросы, которые будут унаследованы 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 редактировать макрос связанного шаблона или глобальный макрос на уровне шаблона, фактически создавая копию макроса в шаблоне.

Вкладка **Преобразование значений** позволяет настроить удобочитаемое представление данных элемента данных в преобразовании 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. Это **включает** сущности (элементы данных, триггеры и т. д.), как унаследованные из связанных шаблонов, так и напрямую прикрепленные к текущему шаблону, но **не включает** поставщика 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 со связанных узлов сети.
Cancel	Отменить редактирование свойств шаблона.

Добавление элементов данных, триггеров, графиков

Attention:
Элементы данных необходимо сначала добавить в шаблон. Триггеры и графики нельзя добавить без соответствующего элемента данных.

Существует два способа добавить элементы данных в шаблон:

1. Чтобы создать новые элементы данных, следуйте инструкциям из раздела Создание элемента данных.

2. Чтобы добавить существующие элементы данных в шаблон:

- Перейдите в *Сбор данных* > *Узлы сети* (или *Шаблоны*).
- Нажмите *Элементы данных* в строке нужного узла сети/шаблона.
- Отметьте флажками элементы данных, которые вы хотите добавить в шаблон.
- Нажмите *Копировать* под списком элементов данных.
- 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 (соответственно из списка триггеров и графиков), при этом следует помнить, что их можно добавить только после того, как будут добавлены необходимые элементы данных.

Добавление панелей

Чтобы добавить панели в шаблон в разделе *Сбор данных* > *Шаблоны*, выполните следующие действия:

1. Нажмите *Панели* в строке шаблона.
2. Настройте панель, следуя рекомендациям из раздела настройки панелей.

Attention:
При настройке виджетов на панели шаблона (вместо глобальной панели) параметры, связанные с узлом сети, недоступны, а некоторые параметры имеют другое название. Это связано с тем, что панели шаблона отображают данные только того узла сети, с которым связан шаблон. Например, параметры *Группы узлов сети*, *Исключить группы узлов сети* и *Узлы сети* в виджете *Проблемы* недоступны, параметр *Группы узлов сети* в виджете *Доступность узла сети* недоступен, а параметр *Показывать узлы сети на обслуживании* переименован в *Показывать данные на обслуживании* и т. д. Дополнительную информацию о доступности параметров в виджетах панели шаблона см. в описании параметров каждого виджета панели.

Note:
Подробнее о доступе к панелям узла сети, созданным из панелей шаблона, см. в разделе панели узла сети.

Настройка правил низкоуровневого обнаружения

См. раздел низкоуровневое обнаружение руководства.

Добавление веб-сценариев

Чтобы добавить веб-сценарии в шаблон в разделе *Сбор данных* > *Шаблоны*, выполните следующие действия:

1. Нажмите *Веб* в строке шаблона.
2. Настройте веб-сценарий обычным способом, как описано в разделе настройки веб-сценариев.

2 Настройка группы шаблонов

Overview

Template groups are used for the logical grouping of templates and assigning user permissions to them.

Each template must have at least one template group assigned. A template may belong to multiple template groups, and each template group may contain multiple templates.

Note that in Zabbix, all permissions are based on groups: user groups, host groups, and template groups. So, even if a single user needs access to a single template, it is granted by adding the user to a user group that has permission to access the template group containing that template.

Настройка

Attention:
Только пользователи с ролью Super admin могут создавать группы шаблонов.

В веб-интерфейсе Zabbix есть два способа создания группы шаблонов.

Первый способ:

- Перейдите в: *Сбор данных* → *Группы шаблонов*
- Нажмите *Создать группу шаблонов* в правом верхнем углу экрана
- Введите имя группы в форме



Второй способ: при настройке шаблона введите несуществующее имя группы в поле ввода *Группы шаблонов*.

После создания группы шаблонов вы можете нажать на имя шаблона в списке в разделе *Сбор данных* → *Группы шаблонов*, чтобы изменить имя группы, клонировать группу или удалить группу.

Удаление группы шаблонов удаляет только логическую группу, но не шаблоны в этой группе. Невозможно удалить группу шаблонов, если она является единственной группой для любого существующего шаблона.

Создание подгрупп шаблонов

Подгруппа шаблонов (или вложенная группа шаблонов) является дочерней по отношению к родительской группе шаблонов, в которой она содержится.

Подгруппа создается с использованием прямой косой черты '/' в поле ввода имени группы для обозначения ее связи с родительской группой(ами). Например:

- ввод `Linux servers/Databases` создает подгруппу `Linux servers/Databases` в родительской группе `Linux servers`.
- ввод `Linux servers/Databases/MySQL/Tokyo` создает соответствующую подгруппу во вложенных родительских группах `Linux servers`, `Linux servers/Databases`, `Linux servers/Databases/MySQL`.

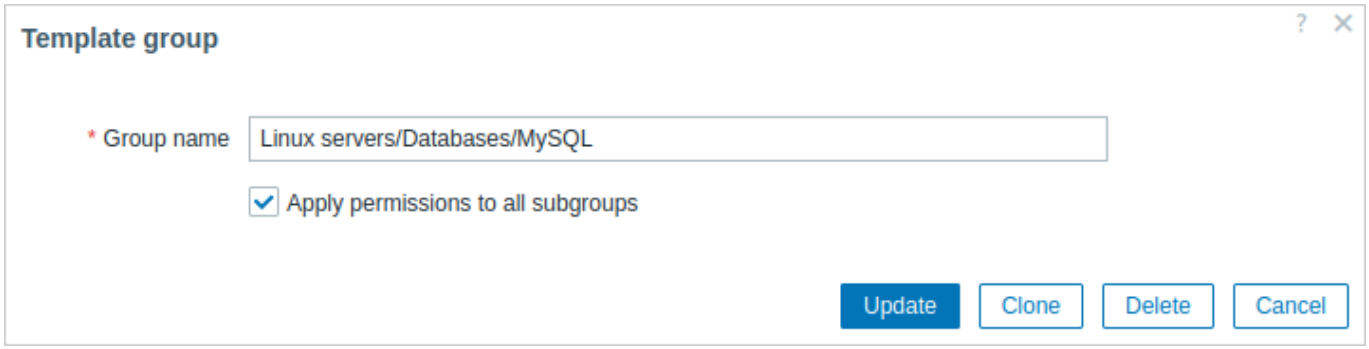
При создании подгруппы не допускается использование косой черты в начале или в конце, а также нескольких косых черт подряд. Экранирование '/' не поддерживается.

Перед созданием подгруппы не требуется создавать какие-либо родительские группы шаблонов. Вы можете выбрать, с чего начать: с создания подгруппы (например, `Linux servers/Databases`) или любой родительской группы шаблонов (в нашем примере — `Linux servers`). Если вы начнете с создания подгруппы, родительские группы шаблонов **не** будут созданы автоматически.

Права доступа к группам шаблонов

- При создании подгруппы в существующей родительской группе шаблонов (например, при создании `Linux servers/Databases`, когда `Linux servers` уже существует) права группы пользователей для подгруппы наследуются 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 устанавливаются.

При редактировании любой группы шаблонов также можно задать дополнительную опцию *Применить права доступа ко всем подгруппам*.



Если установить этот флажок и нажать *Обновить*, одинаковый уровень прав доступа будет применён ко всем текущим и будущим подгруппам редактируемой группы шаблонов.

Таким образом, если каким-либо группам пользователей были назначены разные права доступа к подгруппам редактируемой группы шаблонов, установка этого 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 перезапишет существующие права доступа. Любые изменения, внесённые с помощью этой опции, можно отменить только вручную.

3 Привязка/отвязка

Обзор

Связывание — это процесс, при котором шаблоны применяются к узлам сети, тогда как отвязывание удаляет связь с шаблоном у узла сети.

Привязка шаблона

Чтобы привязать шаблон к узлу сети, выполните следующие действия:

1. Перейдите в *Data collection* → *Hosts*.
2. Нажмите на нужный узел сети.
3. Начните вводить имя шаблона в поле *Templates*. Появится список подходящих шаблонов; прокрутите вниз, чтобы выбрать нужный. Либо нажмите *Select* рядом с полем и выберите один или несколько шаблонов из списка во всплывающем окне.
4. Нажмите *Add/Update* в форме атрибутов узла сети.

Теперь узел сети будет содержать все сущности шаблона. Сюда входят элементы данных, триггеры, графики, правила обнаружения низкого уровня, веб-сценарии, а также панели.

Attention:
Привязка нескольких шаблонов к одному и тому же узлу сети завершится ошибкой, если эти шаблоны содержат элементы данных с одинаковым ключом элемента данных. Кроме того, поскольку триггеры и графики используют элементы данных, их также нельзя привязать к одному узлу сети из нескольких шаблонов, если используются одинаковые ключи элементов данных.

Когда сущности (элементы данных, триггеры и т. д.) добавляются из шаблона:

- ранее существовавшие идентичные сущности на узле сети обновляются как сущности шаблона, и **все существующие пользовательские изменения сущности на уровне узла сети будут потеряны**;
- сущности из шаблона добавляются;
- любые напрямую привязанные сущности, которые до привязки шаблона существовали только на узле сети, остаются без изменений.

В списках все сущности из шаблона теперь имеют префикс в виде имени шаблона, что указывает на их принадлежность к конкретному шаблону. Само имя шаблона (серым текстом) является ссылкой, позволяющей открыть список этих сущностей на уровне шаблона.

Note:
Для некоторых элементов данных, таких как внешние проверки, проверки HTTP agent, простые проверки, проверки SSH и проверки Telnet, интерфейс узла сети необязателен. Если на момент привязки шаблона у узла сети не определён интерфейс, эти элементы данных будут добавлены без интерфейса. Если вы добавите интерфейс узла сети позже, он не будет автоматически назначен уже существующим элементам данных. Чтобы назначить новый интерфейс узла сети всем элементам данных шаблона сразу, отвяжите шаблон от узла сети, а затем привяжите его снова. Чтобы сохранить историю элементов данных, используйте опцию *Unlink*, а не *Unlink and clear*.

Если какая-либо сущность не имеет префикса в виде имени шаблона, это означает, что она существовала на узле сети ранее и не была добавлена шаблоном.

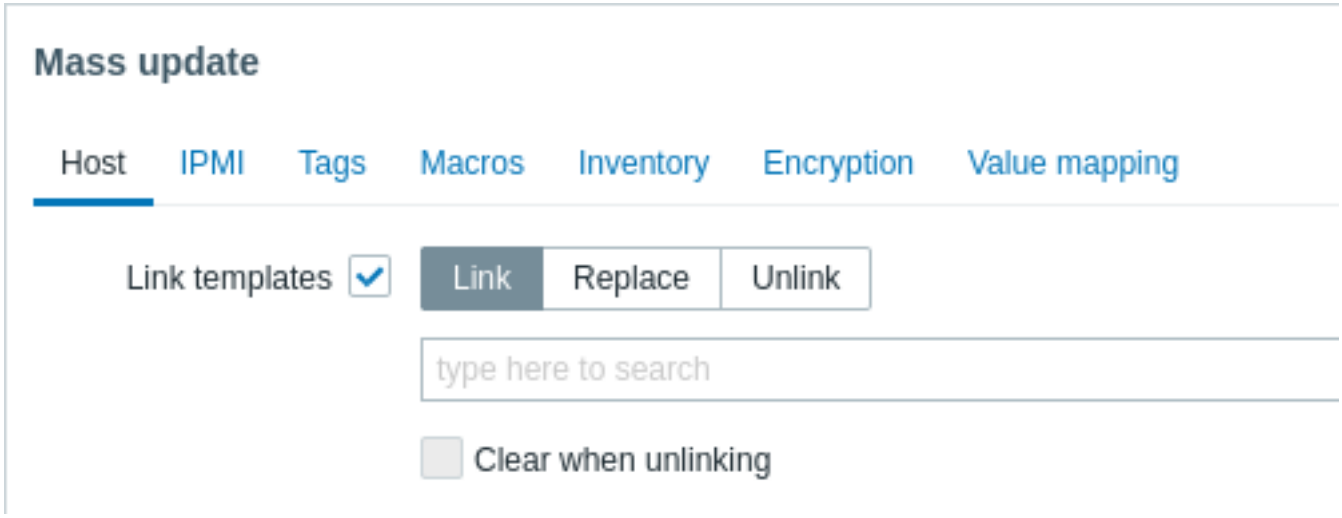
Критерии уникальности сущностей

При добавлении сущностей (элементов данных, триггеров и т. д.) из шаблона важно понимать, какие из этих сущностей уже существуют на узле сети и требуют обновления, а какие сущности отличаются. Критериями уникальности для определения совпадения/различия являются:

- для элементов данных — ключ элемента данных;
- для триггеров — имя триггера и выражение;
- для пользовательских графиков — имя графика и его элементы данных.

Привязка шаблонов к нескольким узлам сети

Чтобы обновить привязку шаблонов для многих узлов сети, в разделе *Сбор данных* → *Узлы сети* выберите несколько узлов сети, отметив их флажками, затем нажмите **Массовое обновление** под списком и выберите *Привязать шаблоны*:



Чтобы привязать дополнительные шаблоны, начните вводить имя шаблона в поле автодополнения, пока не появится выпадающий список с подходящими шаблонами. Просто прокрутите список вниз, чтобы выбрать шаблон для привязки.

Параметр *Заменить* позволяет привязать новый шаблон, одновременно отвязав любой шаблон, который ранее был привязан к узлам сети. Параметр *Отвязать* позволяет указать, какие шаблоны нужно отвязать. Параметр *Очистить при отвязке* позволяет не только отвязать все ранее привязанные шаблоны, но и удалить все сущности, унаследованные от них (элементы данных, триггеры и т. д.).

Note:
Zabbix предлагает большой набор предопределённых шаблонов. Их можно использовать в качестве примера, однако будьте осторожны с использованием их без изменений в рабочей среде, так как они могут содержать слишком много элементов данных и слишком часто опрашивать данные. Если вы решите их использовать, настройте их в соответствии с вашими реальными потребностями.

Редактирование связанных сущностей

Если вы попытаетесь отредактировать элемент данных или триггер, который был связан из шаблона, вы можете заметить, что многие ключевые параметры недоступны для редактирования. Это логично, поскольку идея шаблонов заключается в том, что объекты редактируются централизованно на уровне шаблона. Однако вы по-прежнему можете, например, включать/отключать элемент данных на отдельных узлах сети, а также задавать интервал обновления, длину истории и некоторые другие параметры.

Если вы хотите полностью отредактировать сущность, необходимо редактировать её на уровне шаблона (ссылка быстрого перехода на уровень шаблона отображается в названии формы), при этом следует помнить, что эти изменения затронут все узлы сети, к которым привязан этот шаблон.

> **Attention:**
> Любые пользовательские изменения сущностей, выполненные на уровне шаблона, переопределят предыдущие пользовательские изменения этих сущностей на уровне узла сети.

Отсоединение шаблона

Чтобы отсоединить шаблон от узла сети, выполните следующие действия:

1. Перейдите в *Сбор данных* → *Узлы сети*.
2. Нажмите на нужный узел сети и найдите поле *Шаблоны*.
3. Нажмите *Отсоединить* или *Отсоединить и очистить* рядом с шаблоном, чтобы отсоединить шаблон.
4. Нажмите *Обновить* в форме атрибутов узла сети.

При выборе варианта *Отсоединить* будет просто удалена связь с шаблоном, при этом все его сущности останутся у узла сети. Сюда входят элементы данных, триггеры, графики, правила обнаружения низкого уровня и веб-сценарии, но не панели. Обратите внимание, что карты значений и теги, унаследованные от связанных шаблонов, также будут удалены.

При выборе варианта *Отсоединить и очистить* будут удалены как связь с шаблоном, так и все его сущности (элементы данных, триггеры и т. д.).

4 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 Т1 и Т2 к шаблону Т3, мы дополняем Т3 объектами из Т1 и Т2, а не наоборот. Если мы присоединим шаблон А к шаблонам В и С, мы дополняем шаблоны В и С объектами из А.

Настройка вложенных шаблонов

Чтобы связать шаблоны, необходимо взять существующий шаблон (или создать новый), а затем:

1. Откройте форму настройки шаблона.
2. Найдите поле *Шаблоны*.
3. Нажмите *Выбрать*, чтобы открыть всплывающее окно *Шаблоны*.
4. Во всплывающем окне выберите необходимые шаблоны и нажмите *Выбрать*, чтобы добавить шаблоны в список.
5. Нажмите *Добавить* или *Обновить* в форме конфигурации шаблона.

Таким образом, все сущности настроенного шаблона, а также все сущности связанных шаблонов теперь появятся в настройках шаблона. Сюда входят элементы, триггеры, графики, правила низкоуровневого обнаружения и веб-сценарии, но исключаются панели мониторинга. Однако связанные панели мониторинга шаблонов тем не менее будут унаследованы хостами.

Чтобы отменить привязку любого из связанных шаблонов, нажмите *Отсоединить* или *Отсоединить и очистить* в форме конфигурации шаблона, а затем нажмите *Обновить*.

Опция *Отсоеденить* просто удалит связь со связанным шаблоном, не удаляя при этом ни один из его объектов (элементов, триггеров и т. д.).

Опция *Отсоединить и очистить* удалит как связь со связанным шаблоном, так и все его объекты (элементы, триггеры и т. д.).

5 Массовое обновление

Обзор

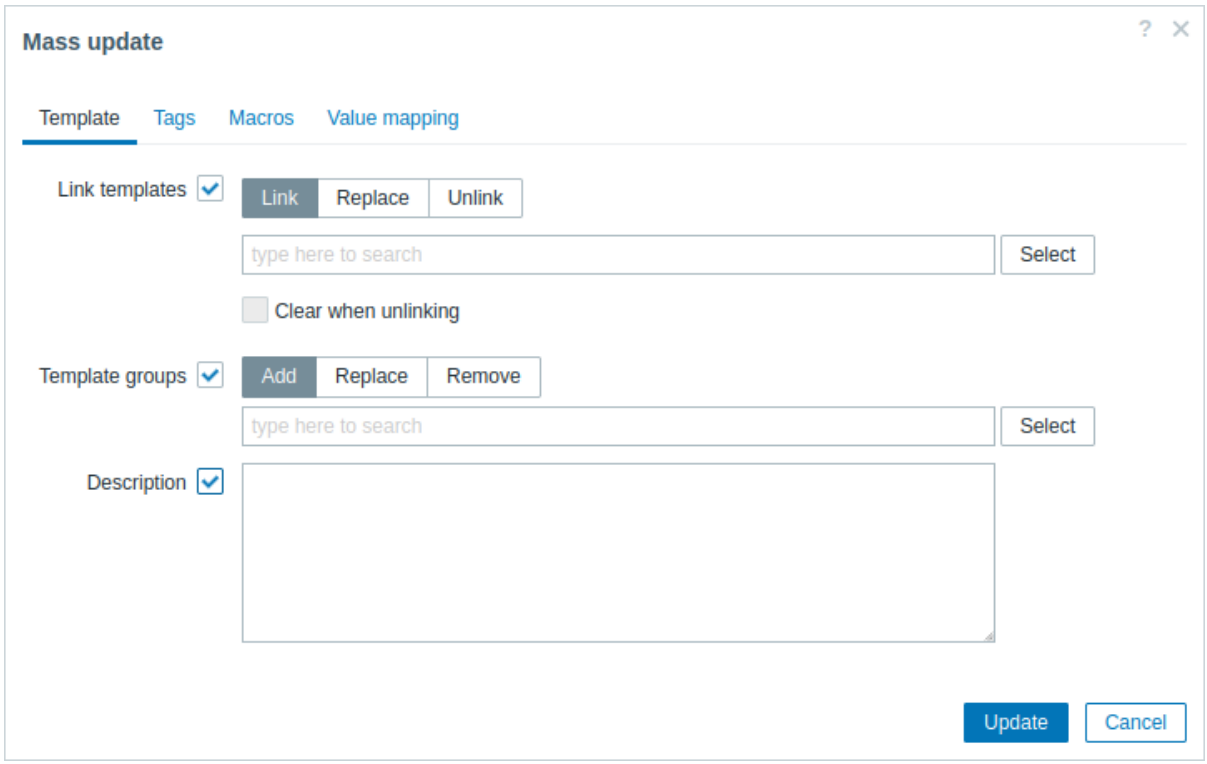
Иногда вы можете захотеть изменить какой-либо атрибут для нескольких шаблонов одновременно. Вместо того, чтобы открывать каждый отдельный шаблон для редактирования, вы можете воспользоваться функцией массового обновления.

Использование массового обновления

Чтобы массово обновить несколько шаблонов, выполните следующие действия:

1. Установите флажки напротив шаблонов, 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. Установите флажки для любых атрибутов, которые нужно обновить, 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, какие дополнительные шаблоны нужно связать;
- *Заменить* — укажите, какие шаблоны нужно связать, одновременно отвязав все ранее связанные шаблоны;
- *Отвязать* — укажите, какие шаблоны нужно отвязать.

Чтобы указать шаблоны для связывания/отвязки, начните вводить имя шаблона в поле автодополнения, пока не появится выпадающий список с подходящими шаблонами. Просто прокрутите список вниз и выберите нужные шаблоны.

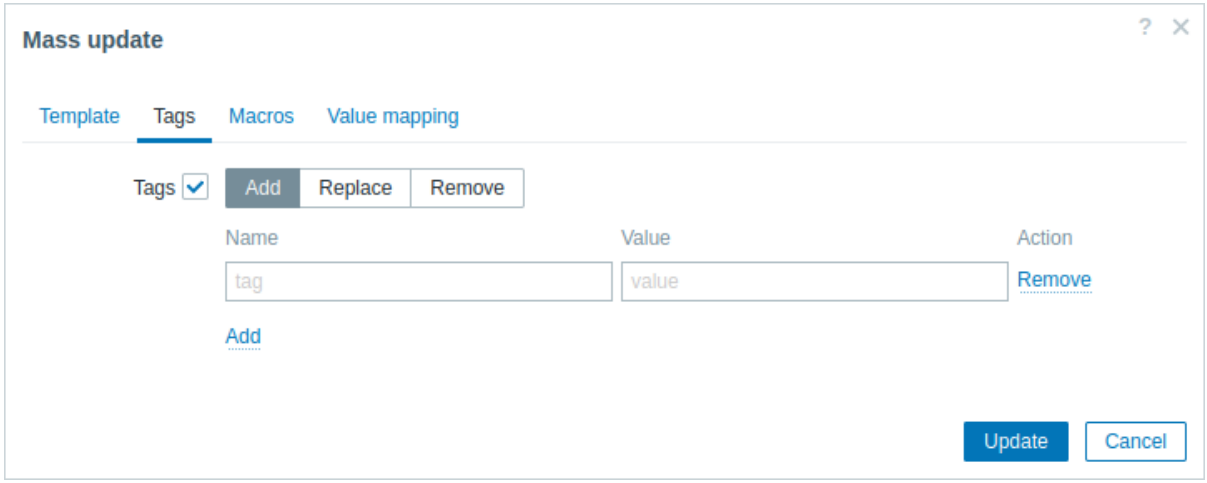
Параметр *Очистить при отвязке* позволяет отвязать все ранее связанные шаблоны, а также удалить все унаследованные от них элементы (элементы данных, триггеры, графики и т. д.).

При выборе соответствующей кнопки для обновления *Группы шаблонов* доступны следующие параметры:

- *Добавить* — позволяет указать дополнительные группы шаблонов из существующих или ввести для шаблонов совершенно новые группы шаблонов;
- *Заменить* — удалит шаблон из всех существующих групп шаблонов и заменит их на указанные в этом поле группы (существующие или новые);
- *Удалить* — удалит определённые группы шаблонов из шаблонов.

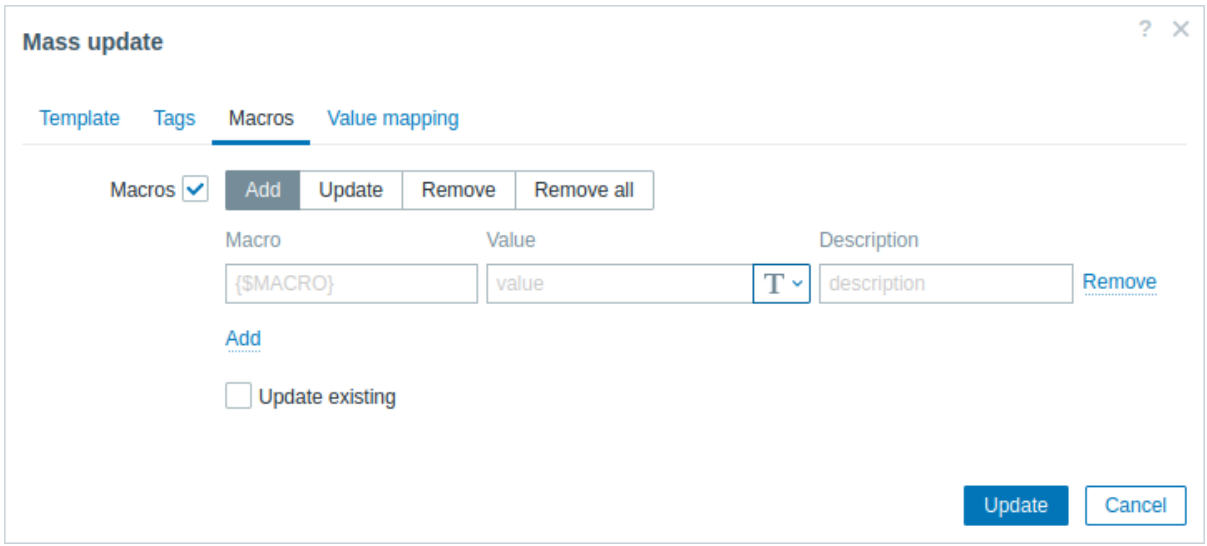
Эти поля поддерживают автодополнение — при вводе в них появляется выпадающий список подходящих групп шаблонов. Если группа шаблонов новая, она также появляется в выпадающем списке и помечается как *(new)* после строки. Просто прокрутите список вниз и выберите нужное.

Вкладка **Теги** позволяет массово обновлять теги на уровне шаблона.



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 в один и тот же шаблон.

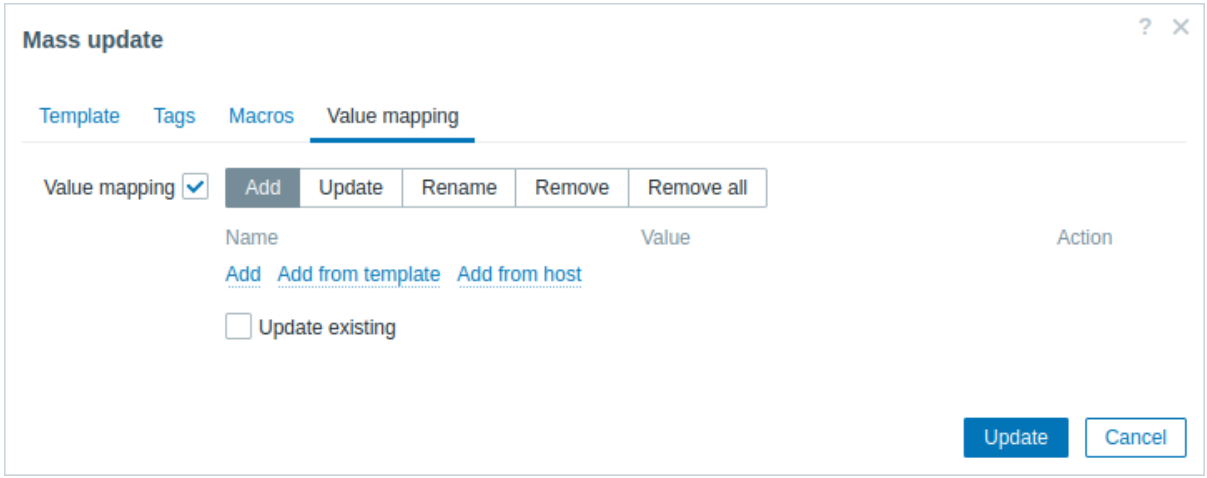
Вкладка **Макросы** позволяет массово обновлять макросы на уровне шаблона.



При выборе соответствующей кнопки для обновления макросов доступны следующие параметры:

- *Добавить* — позволяет указать дополнительные пользовательские макросы для шаблонов. Если установлен флажок *Обновить существующие*, будут обновлены значение, тип и описание для указанного имени макроса. Если флажок не установлен и макрос с таким именем уже существует в шаблоне(ах), он не будет обновлен.
- *Обновить* — заменит значения, типы и описания макросов, указанных в этом списке. Если установлен флажок *Добавить отсутствующие*, макрос, который ранее не существовал в шаблоне, будет добавлен как новый макрос. Если флажок не установлен, будут обновлены только те макросы, которые уже существуют в шаблоне.
- *Удалить* — удалит 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* — удалит все пользовательские макросы из шаблонов. Если флажок *Я подтверждаю удаление всех макросов* не установлен, откроется новое всплывающее окно с запросом на подтверждение удаления всех макросов.

Вкладка **Преобразование значений** позволяет массово обновлять преобразования значений.



Для обновления преобразований значений доступны кнопки со следующими параметрами:

- *Добавить* — добавить преобразования значений в шаблоны. Если отметить *Обновить существующие*, будут обновлены все свойства преобразования значений с таким именем. В противном случае, если преобразование значений с таким именем уже существует, оно не будет обновлено.
- *Обновить* — обновить существующие преобразования значений. Если отметить *Добавить отсутствующие*, преобразование значений, которого ранее не было в шаблоне, будет добавлено как новое. В противном случае будут обновлены только те преобразования значений, которые уже существуют в шаблоне.
- *Переименовать* — задать новое имя существующему преобразованию значений.
- *Удалить* — удалить указанные преобразования значений из шаблонов. Если отметить *Кроме выбранных*, будут удалены все преобразования значений, **кроме** указанных.
- *Удалить все* — удалить все преобразования значений из шаблонов. Если флажок *Я подтверждаю удаление всех преобразований значений* не установлен, откроется новое всплывающее окно с запросом подтверждения удаления.

Для операций добавления/обновления преобразований значений доступны параметры *Добавить из шаблона* и *Добавить из узла сети*.
Они позволяют выбрать преобразования значений соответственно из шаблона или узла сети.

Когда все необходимые изменения будут внесены, нажмите *Обновить*. Атрибуты будут соответствующим образом обновлены для всех выбранных шаблонов.

1. 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, с именем «Server X», а затем можно добавить к нему элементы данных мониторинга.

Узлы сети организованы в группы узлов сети.

Перейдите к:

- Мастер узлов сети для пошагового интерфейса создания и настройки нового или существующего узла сети.
- Настройка узла сети для классического подхода к созданию и настройке узла сети.

1 Мастер узла сети

Обзор

Мастер узлов сети — это пошаговый интерфейс с подсказками для настройки цели мониторинга (устройства, приложения, сервиса и т. д.) в Zabbix. Он проводит вас через следующие шаги:

- Выбор шаблона
- Создание или выбор узла сети
- Установка Zabbix агент или агент 2
- Добавление интерфейса узла сети

- Применение дополнительной конфигурации к вашей цели мониторинга или узлу сети Zabbix (если это требуется шаблоном)

Чтобы открыть мастер узлов сети в веб-интерфейсе Zabbix, выполните следующие действия:

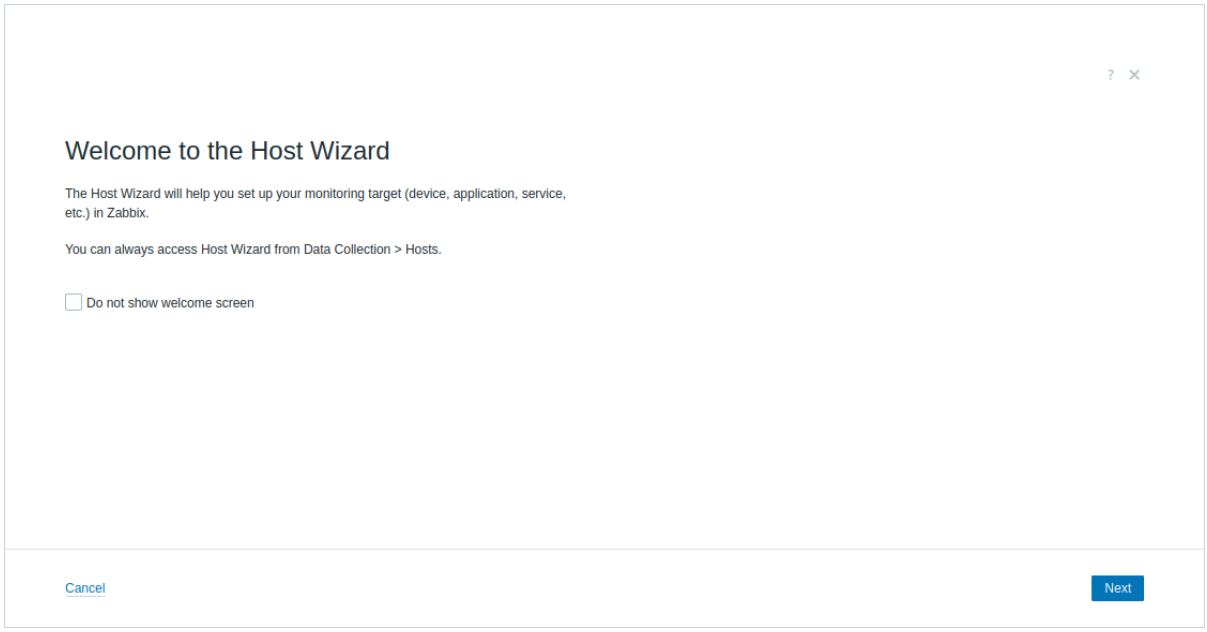
- Перейдите в: *Сбор данных* > *Узлы сети*
- Нажмите *Мастер узлов сети* в правом верхнем углу экрана
- Следуйте инструкциям мастера

Вы также можете редактировать существующий узел сети, щёлкнув 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 содержит конфигурацию, требуемую выбранным шаблоном.

В качестве альтернативы вы можете использовать классический подход к созданию и настройке узла сети.

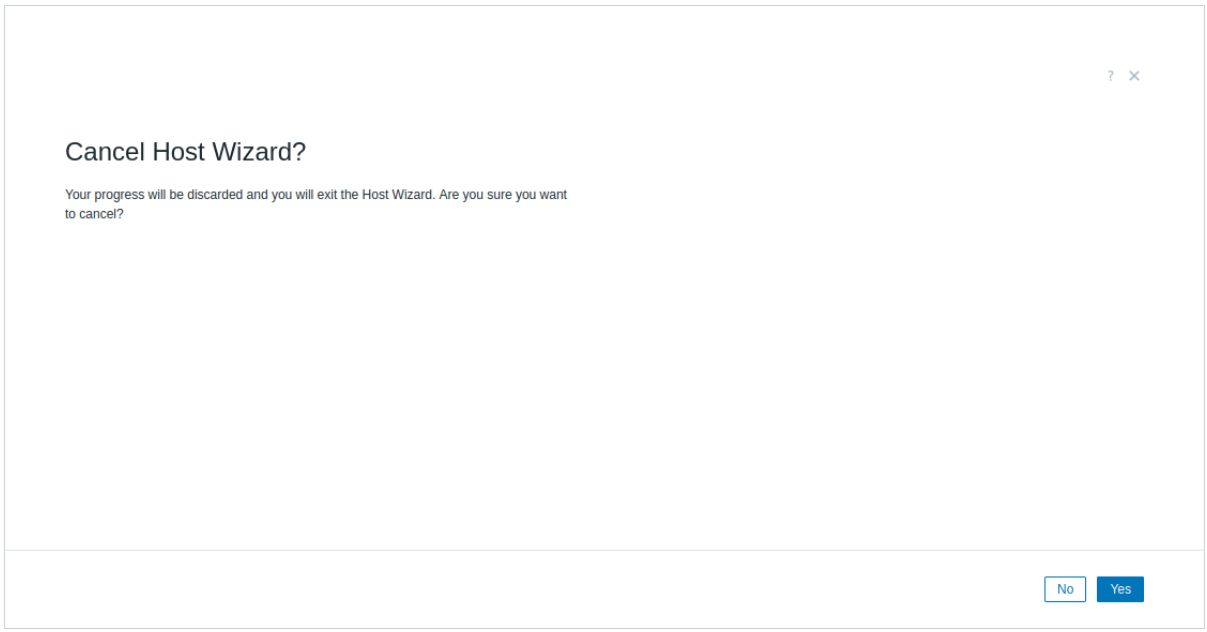
Добро пожаловать в мастер настройки узла сети

При запуске мастера настройки узла сети отображается экран приветствия:



Чтобы пропустить этот экран в будущих сеансах, вы можете установить флажок *Не показывать экран приветствия* и нажать *Далее*.

Если на следующих шагах мастера настройки узла сети вы начнёте настраивать параметры и попытаетесь закрыть мастер, щёлкнув по значку закрытия или нажав ESC, появится экран подтверждения:



Нажатие *Да* приведёт к выходу из мастера настройки узла сети без сохранения прогресса. Нажатие *Нет*, ESC или значка закрытия вернёт вас к последнему шагу.

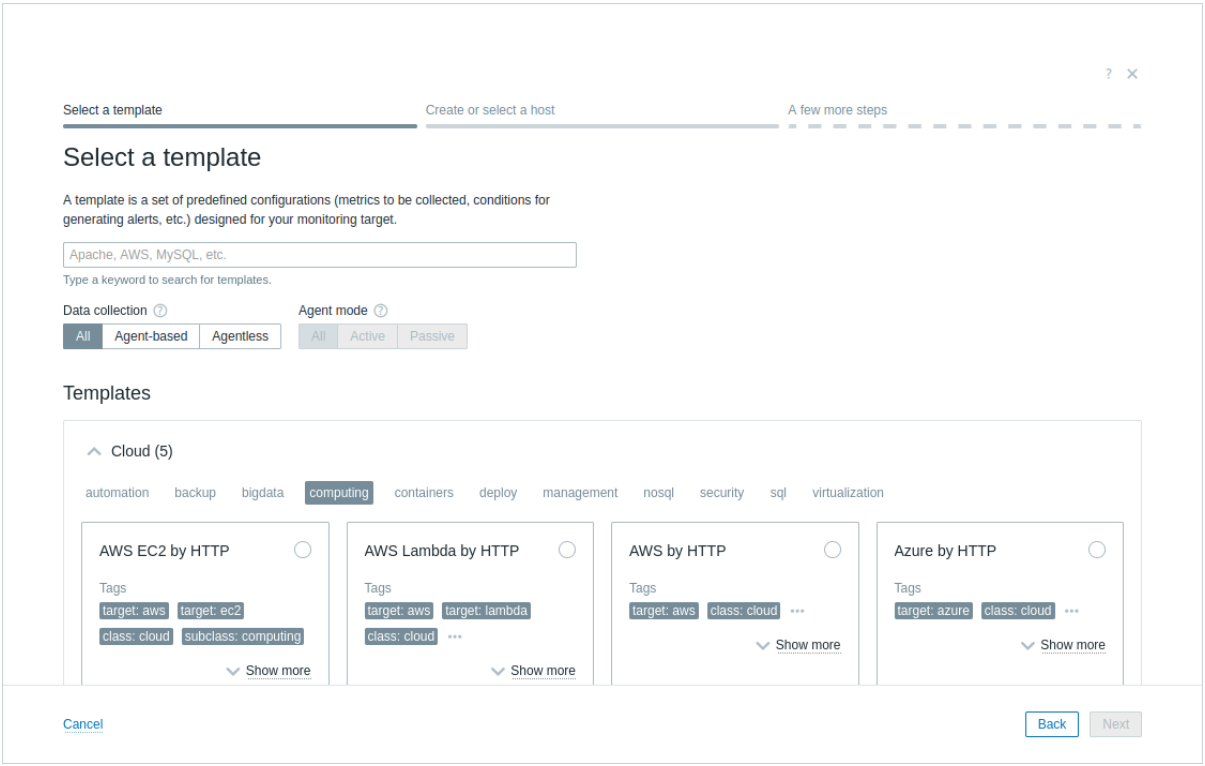
Выберите шаблон

Первым шагом при настройке цели мониторинга является выбор шаблона — набора предопределенных конфигураций (метрик для сбора, условий для генерации оповещений и т. д.), предназначенных для вашей цели мониторинга.

Готовые шаблоны Zabbix предлагают различные предопределенные конфигурации мониторинга для операционных систем, приложений, баз данных, сетевых устройств, сервисов и многого другого.

На этом шаге вы можете:

- Просматривать шаблоны по классу (Cloud, Database, Network и т. д.) и подклассу (например, automation, backup), которые основаны на тегах шаблона.
- Искать шаблоны по ключевым словам в имени шаблона или в именах и значениях тегов шаблона.
- Фильтровать шаблоны по методу сбора данных (на основе агента или без агента; шаблоны на основе агента включают как минимум один элемент данных Zabbix agent).
- Фильтровать шаблоны по режиму агента (активный или пассивный; активные шаблоны включают как минимум один элемент данных Zabbix agent (active), тогда как пассивные шаблоны включают как минимум один пассивный элемент данных; подробнее см. Пассивные и активные проверки агента).
- При использовании Host Wizard для настройки существующего узла сети вы также можете фильтровать шаблоны, уже привязанные к узлу сети.



Attention:
При выборе шаблона вы можете увидеть следующее сообщение: *Some templates (n) are incompatible with the Host Wizard. See how to update them. Custom templates are not supported.* Это означает, что некоторые готовые шаблоны Zabbix пока несовместимы с Host Wizard и требуют обновления; см. инструкции по обновлению шаблонов.

Выбранный вами шаблон определит следующие шаги в Host Wizard. Например, если вы выберете шаблон *MySQL by Zabbix agent*, который использует Zabbix agent для сбора данных, мастер проведет вас через процесс установки агента.

Создать или выбрать узел сети

Шаблон должен быть связан с узлом сети — сущностью в Zabbix, которая представляет цель мониторинга. После связывания узел сети получает все сущности шаблона, такие как элементы данных (метрики, которые необходимо собирать) и триггеры (условия для генерации оповещений), необходимые для мониторинга.

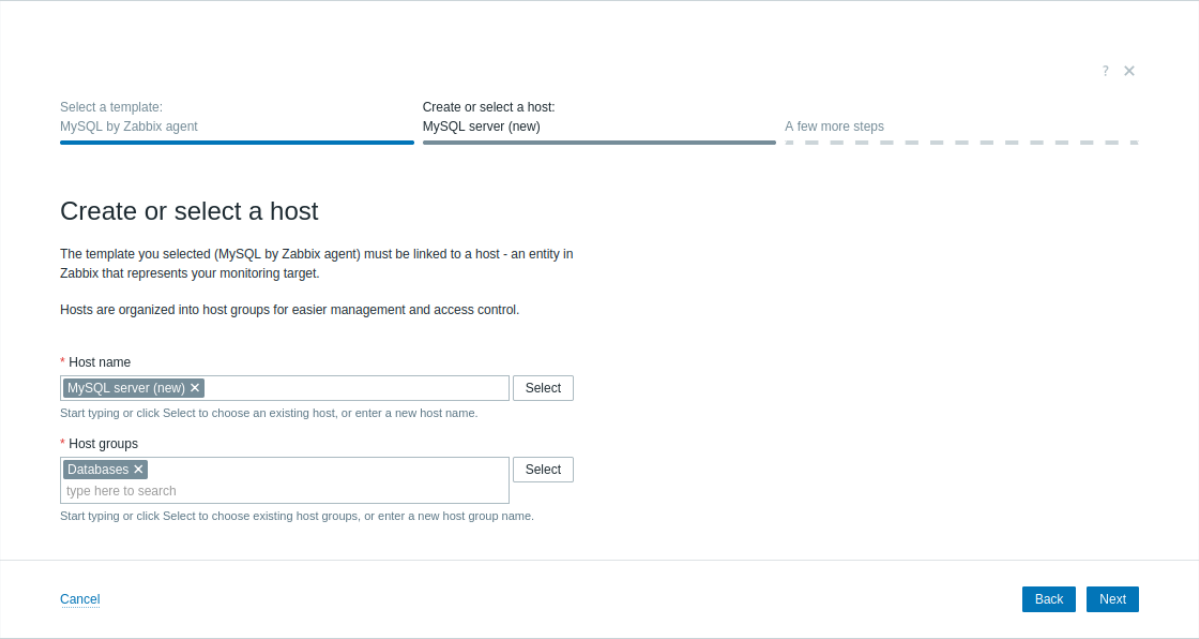
Каждый узел сети также должен принадлежать как минимум одной группе узлов сети, которая используется для организации узлов сети и назначения им пользовательских прав.

На этом шаге вы можете:

- Создать новый узел сети и группу узлов сети, указав имена для обоих.
- Создать новый узел сети, указав его имя, и выбрать существующую группу узлов сети, в которую будет добавлен новый узел сети.
- Выбрать существующий узел сети; вы также можете назначить этот узел сети дополнительным группам 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 следующих шагах. Обнаруженные узлы сети нельзя выбрать.

Например, вы можете создать новый узел сети с именем *MySQL server*, чтобы представить локально установленный сервер MySQL, который вы хотите отслеживать. Ранее выбранный шаблон *MySQL by Zabbix agent* будет связан с этим узлом сети. Кроме того, вы можете выбрать существующую группу узлов сети *Databases* (или создать новую, например, *MySQL servers*), чтобы организовать этот узел сети вместе с другими узлами сети, связанными с базами данных, и в дальнейшем упростить управление правами доступа.



Установка Zabbix агент

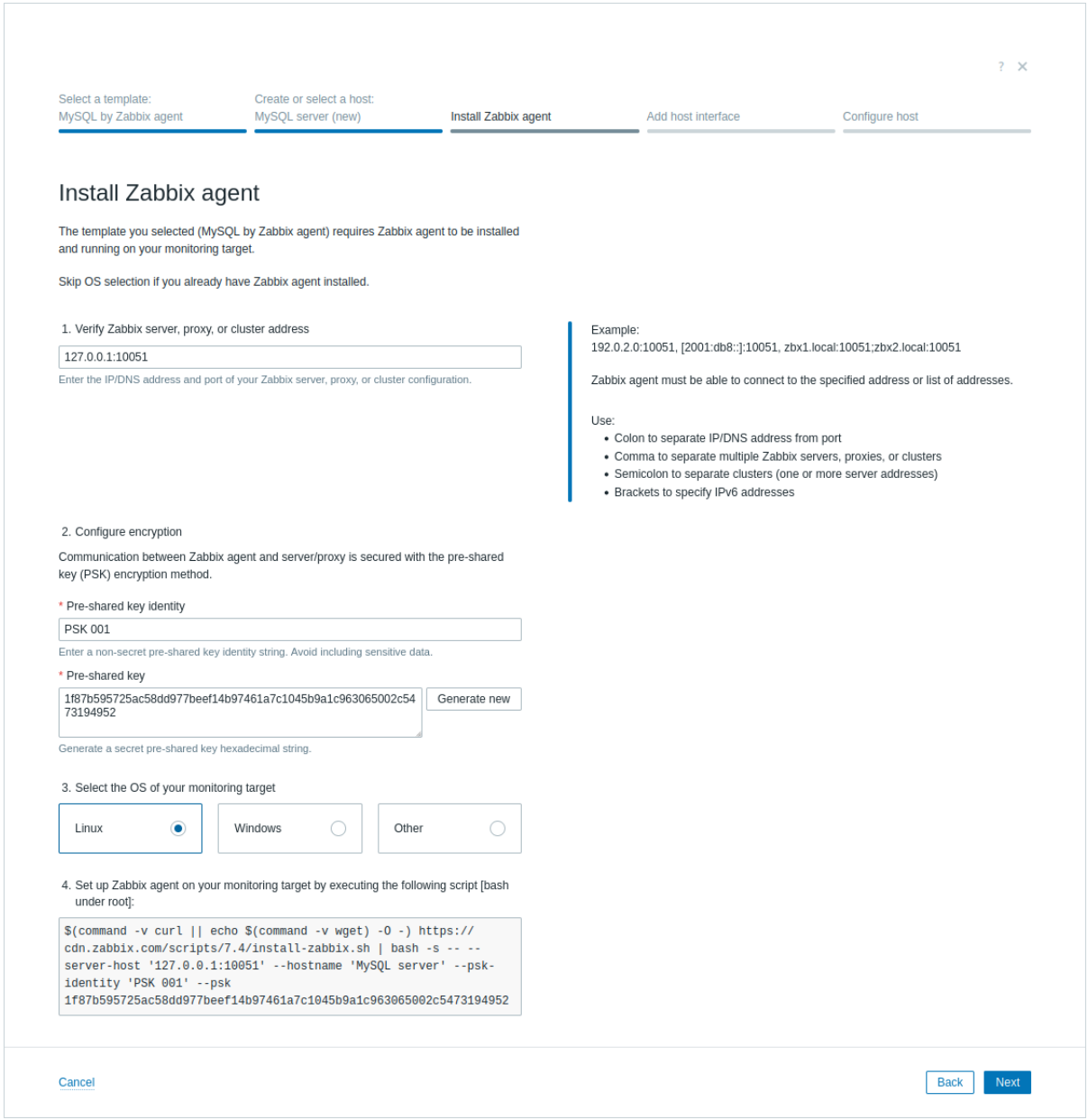
Шаблоны на основе агента (например, *MySQL by Zabbix agent*) требуют установки Zabbix агент или agent 2 — процесса, развертываемого на целевом объекте мониторинга для активного мониторинга локальных ресурсов и приложений (если это требуется шаблоном).

На этом шаге:

- Проверьте адрес Zabbix сервер, прокси или кластера (например, 192.0.2.0:10051). Это значение будет использоваться для настройки параметров Server и ServerActive для Zabbix агент.
- Введите новый, уникальный и не являющийся секретом идентификатор предварительно разделяемого ключа (например, *PSK 001* или *mysql-agent-psk1*) и сгенерируйте предварительно разделяемый ключ. Эти значения будут использоваться для настройки шифрования с pre-shared key (PSK) для узла сети Zabbix (например, *MySQL server*) и Zabbix агент.
- Выберите операционную систему машины, на которой размещен ваш целевой объект мониторинга.
- Установите Zabbix агент в этой системе, запустив на ней предоставленный скрипт или следуя инструкциям по установке.

Attention:
При редактировании существующего узла сети конфигурация PSK перезапишет все существующие настройки шифрования на узле сети.

Например, вы можете задать *Pre-shared key identity* как *PSK 001* и сгенерировать новый *Pre-shared key*. Затем вы можете выбрать *Linux* в качестве операционной системы и запустить предоставленный скрипт в этой системе. После установки Zabbix агент вернитесь в мастер настройки узла сети.

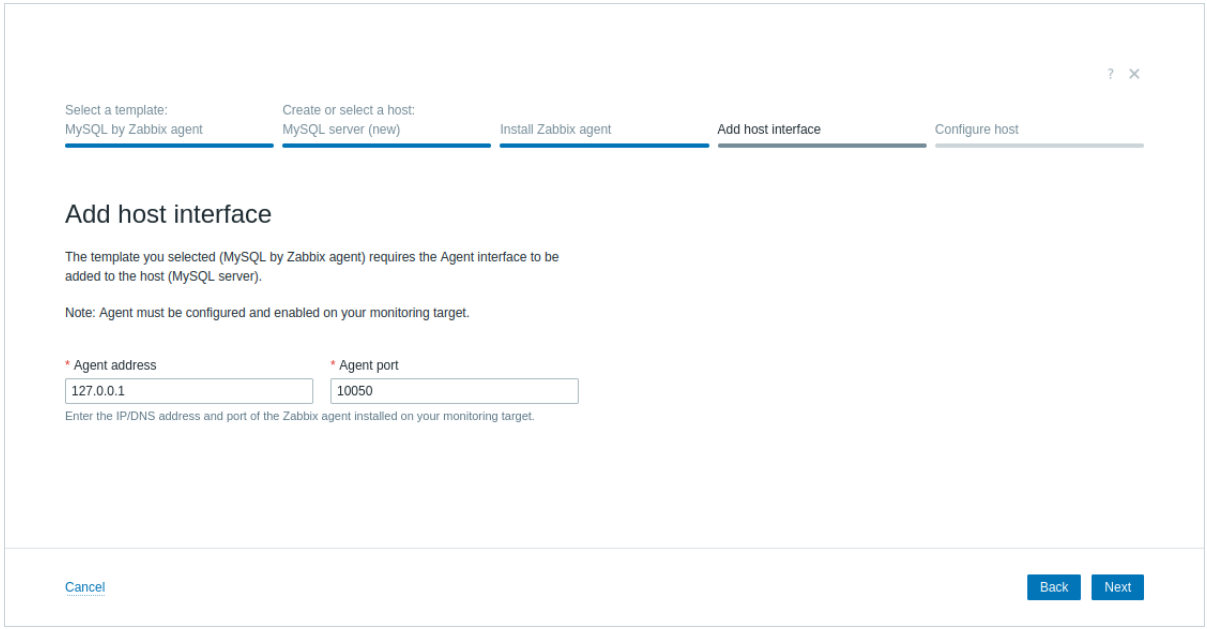


Добавление интерфейса узла сети

Интерфейс узла сети определяет, как сервер Zabbix подключается к цели 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, необходимые для выбранного шаблона.

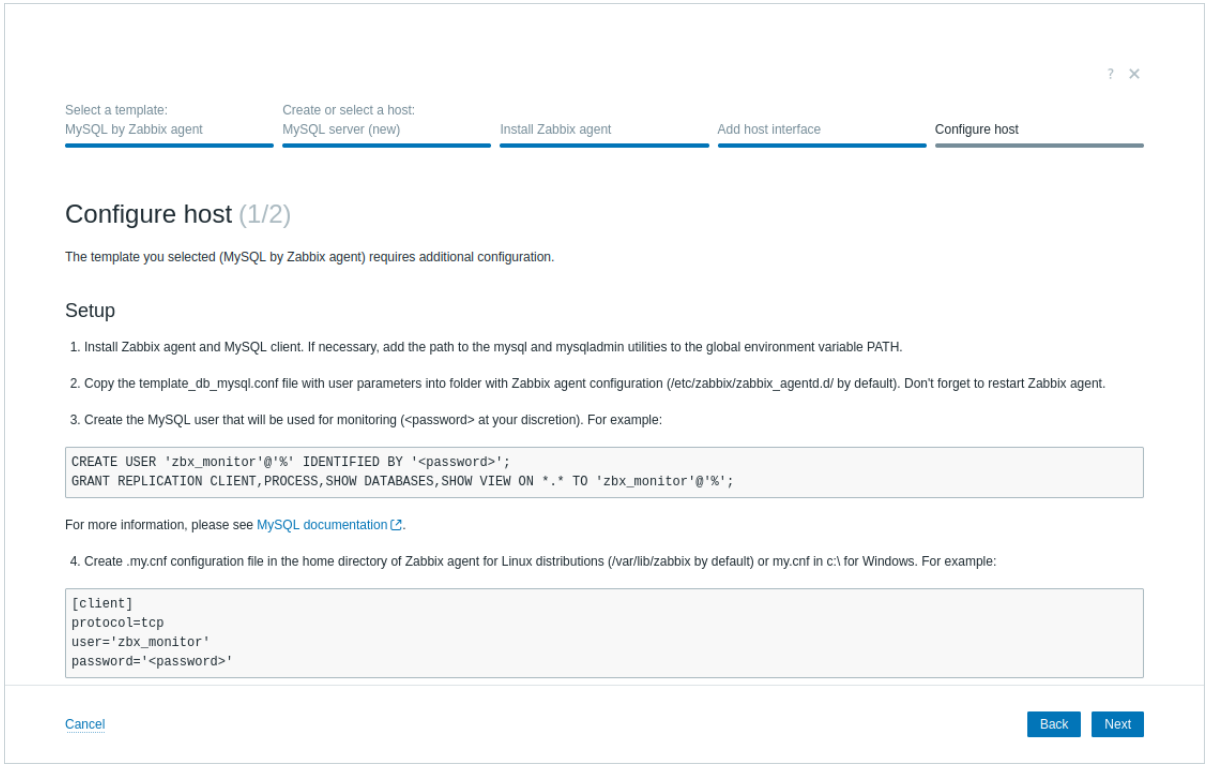
Например, вы можете использовать адрес агента по умолчанию (127.0.0.1) и порт (10050), если сервер Zabbix, агент Zabbix и сервер MySQL работают локально на одной и той же машине.



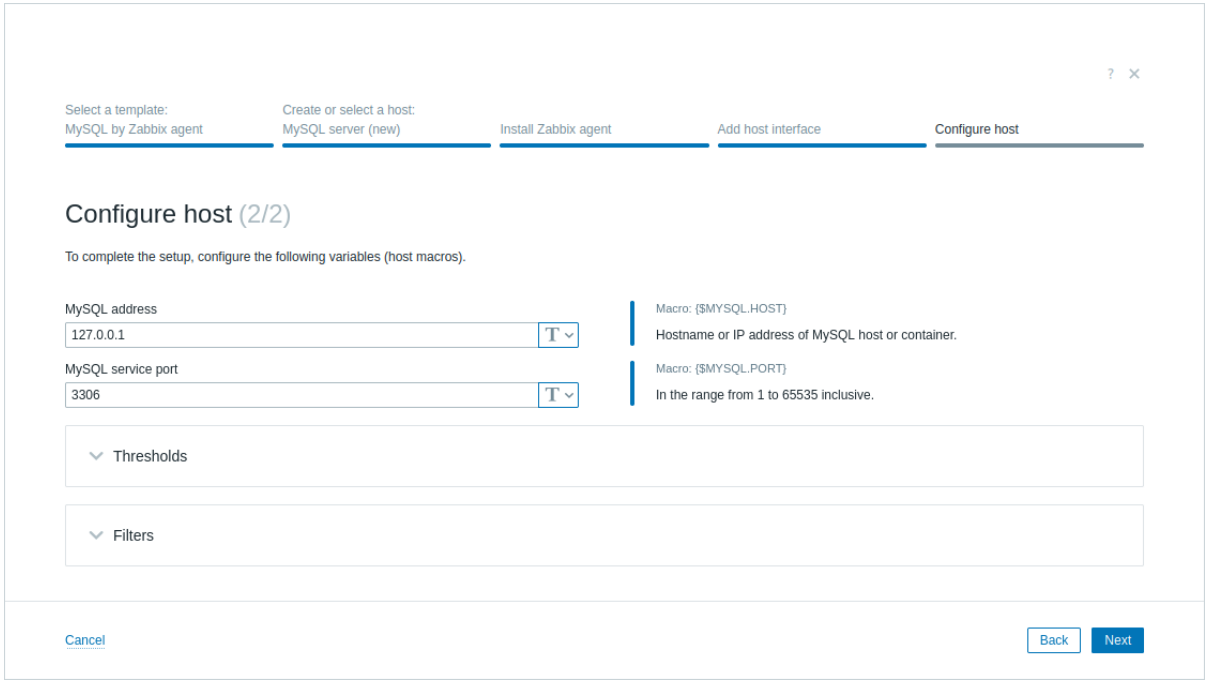
Настройка узла сети

Некоторые шаблоны требуют дополнительной настройки перед тем, как узел сети может быть создан. Это может включать:

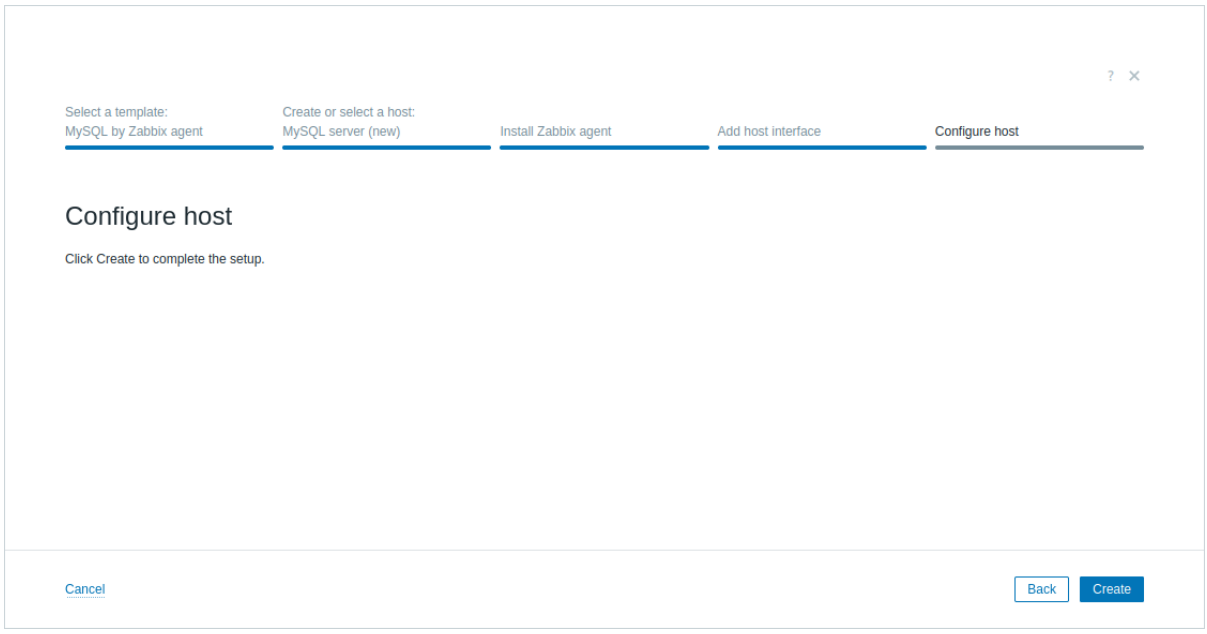
- Ручную настройку вашей цели мониторинга (например, включение определённых сервисов, создание пользователей сервисов или предоставление разрешений):



- Определение макросов узла сети — переменных, которые управляют поведением элементов данных, параметрами соединения, учётными данными для аутентификации и другими параметрами:



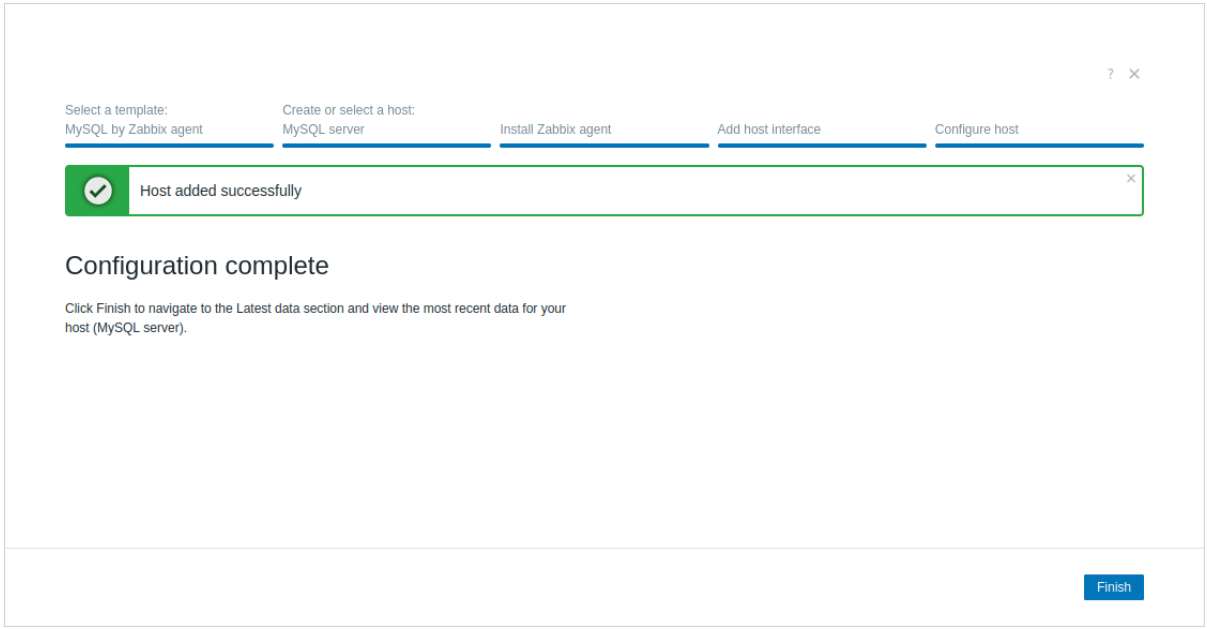
После выполнения всех необходимых шагов настройки нажмите *Create*, чтобы добавить узел сети в Zabbix (или *Update*, чтобы обновить конфигурацию существующего узла сети):



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.



2 Настройка узла сети

Обзор

Чтобы настроить узел сети в веб-интерфейсе Zabbix, выполните следующие действия:

- Перейдите в: *Data collection* > *Hosts* или *Monitoring* > *Hosts*
- Нажмите *Create host* в правом верхнем углу экрана (или на имя узла сети, чтобы изменить существующий узел сети)
- Введите параметры узла сети в форме

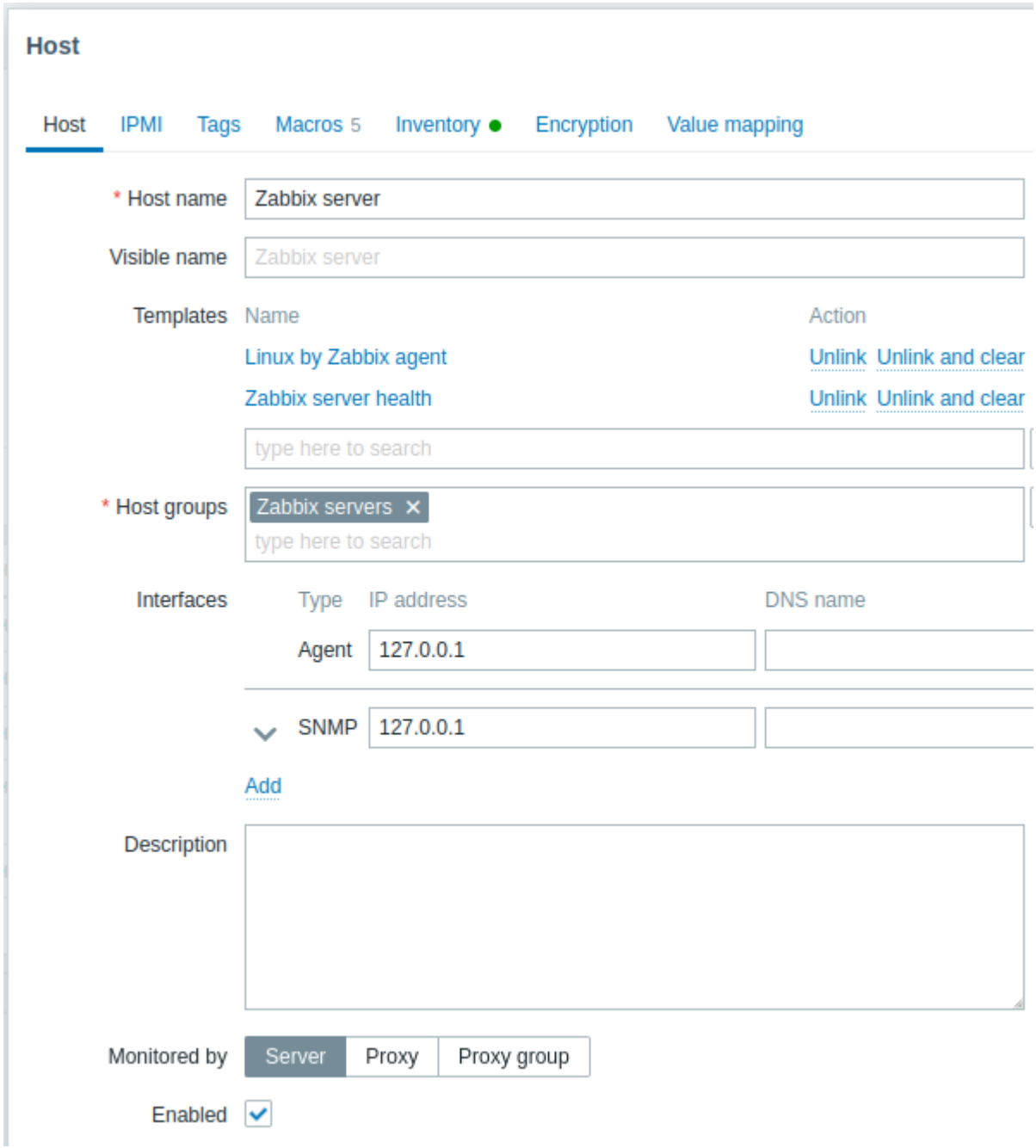
Вы также можете использовать кнопку *Clone* в форме настройки существующего узла сети, чтобы создать новый узел сети. Этот узел сети будет иметь все свойства существующего узла сети, включая связанные шаблоны, сущности (элементы данных, триггеры и т. д.) из этих шаблонов, а также сущности, напрямую прикрепленные к существующему узлу сети.

Обратите внимание, что при клонировании узла сети все сущности шаблона сохраняются в том виде, в котором они изначально заданы в шаблоне. Любые изменения этих сущностей, сделанные на уровне существующего узла сети (например, измененный интервал элемента данных, измененное регулярное выражение или добавленные прототипы в правило низкоуровневого обнаружения), не будут клонированы на новый узел сети; вместо этого они будут такими, как в шаблоне.

В качестве альтернативы вы можете использовать Host Wizard, чтобы настроить узел сети с помощью пошагового интерфейса с подсказками.

Конфигурация

Вкладка **Узел сети** содержит общие атрибуты узла сети:



Все обязательные для заполнения поля помечены красной звёздочкой.

Параметр	Описание
Имя узла сети	Введите уникальное имя узла сети. Допускаются буквенно-цифровые символы, пробелы, точки, дефисы и символы подчёркивания. Однако пробелы в начале и в конце не допускаются. *Примечание:* Если на настраиваемом узле сети запущен Zabbix агент, параметр *Hostname* в файле конфигурации агента должен иметь то же значение, что и имя узла сети, введённое здесь. Имя в параметре необходимо для обработки активных проверок.
Видимое имя	Введите уникальное видимое имя узла сети. Если вы зададите это имя, именно оно будет отображаться в списках, картах и т. д. вместо технического имени узла сети. Этот атрибут поддерживает UTF-8.

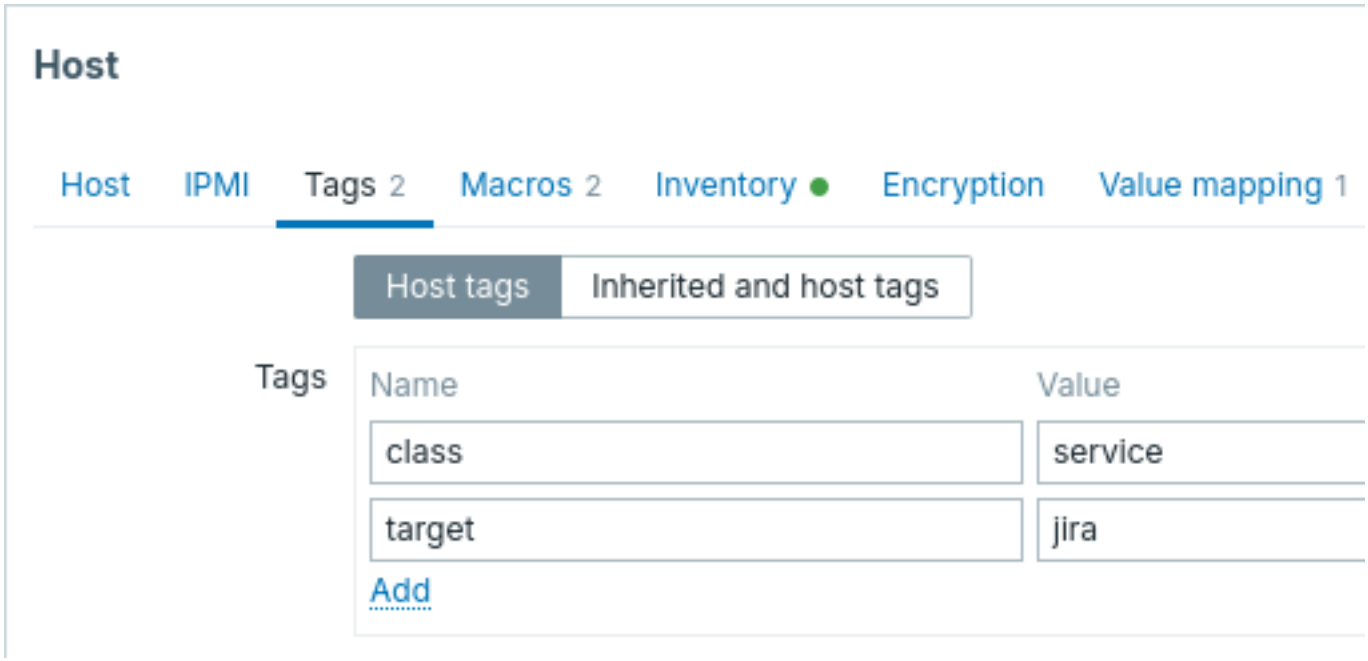
Параметр	Описание
Шаблоны	Свяжите шаблоны с узлом сети. Все сущности (элементы данных, триггеры, и т. д.) будут унаследованы из шаблона. Чтобы связать новый шаблон, начните вводить имя шаблона в текстовом поле. Появится список подходящих шаблонов; прокрутите его вниз для выбора. Также можно нажать *Выбрать* рядом с полем и выбрать шаблоны из списка во всплывающем окне. Все выбранные шаблоны будут связаны с узлом сети при сохранении или обновлении формы конфигурации узла сети. Чтобы отвязать шаблон, используйте один из двух вариантов в блоке *Связанные шаблоны*: *Отвязать* — отвязать шаблон, но сохранить его сущности (элементы данных, триггеры, и т. д.); *Отвязать и очистить* — отвязать шаблон и удалить все его сущности (элементы данных, триггеры, и т. д.). Имена шаблонов в списке являются кликабельными ссылками, ведущими к форме настройки шаблона.
Группы узлов сети	Выберите группы узлов сети, к которым принадлежит узел сети. Узел сети должен принадлежать 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 несколько типов 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 информацию *IP/DNS*, *Подключаться через* и *Порт*. *Примечание:* Интерфейсы, которые используются в каких-либо элементах данных, удалить нельзя, и ссылка *Удалить* для них неактивна. Значение ”IP” или ”DNS” из интерфейса SNMP также используется для SNMP trap. 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).
DNS-имя	DNS-имя узла сети (необязательно).
Подключаться через	Нажатие соответствующей кнопки укажет Zabbix серверу, что использовать для получения данных от агентов: **IP** — подключаться к IP-адресу узла сети (рекомендуется) **DNS** — подключаться к DNS-имени узла сети
Порт	Номер порта TCP/UDP. Значения по умолчанию: 10050 для Zabbix агента, 161 для SNMP агента, 12345 для JMX и 623 для IPMI.
По умолчанию	Установите переключатель, чтобы задать интерфейс по умолчанию.
Описание	Введите описание узла сети.
Контролируется	Выберите, кем контролируется узел сети: **Сервер** — узел сети контролируется Zabbix сервером; **Прокси** — узел сети контролируется автономным прокси; **Группа прокси** — узел сети контролируется группой прокси.
Прокси	Отображается имя назначенного прокси (только если Zabbix сервер назначил его из выбранной группы прокси). Это поле отображается только в том случае, если узел сети контролируется группой прокси.

Параметр	Описание
Включён	Если флажок установлен, узел сети включён — готов к мониторингу. Если флажок снят, узел сети отключён — не контролируется: Для пассивных запросов данных, инициированных Zabbix сервером/прокси (например, Zabbix агент, SNMP агент, простые проверки), мониторинг отключается после синхронизации конфигурации. Триггеры и действия, связанные с узлом сети, также отключаются только после перезагрузки конфигурации. Для активных проверок Zabbix агента мониторинг прекращается в течение времени (примерно 5 секунд), за которое Zabbix агент получает информацию о том, что узел сети был отключён. В течение этого короткого интервала узел сети продолжит локально собирать данные для активных проверок и пытаться отправить их на сервер/прокси; однако, поскольку узел сети помечен как *Отключён*, сервер/прокси отклонит эти данные. При отключении узла сети его элементы данных немедленно удаляются из кэша истории (кроме их последних значений, которые сохраняются для журналов).

Вкладка **IPMI** содержит атрибуты управления IPMI.

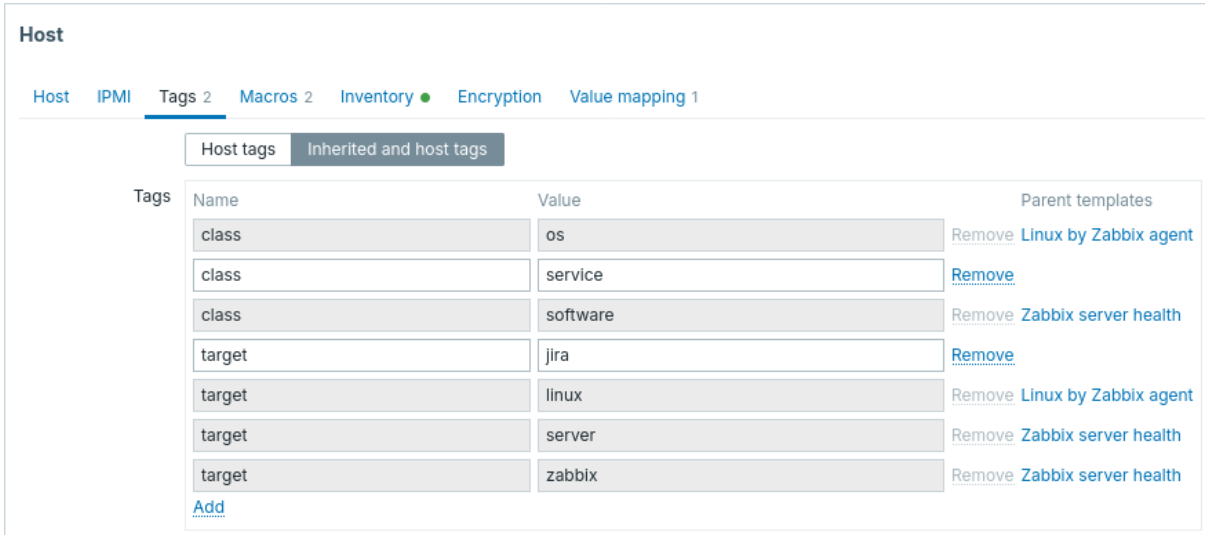
Параметр	Описание
Алгоритм аутентификации	Выберите алгоритм аутентификации.
Уровень привилегий	Выберите уровень привилегий.
Имя пользователя	Имя пользователя для аутентификации. Можно использовать пользовательские макросы.
Пароль	Пароль для аутентификации. Можно использовать пользовательские макросы.

Вкладка **Теги** позволяет определить теги на уровне узла сети. Все проблемы этого узла сети будут помечены указанными здесь значениями.



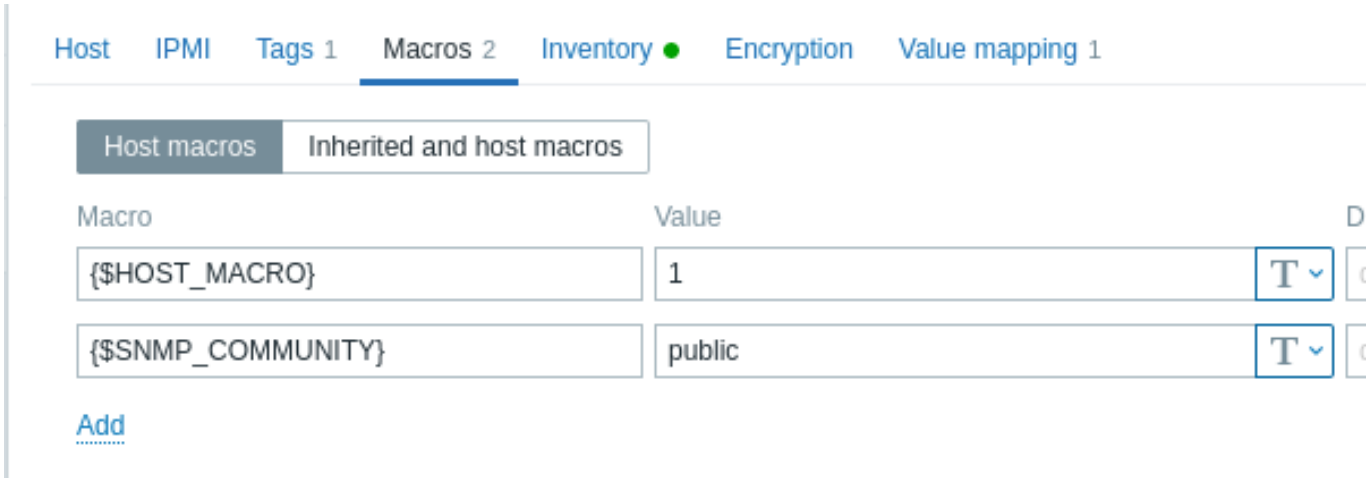
В тегах поддерживаются пользовательские макросы, макросы {INVENTORY.*}, {HOST.HOST}, {HOST.NAME}, {HOST.CONN}, {HOST.DNS}, {HOST.IP}, {HOST.PORT} и {HOST.ID}.

Здесь также можно просматривать теги уровня шаблона, если выбрать опцию *Унаследованные и теги узла сети*. Здесь отображаются все определённые пользовательские теги узла сети, а также их источник.

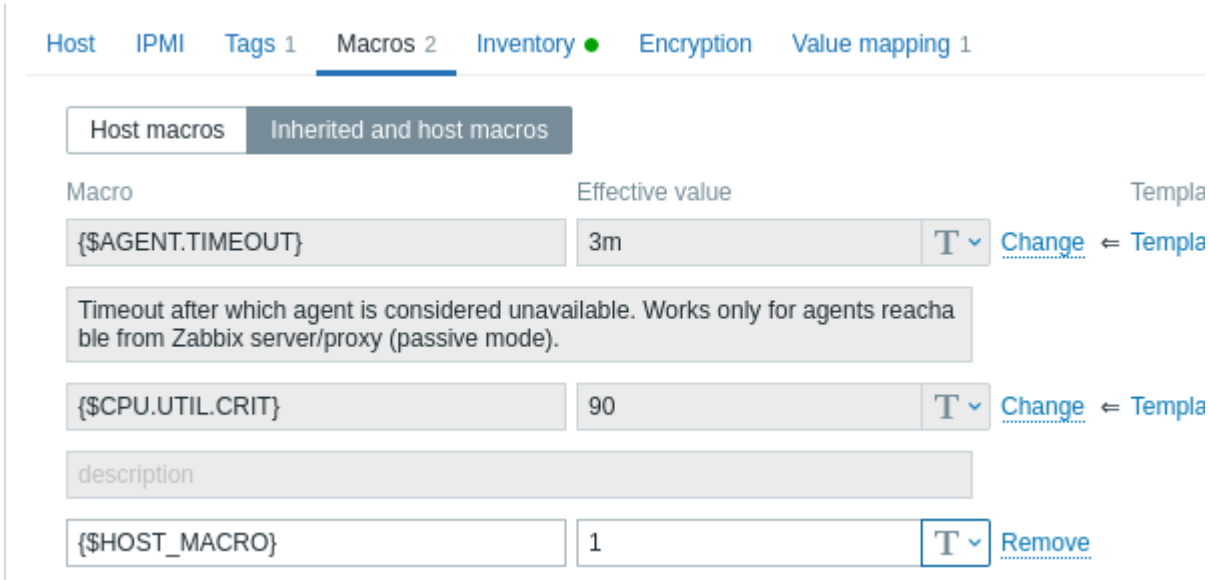


Для удобства предусмотрены ссылки на соответствующие шаблоны.

Вкладка **Макросы** позволяет определять пользовательские макросы на уровне узла сети в виде пар имя-значение. Обратите внимание, что значения макросов могут храниться как обычный текст, секретный текст или секрет Vault. Также поддерживается добавление описания.

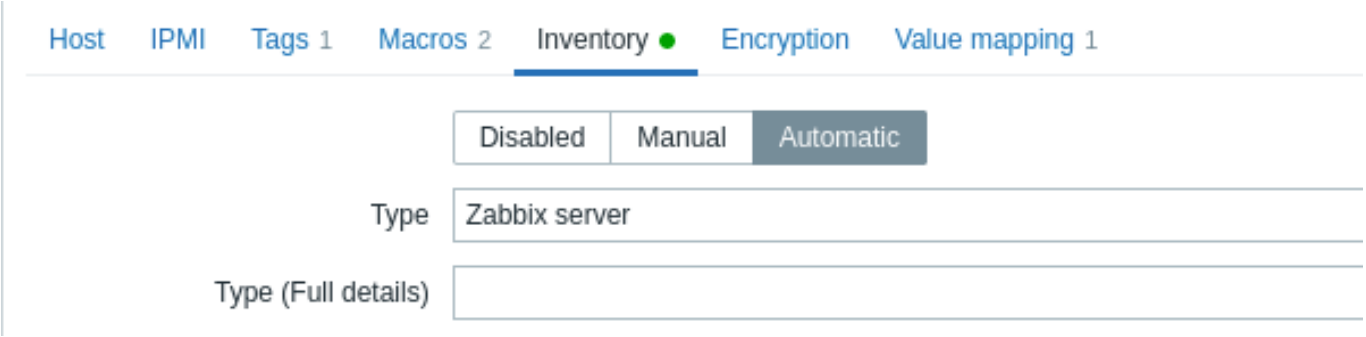


Здесь также можно просматривать пользовательские макросы уровня шаблона и глобальные пользовательские макросы, если выбрать опцию *Унаследованные и макросы узла сети*. Здесь отображаются все определённые для узла сети пользовательские макросы, их итоговые значения, а также их источник.



Для удобства предусмотрены ссылки на соответствующие шаблоны и конфигурацию глобальных макросов. Также можно редактировать макрос шаблона/глобальный макрос на уровне узла сети, фактически создавая копию макроса на узле сети.

Вкладка **Инвентаризация** позволяет вручную вводить информацию инвентаризации для узла сети. Также можно включить *Автоматическое* заполнение инвентаризации или отключить заполнение инвентаризации для этого узла сети.



Если инвентаризация включена (вручную или автоматически), рядом с названием вкладки отображается зелёная точка.

Шифрование

Вкладка **Шифрование** позволяет требовать зашифрованные соединения с узлом сети.

Параметр	Описание
Соединения с узлом сети	Как сервер Zabbix или прокси подключается к агенту Zabbix на узле сети: без шифрования (по умолчанию), с использованием PSK (предварительно разделённого ключа) или сертификата.
Соединения от узла сети	Выберите, какие типы соединений разрешены от 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 используется ”Без шифрования”.
Issuer	Разрешённый издатель сертификата. Сначала сертификат проверяется с помощью CA (центра сертификации). Если он действителен и подписан этим CA, то поле *Issuer* можно использовать для дополнительного ограничения допустимого CA. Это поле предназначено для использования, если ваша установка Zabbix использует сертификаты от нескольких CA. Если это поле пустое, принимается любой CA.
Subject	Разрешённый субъект сертификата. Сначала сертификат проверяется с помощью CA. Если он действителен и подписан этим CA, то 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 разделённого ключа. Не указывайте конфиденциальную информацию в идентификаторе PSK, так как он передаётся по сети в незашифрованном виде, чтобы сообщить получателю, какой PSK следует использовать.
PSK	Предварительно разделённый ключ (hex-строка). Максимальная длина: 512 шестнадцатеричных цифр (PSK размером 256 байт), если Zabbix использует библиотеку GnuTLS или OpenSSL, и 64 шестнадцатеричных цифры (PSK размером 32 байта), если Zabbix использует библиотеку mbed TLS (PolarSSL). Пример: 1f87b595725ac58dd977beef14b97461a7c1045b9a1c963065002c5473194952

Преобразование значений

Вкладка **Преобразование значений** позволяет настроить удобное для пользователя представление данных элемента данных в преобразованиях значений.

3 Настройка группы узлов сети

Обзор

Группы узлов сети используются для логического объединения узлов сети и назначения им пользовательских прав доступа.

Каждому узлу сети должна быть назначена как минимум одна группа узлов сети. Узел сети может принадлежать нескольким группам узлов сети, и 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. Поэтому, даже если одному пользователю нужен доступ к одному узлу сети, он предоставляется

путем добавления пользователя в группу пользователей, у которой есть право доступа к группе узлов сети, содержащей этот узел сети.

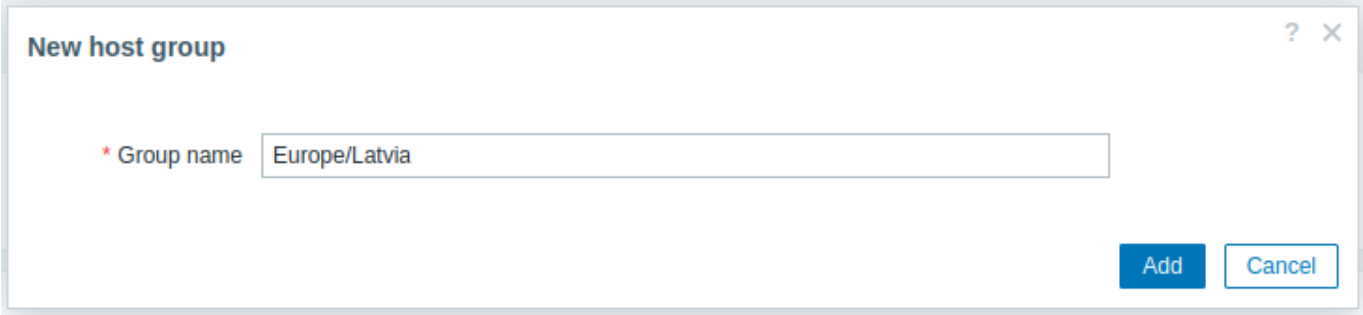
Настройка

> **Attention:**
> Только пользователи с ролью Super admin могут создавать группы узлов сети.

В веб-интерфейсе Zabbix есть два способа создания группы узлов сети.

Первый способ:

- Перейдите в: *Сбор данных* → *Группы узлов сети*
- Нажмите *Создать группу узлов сети* в правом верхнем углу экрана
- Введите имя группы в форме



Второй способ: при настройке узла сети введите несуществующее имя группы в поле ввода *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 её.

Удаление группы узлов сети удаляет только логическую группу, но 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, в которую она входит.

Подгруппа создаётся с использованием прямой косой черты '/' в поле ввода имени группы для обозначения её связи с родительской группой(ами). Например:

- ввод `Europe/Latvia` создаёт подгруппу `Europe/Latvia` родительской группы `Europe`.
- ввод `Europe/Latvia/Riga/Zabbix servers` создаёт соответствующую подгруппу во вложенных родительских группах `Europe`, `Europe/Latvia`, `Europe/Latvia/Riga`.

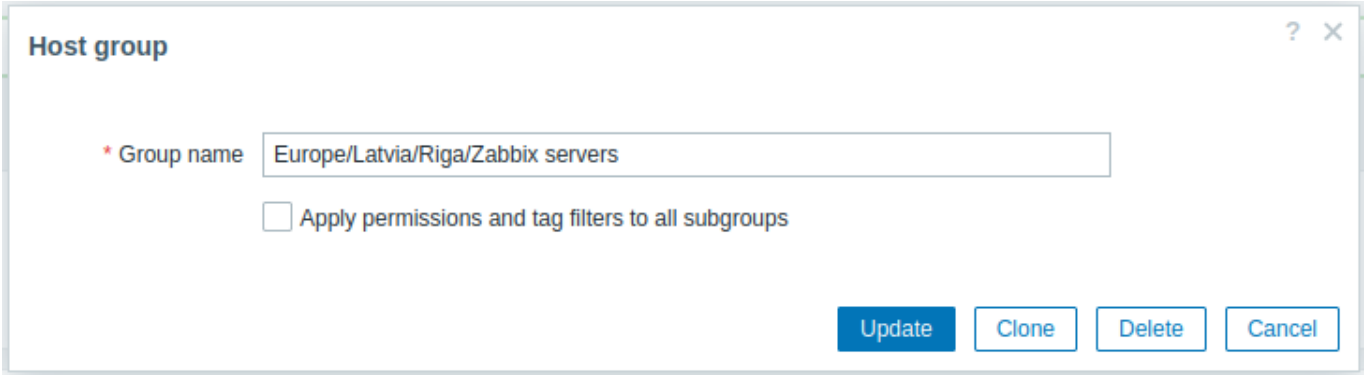
При создании подгруппы не допускается использование начальных или конечных косых черт, а также нескольких косых черт подряд. Экранирование '/' не поддерживается.

Перед созданием подгруппы не требуется предварительно создавать какие-либо родительские группы узлов сети. Вы можете выбрать, с чего начать: с создания подгруппы (например, `Europe/Latvia`) или любой родительской группы узлов сети (в нашем примере — `Europe`). Если вы начнёте с создания подгруппы, родительские 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), права группы пользователей для подгруппы наследуются от родительской группы.
- При создании родительской группы узлов сети для существующей подгруппы (например, при создании `Europe`, когда `Europe/Latvia` уже существует), права доступа к родительской группе не задаются.

При редактировании любой группы узлов сети также можно задать дополнительную опцию *Применить права доступа и фильтры тегов ко всем подгруппам*.



Если установить этот флажок и нажать *Обновить*, одинаковый уровень прав доступа и фильтры тегов будут применены 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 этого флажка предоставит всем текущим и будущим подгруппам те же пользовательские права и права на основе тегов, что и у редактируемой группы.

Обратите внимание, что эта опция не сохраняется в базе данных и переопределит существующие права доступа. Любые изменения, внесённые с помощью этой опции, можно отменить только вручную.

4 Инвентаризация

Обзор

Вы можете хранить инвентарные данные сетевых устройств в Zabbix.

В веб-интерфейсе Zabbix существует специальное меню *Инвентаризация*. 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* вы можете ввести такие подробности, как тип устройства, серийный номер, местоположение, ответственное лицо, ссылки и прочее — они пополнят собой инвентарные данные.

Если в инвентарные данные включён URL и он начинается с «http» или «https», в разделе *Инвентаризация* это отобразится в виде ссылки.

Автоматический режим

Также инвентарные данные узлов сети могут заполняться автоматически. Чтобы эта функция заработала, при настройке узла сети на вкладке *Инвентаризация* для режима инвентарных данных должно быть выставлено *Автоматически*.

Затем вы можете настроить элементы данных узла сети таким образом, чтобы они заполняли любые поля инвентарных данных своими значениями. Для этого при настройке элемента данных в соответствующем атрибуте (называемом *Заполнение поля инвентаря узла сети*) нужно указать поле назначения.

Элементы данных, которые особенно полезны для автоматического сбора инвентарных данных:

- system.hw.chassis[full|type|vendor|model|serial] — по умолчанию [full], требуются права root
- system.hw.cpu[all|cpunum,full|maxfreq|vendor|model|curfreq] — по умолчанию [all,full]
- system.hw.devices[pci|usb] — по умолчанию [pci]
- system.hw.macaddr[интерфейс,short|full] — по умолчанию [all,full], интерфейс является регулярным выражением
- system.sw.arch
- system.sw.os[name|short|full] — по умолчанию [name]
- system.sw.packages[пакет,менеджер,short|full] — по умолчанию [all,all,full], пакет является регулярным выражением

Выбор режима инвентаризации

Режим инвентаризации можно выбрать в диалоге настройки узла сети.

Режим инвентаризации по умолчанию для новых узлов сети выбирается на основе настройки *Режим инвентаризации узлов сети по умолчанию* в *Администрирование* → *Общие* → *Другое*.

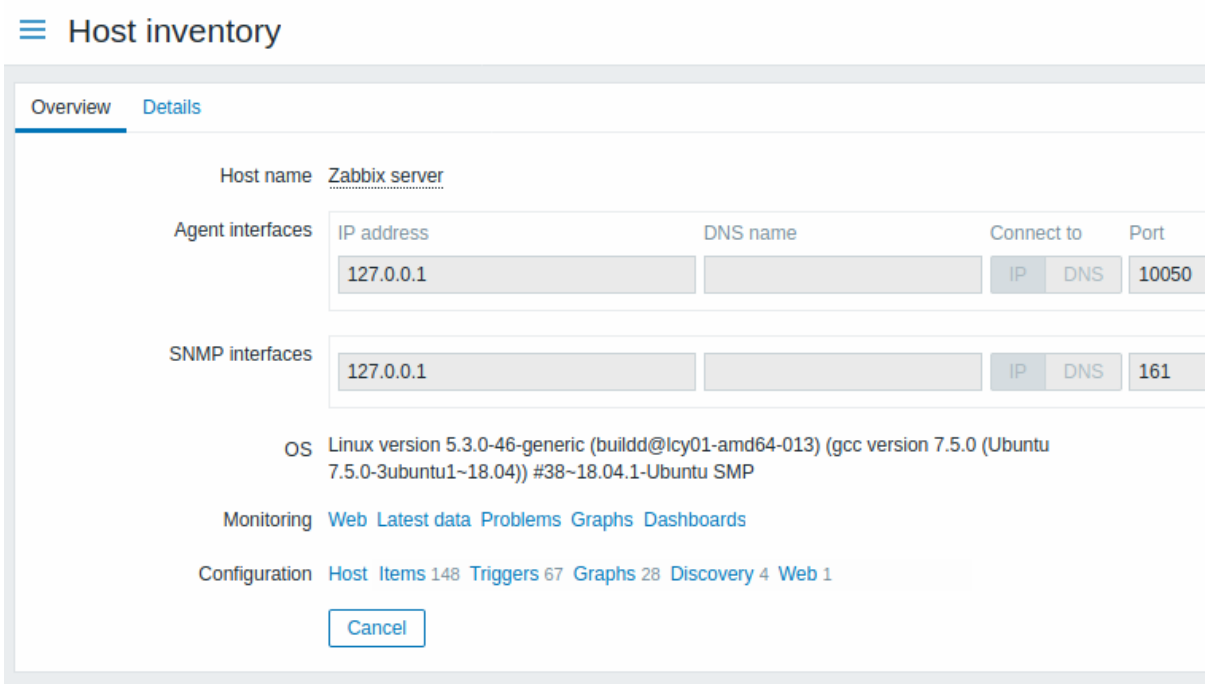
Для узлов сети, добавляемых с помощью действий по сетевым обнаружениям или авторегистрации, имеется возможность указать операцию *Задать режим инвентарных данных узлов сети*, выбрав ручной или автоматический режим. Эта операция переопределит настройку *Режим инвентаризации узлов сети по умолчанию*.

Просмотр инвентарных данных

Подробности всех существующих инвентарных данных доступны в меню *Инвентаризация*.

В *Инвентаризация → Обзор* вы можете получить количество узлов сети, сгруппированных по различным полям инвентарных данных.

В *Инвентаризация → Узлы сети* вы можете увидеть все узлы сети, содержащие инвентарные данные. При нажатии на имя узла сети откроется диалоговое окно с подробностями.



Вкладка **Обзор** отображает:

Параметр	Описание
Имя узла сети	Имя узла сети. При нажатии мышью на имя появится меню с заданными для этого узла сети скриптами. В случае, если узел сети находится в обслуживании, его имя отображается с оранжевой иконкой.
Видимое имя	Видимое имя узла сети (если задано).
Интерфейсы (Агент, SNMP, JMX, IPMI) узла сети	Этот блок отображает подробности всех интерфейсов, настроенных у узла сети.
ОС	Поле операционная система из инвентарных данных узла сети (если задано).
Аппаратное обеспечение	Поле аппаратного обеспечения из инвентарных данных узла сети (если задано).
Программное обеспечение	Поле программного обеспечения из инвентарных данных узла сети (если задано).
Описание	Описание узла сети.
Мониторинг	Ссылки на разделы мониторинга этого узла сети: *Веб*, *Последние данные*, *Проблемы*, *Графики*, *Панели*.
Настройка	Ссылки на разделы настроек этого узла сети: *Узел сети*, *Элементы данных*, *Триггеры*, *Графики*, *Обнаружение*, *Веб*. После каждой ссылки указано количество сконфигурированных объектов.

Вкладка **Детали** отображает все поля инвентарных данных, которые заполнены (не пустые).

Макросы инвентаризации

Макросы инвентаризации узла сети {INVENTORY.*} доступны для использования в уведомлениях, например:

”Сервер в {INVENTORY.LOCATION1} имеет проблему, ответственное лицо — {INVENTORY.CONTACT1}, номер телефона {INVENTORY.POC.PRIMARY.PHONE.A1}.”

Подробнее смотрите на странице Поддерживаемые макросы.

5 Массовое обновление

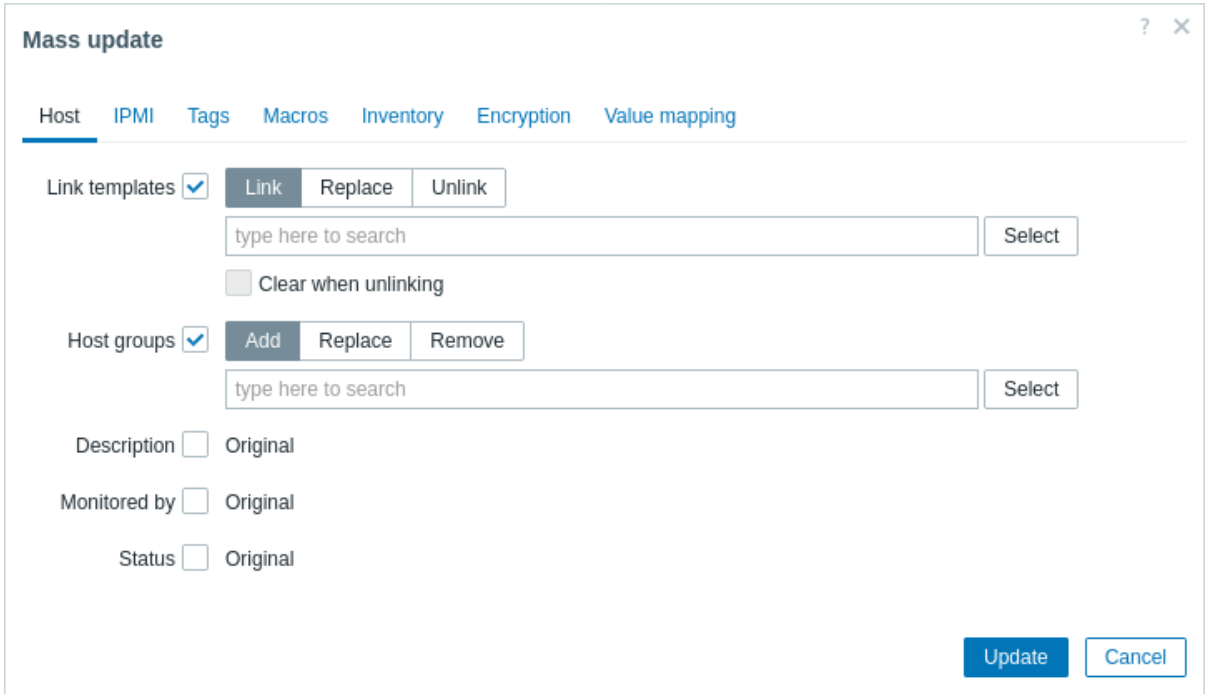
Обзор

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 массово обновить несколько узлов сети, выполните следующие действия:

- Установите флажки напротив узлов сети, которые вы хотите обновить, в списке узлов сети
- Нажмите *Mass update* под списком
- Перейдите на вкладку с требуемыми атрибутами (*Host*, *IPMI*, *Tags*, *Macros*, *Inventory*, *Encryption* или *Value mapping*)
- Установите флажки для любых атрибутов, которые нужно обновить, и введите для них новые значения



При выборе соответствующей кнопки для обновления привязки **шаблонов** доступны следующие параметры:

- *Link* — укажите, какие дополнительные шаблоны нужно привязать
- *Replace* — укажите, какие шаблоны нужно привязать, одновременно отвязав все шаблоны, которые ранее были привязаны к узлам сети
- *Unlink* — укажите, какие шаблоны нужно отвязать

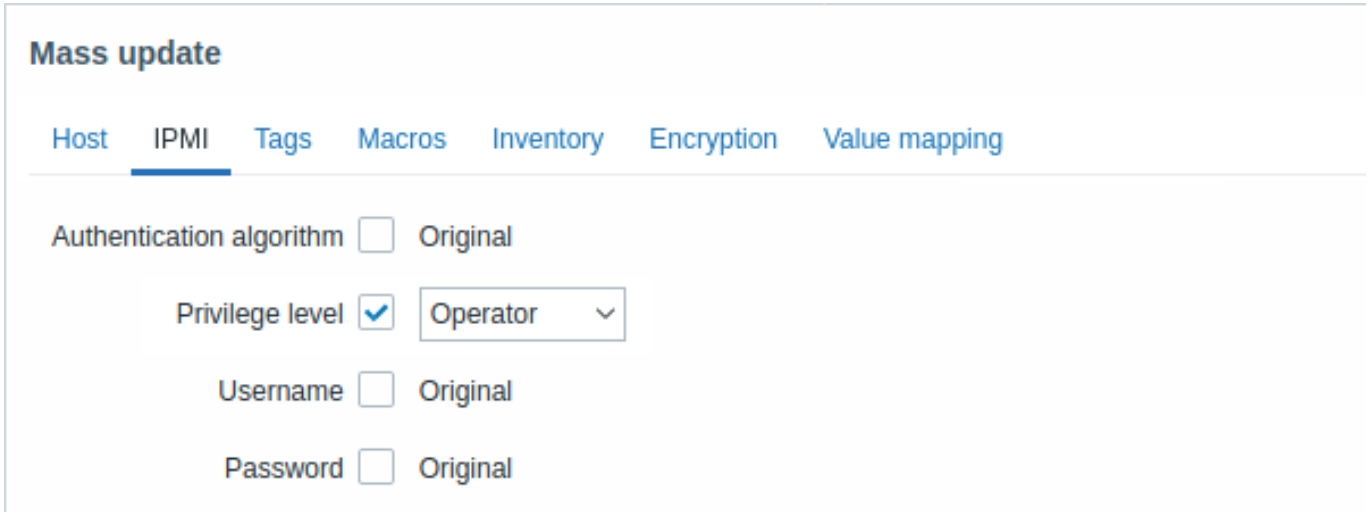
Чтобы указать шаблоны для привязки/отвязки, начните вводить имя шаблона в поле автодополнения, пока не появится выпадающий список с подходящими шаблонами. Просто прокрутите список вниз и выберите нужный шаблон.

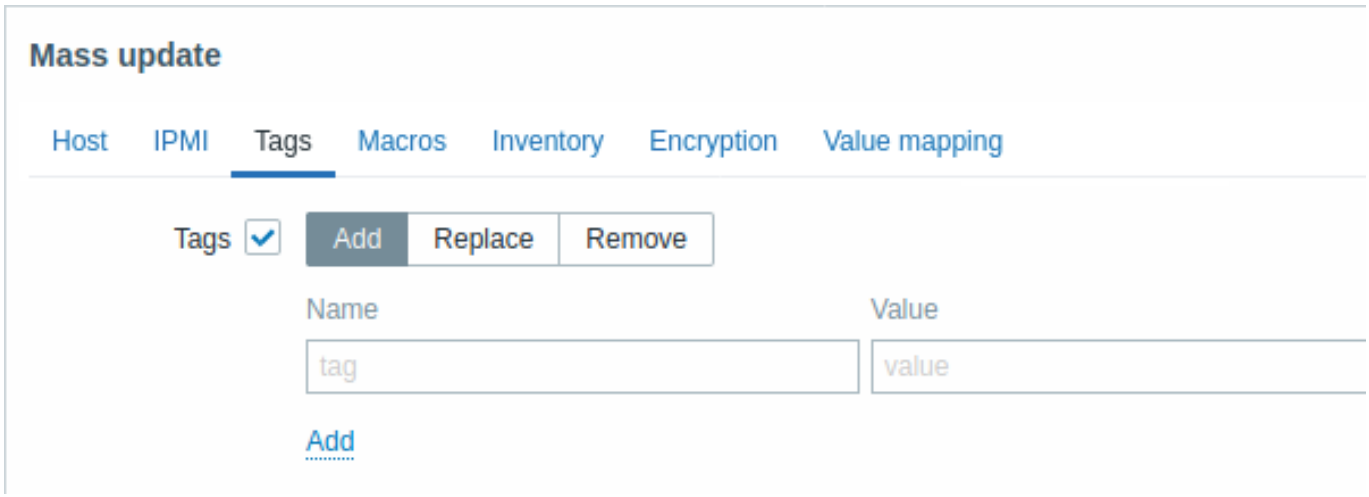
Параметр *Clear when unlinking* позволяет не только отвязать все ранее привязанные шаблоны, но и удалить все унаследованные от них элементы данных (элементы данных, триггеры и т. д.).

При выборе соответствующей кнопки для обновления **групп узлов сети** доступны следующие параметры:

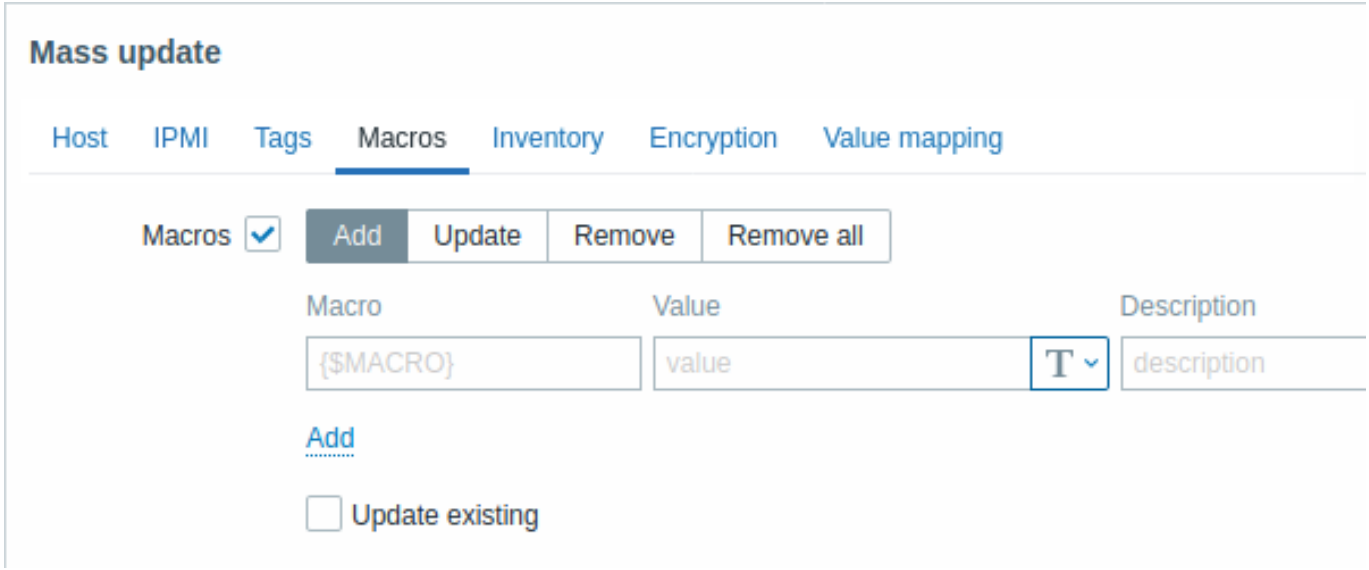
- *Add* — позволяет указать дополнительные группы узлов сети из уже существующих или ввести для узлов сети совершенно новые группы узлов сети
- *Replace* — удалит узел сети из всех существующих групп узлов сети и заменит их на группу(ы), указанные в этом поле (существующие или новые группы узлов сети)
- *Remove* — удалит указанные группы узлов сети у узлов сети

Эти поля поддерживают автодополнение — при вводе в них предлагается выпадающий список подходящих групп узлов сети. Если группа узлов сети новая, она также появится в выпадающем списке и будет помечена как *(new)* после строки. Просто прокрутите список вниз и выберите нужный вариант.





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 и тому же узлу сети.

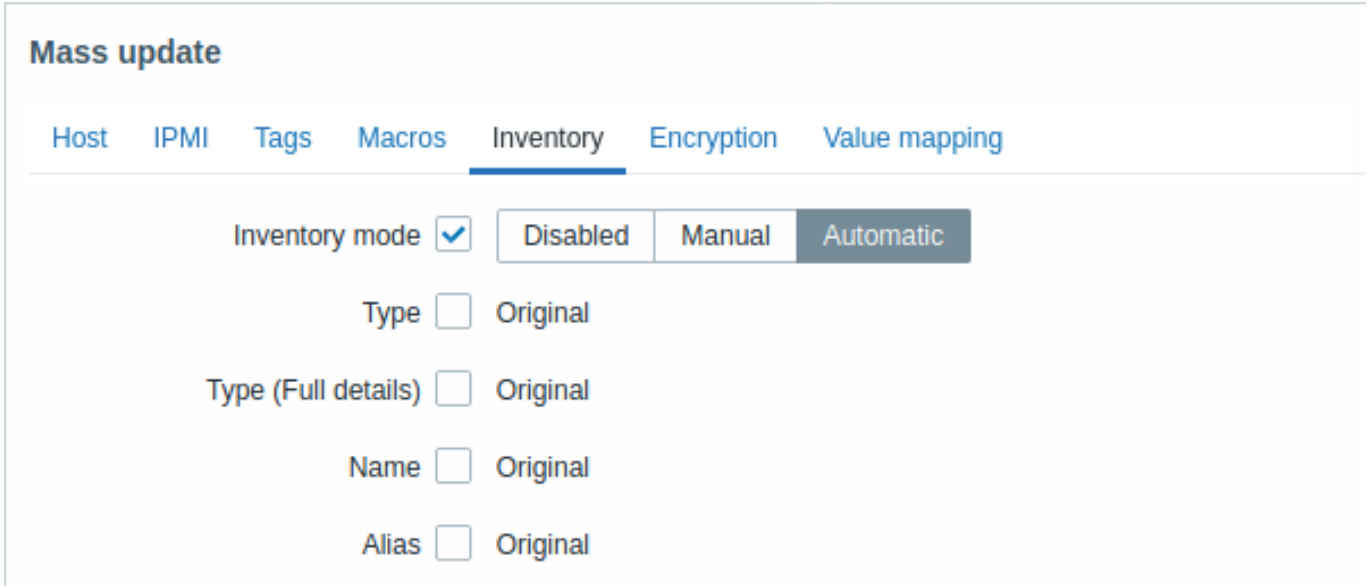


При выборе соответствующей кнопки для обновления макросов доступны следующие параметры:

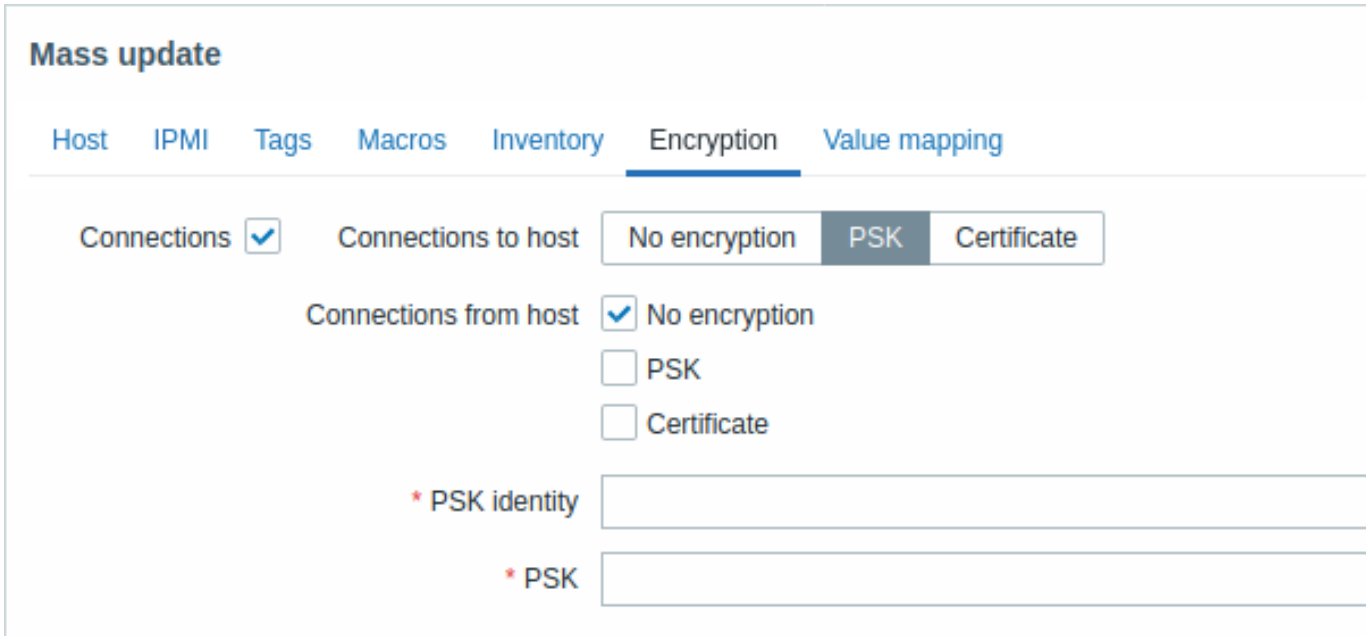
- *Add* — позволяет указать дополнительные пользовательские макросы для узлов сети. Если установлен флажок *Update existing*, будут обновлены значение, тип и описание для указанного имени макроса. Если флажок не установлен и макрос с таким именем уже существует на узле сети, он не будет обновлен.
- *Update* — заменит значения, типы и описания макросов, указанных в этом списке. Если установлен флажок *Add missing*, макрос, который ранее не существовал на узле сети, будет добавлен как новый макрос. Если флажок не установлен, будут обновлены только макросы, которые уже существуют на узле сети.
- *Remove* — удалит указанные макросы у узлов сети. Если установлен флажок *Except selected*, будут удалены все

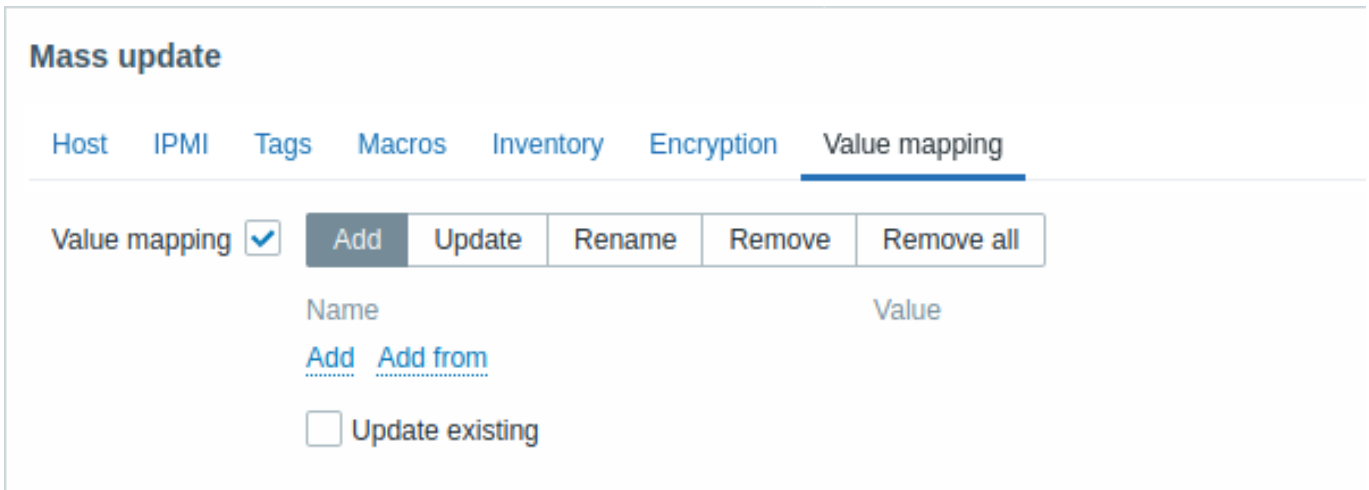
макросы, **кроме** указанных в списке. Если флажок не установлен, будут удалены только макросы, указанные в списке.

- *Remove all* — удалит все пользовательские макросы у узлов сети. Если флажок *I confirm to remove all macros* не установлен, откроется новое всплывающее окно с запросом подтверждения удаления всех макросов.



Чтобы можно было массово обновлять поля инвентарных данных, параметр *Inventory mode* должен быть установлен в значение 'Manual' или 'Automatic'.





Для обновления преобразования значений доступны кнопки со следующими параметрами:

- *Add* — добавить преобразования значений к узлам сети. Если установлен флажок *Update existing*, будут обновлены все свойства преобразования значений с этим именем. В противном случае, если преобразование значений с таким именем уже существует, оно не будет обновлено.
- *Update* — обновить существующие преобразования значений. Если установлен флажок *Add missing*, преобразование значений, которое ранее не существовало на узле сети, будет добавлено как новое преобразование значений. В противном случае будут обновлены только те преобразования значений, которые уже существуют на узле сети.
- *Rename* — задать новое имя существующему преобразованию значений
- *Remove* — удалить указанные преобразования значений у узлов сети. Если установлен флажок *Except selected*, будут удалены все преобразования значений, **кроме** указанных.
- *Remove all* — удалить все преобразования значений у узлов сети. Если флажок *I confirm to remove all value maps* не установлен, откроется новое всплывающее окно с запросом подтверждения удаления.

После внесения всех необходимых изменений нажмите *Update*. Атрибуты будут соответствующим образом обновлены для всех выбранных узлов сети.

2. 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.

1. Создание элемента данных

Обзор

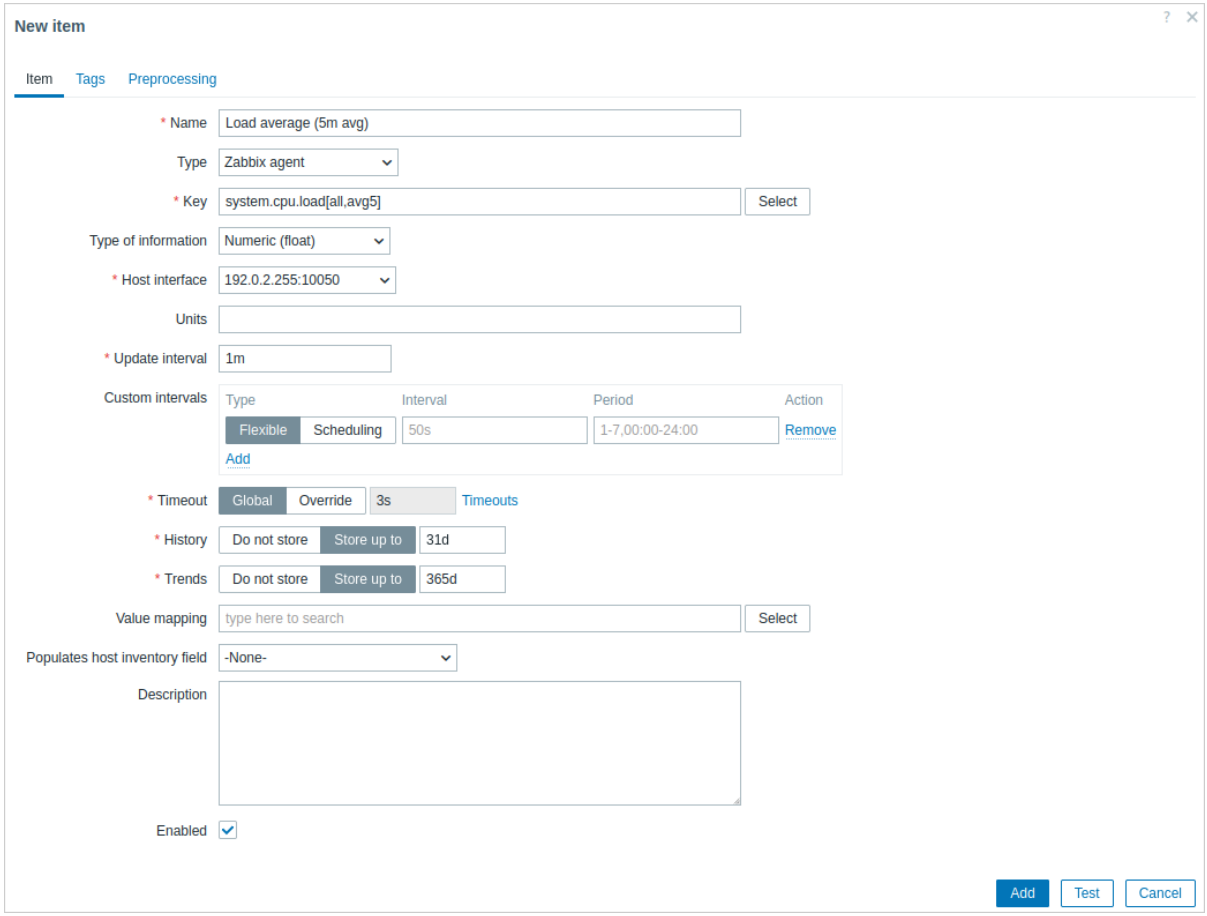
Чтобы создать элемент данных в веб-интерфейсе Zabbix, выполните следующие действия:

- Перейдите в: *Сбор данных* > *Узлы сети*
- Нажмите *Элементы данных* в строке нужного узла сети
- Нажмите *Создать элемент данных* в правом верхнем углу экрана
- Введите параметры элемента данных в форме

Вы также можете создать элемент данных, открыв существующий, нажав кнопку *Клонировать*, а затем сохранив его под другим именем.

Конфигурация

Вкладка **Элемент данных** содержит общие атрибуты элемента данных.



Все обязательные поля ввода отмечены красной звёздочкой.

Параметр	Описание
Name	Имя элемента данных. Поддерживаются пользовательские макросы.
Type	Тип элемента данных. См. отдельные разделы типов элементов данных.
Key	Ключ элемента данных (до 2048 символов). Поддерживаемые ключи элементов данных можно найти в отдельных разделах по типам 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 Zabbix агентом или Zabbix сервером. См. также: правильный формат ключа.
Type of information	Тип данных, используемый для проверки значения элемента данных и его хранения в базе данных, в том числе после любых преобразований: **Numeric (unsigned)** — 64-битное целое число без знака; **Numeric (float)** — 64-битное число с плавающей точкой; **Character** — короткий текст; **Log** — длинный текст с атрибутами журнала или без них (метка времени, источник, важность, ID события журнала); **Text** — длинный текст; **Binary** — двоичное число (поддерживается только для зависимых элементов данных); **JSON** — структурированные данные JSON, хранящиеся в базе данных в исходном виде (не поддерживается для вычисляемых элементов данных). Элементы данных, возвращающие значения с типом данных binary или JSON, не поддерживаются в формулах вычисляемых элементов данных или выражениях триггеров. См. ограничения данных элемента данных для ограничений хранения по каждому типу данных и используемой СУБД. Для элементов данных, возвращающих данные только в одном определённом формате, соответствующий тип данных выбирается автоматически — например, `system.cpu.load` и `system.cpu.util` оба возвращают 64-битное число с плавающей точкой, поэтому *Numeric (float)* выбирается автоматически.

Параметр	Описание
Host interface	Выберите интерфейс узла сети. Это поле доступно при редактировании элемента данных на уровне узла сети.
Units	Если задан символ единицы измерения, Zabbix применяет постобработку к полученному значению 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, отображаются с использованием до трёх старших ненулевых единиц времени (881764 → 10d 4h 56m) **uptime** — прошедшее время в формате hh:mm:ss или N days, hh:mm:ss (881764 → 10 days, 04:56:04) **unixtime** — временная метка Unix, форматируется как yyyy.mm.dd hh:mm:ss (881764 → 1970-01-11 04:56:04 AM); для корректного форматирования полученное значение элемента данных должно иметь тип *Numeric (unsigned)*. Для других единиц измерения (например, Hz, W 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 префикс !, преобразование единиц и добавление префиксов отключаются (1024 !B → 1024 B, 61 !s → 61 s). Дополнительные примеры и подробности о символах единиц измерения и преобразовании единиц см. в разделе Суффиксы значений элементов данных.
Update interval	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. Одно макро-выражение должно заполнять всё поле целиком. Несколько макросов в одном поле или макросы, смешанные с текстом, не поддерживаются. *Примечание*: интервал обновления может быть установлен в '0' только если существуют пользовательские интервалы с ненулевым значением. Если установлено '0' и существует пользовательский интервал (гибкий или по расписанию) с ненулевым значением, элемент данных будет опрашиваться в течение действия пользовательского интервала. *Обратите внимание*, что первый опрос элемента данных после того, как он стал активным, или после изменения интервала обновления может произойти раньше заданного значения. Новые элементы данных будут проверены в течение 60 секунд после их создания, если только для них не задано расписание или гибкий интервал обновления, а *Update interval* не установлен в 0. Для существующего пассивного элемента данных можно немедленно запросить значение, нажав кнопку *Execute now* button.
Custom intervals	Вы можете создать пользовательские правила для проверки элемента данных: **Flexible** — создать исключение для *Update interval* (интервал с другой частотой). **Scheduling** — создать пользовательское расписание опроса. Подробную информацию см. в разделе Пользовательские интервалы. В поле *Interval* поддерживаются суффиксы времени, например 30s, 1m, 2h, 1d. Поддерживаются пользовательские макросы. Одно макро-выражение должно заполнять всё поле целиком. Несколько макросов в одном поле или макросы, смешанные с текстом, не поддерживаются.
Timeout	Задайте тайм-аут проверки элемента данных (доступно для поддерживаемых типов элементов данных). Выберите вариант тайм-аута: **Global** — используется тайм-аут прокси/глобальный тайм-аут (отображается в неактивном поле *Timeout*). **Override** — используется пользовательский тайм-аут (задаётся в поле *Timeout*; допустимый диапазон: 1 - 600s). Поддерживаются суффиксы времени, например 30s, 1m, и пользовательские макросы. Нажатие на ссылку *Timeouts* позволяет настроить тайм-ауты прокси или глобальные тайм-ауты (если прокси не используется). Обратите внимание, что ссылка *Timeouts* видна только пользователям типа *Super admin* с правами на разделы веб-интерфейса *Administration* > *General* или *Administration* > *Proxies*.

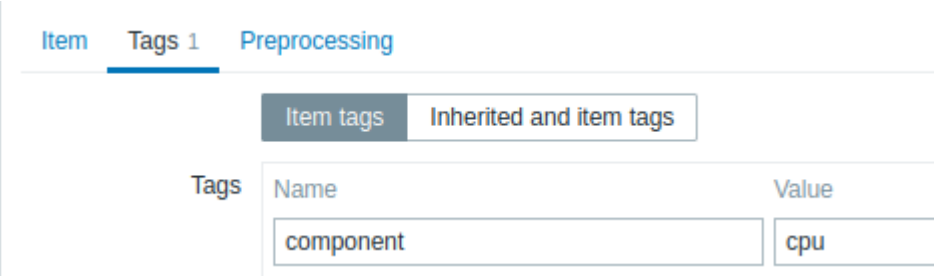
Параметр	Описание
History	Выберите одно из значений: **Do not store** — история элемента данных не сохраняется. Полезно для мастер-элементов данных, если историю должны хранить только зависимые элементы данных. Этот параметр не может быть переопределён глобальными настройками housekeeper. **Store up to** — укажите срок хранения подробной истории в базе данных (от 1 часа до 25 лет). Более старые данные будут удалены housekeeper. Хранится в секундах. Поддерживаются суффиксы времени, например 2h, 1d. Поддерживаются пользовательские макросы. Значение *Store up to* может быть глобально переопределено в *Administration* > *Housekeeping*. Если существует глобальная настройка переопределения, отображается оранжевый значок информации  . Если навести на него указатель мыши, будет показано предупреждающее сообщение, например: *Overridden by global housekeeper settings (1d)*. Рекомендуется хранить записанные значения как можно меньшее время, чтобы уменьшить размер истории значений в базе данных. Вместо хранения длительной истории значений можно дольше хранить данные трендов. См. также История и тренды.
Trends	Выберите одно из значений: **Do not store** — тренды не сохраняются. Этот параметр не может быть переопределён глобальными настройками housekeeper. **Store up to** — укажите срок хранения агрегированной истории (почасовые min, max, avg, count) в базе данных (от 1 дня до 25 лет). Более старые данные будут удалены housekeeper. Хранится в секундах. Поддерживаются суффиксы времени, например 24h, 1d. Поддерживаются пользовательские макросы. Значение *Store up to* может быть глобально переопределено в *Administration* > *Housekeeping*. Если существует глобальная настройка переопределения, отображается оранжевый значок информации  . Если навести на него указатель мыши, будет показано предупреждающее сообщение, например: *Overridden by global housekeeper settings (7d)*. *Примечание:* хранение трендов недоступно для нечисловых данных — character, log и text. См. также История и тренды.
Value mapping	Применить сопоставление значений к этому элементу данных. Сопоставление значений не изменяет полученные значения, оно используется только для отображения данных. Работает с элементами данных типов *Numeric(unsigned)*, *Numeric(float)* и *Character*. Например, ”Windows service states”.
Log time format	Доступно только для элементов данных типа **Log**. 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 оставить пустым, временная метка будет установлена в 0 по времени Unix, что соответствует 1 января 1970 года. Например, рассмотрим следующую строку из файла журнала Zabbix агента: ” 23480:20100328:154718.045 Zabbix agent started. Zabbix 1.8.2 (revision 11211).” Она начинается с шести символьных позиций для PID, за которыми следуют дата, время и остальная часть сообщения. Формат времени журнала для этой строки будет ”ppppp:yyyyMMdd:hhmmss”. Обратите внимание, что символы ”p” и ”:” являются заполнителями и могут быть любыми символами, кроме ”yMdhms”.
Populates host inventory field	Вы можете выбрать поле инвентаризации узла сети, которое будет заполняться значением элемента данных. Это будет работать, если для узла сети включено автоматическое заполнение инвентаризации, а также для элементов данных типов *Numeric(unsigned)*, *Numeric(float)*, *Character* и *Text*.
Description	Введите описание элемента данных. Поддерживаются пользовательские макросы.
Enabled	Установите флажок, чтобы включить элемент данных и он начал обрабатываться. При отключении элемента данных он немедленно удаляется из кэша истории (кроме его последнего значения, которое сохраняется для журналов).

Параметр	Описание
Latest data	Нажмите на ссылку, чтобы просмотреть последние данные для элемента данных. Эта 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. Вы можете воспользоваться ссылкой в заголовке формы, перейти на уровень шаблона и отредактировать их там, помня, что изменения на уровне шаблона изменят элемент данных для всех узлов сети, к которым привязан шаблон.

Вкладка **Tags** позволяет определить теги на уровне 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 сервера отстает, тестирование элемента данных может вернуть сообщение об ошибке "The task has been expired." При этом установка разных часовых поясов на сервере и прокси не повлияет на результат тестирования.

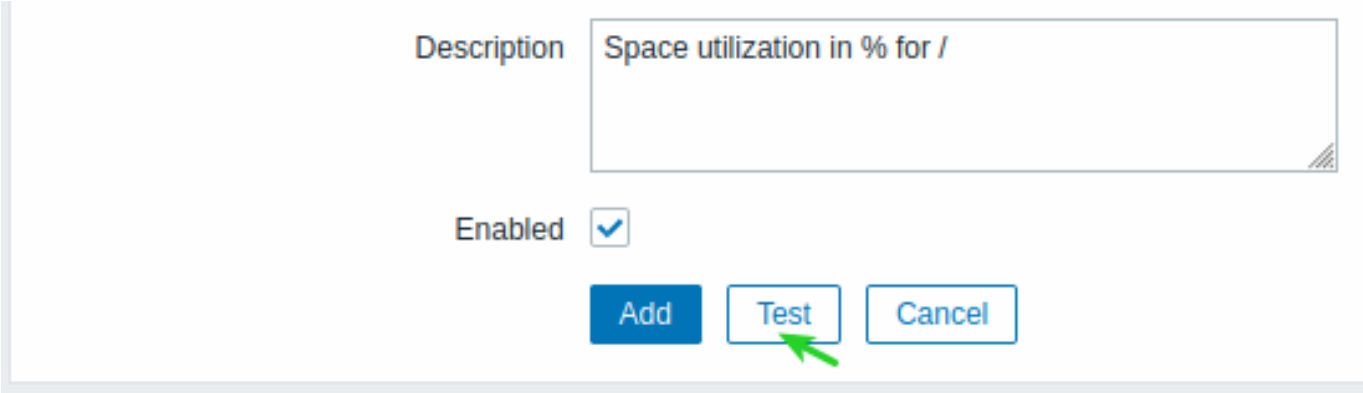
Можно протестировать элемент данных и, если он настроен правильно, получить в ответ реальное значение. Тестирование можно выполнять даже до сохранения элемента данных.

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 типов пассивных элементов данных:

- агент Zabbix
- SNMP-агент (v1, v2, v3)
- IPMI-агент
- SSH-проверки
- Telnet-проверки
- JMX-агент
- Простые проверки (кроме элементов данных `icmpping*`, `vmware.*`)
- Внутренние проверки Zabbix
- Вычисляемые элементы данных
- Внешние проверки
- Мониторинг баз данных
- HTTP-агент
- Скрипт
- Браузер

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 тестировать (например, активные проверки, исключенные простые проверки).

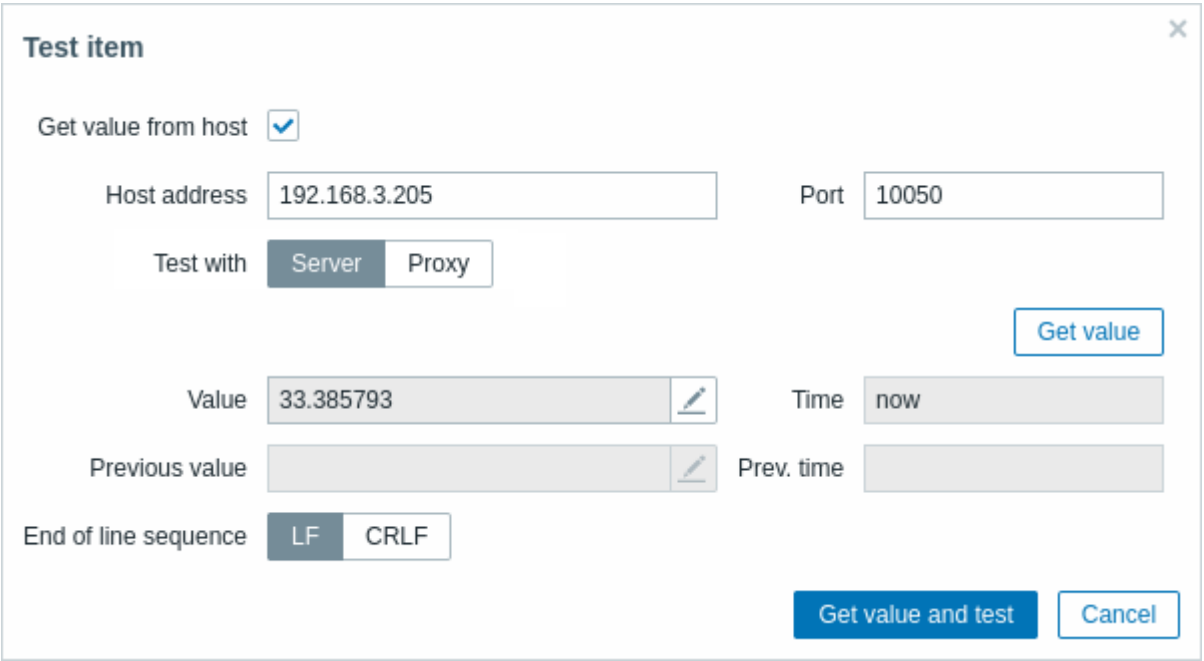


Форма тестирования элемента данных содержит поля для обязательных параметров узла сети (адрес узла сети, порт, тестирование через сервер/прокси (имя прокси)) и специфичных для элемента данных сведений (например, community для SNMPv2 или учетные данные безопасности SNMPv3). Эти поля зависят от контекста:

- Значения заполняются автоматически, когда это возможно, например, для элементов данных, которым требуется агент, — на основе информации из выбранного интерфейса агента узла сети.
- Для элементов данных шаблона значения необходимо заполнять вручную.
- Значения макросов в открытом виде подставляются автоматически.
- Поля, в которых значение (или часть значения) является секретным макросом или макросом Vault, остаются пустыми и должны быть заполнены 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)

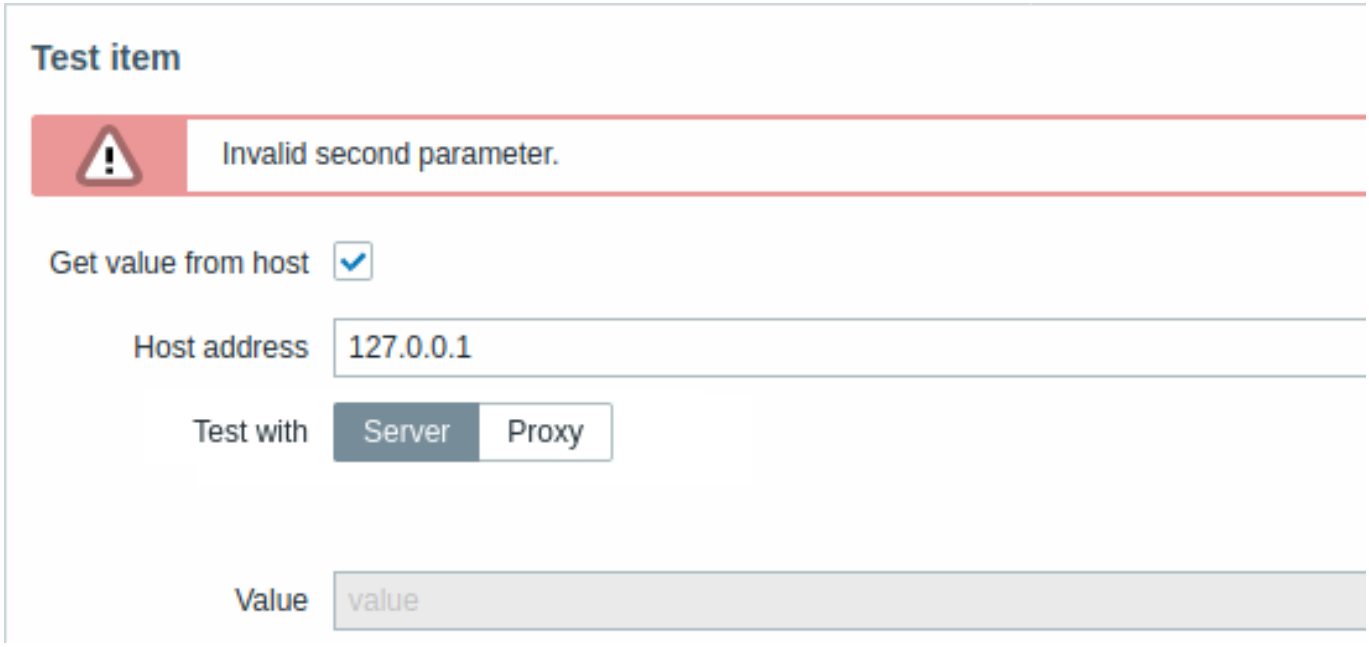
Чтобы протестировать элемент данных, нажмите *Get value*. Если значение успешно получено, оно будет помещено в поле *Value*, а текущее значение (если оно есть) будет перенесено в поле *Previous value*; при этом также будет вычислено поле *Prev. time*, то есть разница во времени между двумя значениями (нажатиями), а также будет предпринята попытка определить последовательность EOL и переключиться на CRLF при обнаружении ”\n\r” в полученном значении.

Нажмите *Get value and test*, чтобы протестировать предобработку.



Значения, полученные от узла сети, и результаты тестирования усекаются до максимального размера 512 КБ при отправке в веб-интерфейс. Если результат тестирования усечен, отображается значок предупреждения; наведите на него курсор, чтобы увидеть подробности. Если значение усечено, оно может не пройти проверку типа данных при последующих тестах, сравнивающих большие значения (например, JSON); в таких случаях отображается сообщение об ошибке. Обратите внимание, что данные размером более 512 КБ все равно полностью обрабатываются сервером Zabbix.

Если конфигурация неверна, отображается сообщение об ошибке с описанием возможной причины.



Успешно полученное от узла сети значение также можно использовать для тестирования шагов предобработки.

Кнопки диалога

Кнопки в нижней части диалога позволяют выполнить несколько видов операций.

Add (Добавить)	Добавление элемента данных. Эта кнопка доступна только для новых элементов данных.
Update (Обновить)	Обновление свойств элемента данных.
Clone (Клонировать)	Создание другого элемента данных на основе свойств текущего элемента данных.
Execute now (Выполнить сейчас)	Выполнение немедленной проверки нового значения элемента данных. Поддерживается только для **пассивных** проверок (смотрите более детальную информацию). *Обратите внимание*, что когда выполняется немедленная проверка значения, кэш конфигурации не обновляется, поэтому значение не будет отражать совсем недавние изменения настроек элемента данных.
Test (Тест)	Проверка корректности настройки элемента данных методом попытки получения значения.
Clear history and trends (Очистить историю и динамику изменений)	Удаление истории и динамики изменений элемента данных.
Delete (Удалить)	Удаление элемента данных.
Cancel (Отмена)	Отмена изменения свойств элемента данных.

Ограничения данных элемента данных

Ограничения данных элемента данных зависят от типа данных элемента данных и используемой СУБД.

Значения Numeric (unsigned) сохраняются как есть. Если получено значение с плавающей точкой, сохраняется только целая часть (например, 1.23 → 1).

Значения Numeric (float) сохраняются как есть и поддерживают приблизительно от 15 до 17 значащих цифр, в диапазоне

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 перед сохранением усекаются в соответствии с ограничением типа значения в базе данных:

СУБД	Тип информации		
	Символьный	**Журнал**	**Текст**
MySQL	255 символов	65536 байт	65536 байт
PostgreSQL	255 символов	65536 символов	65536 символов
SQLite (только Zabbix прокси)	255 символов	65536 символов	65536 символов

Двоичные значения ограничены 16 MiB (16777216 байт). Значения, превышающие это ограничение, отбрасываются, а соответствующая ошибка отображается в веб-интерфейс.

Значения с типом данных JSON ограничены 128 MiB (134217728 байт). Значения, превышающие это ограничение, отбрасываются, а соответствующая ошибка отображается в веб-интерфейс. Для MySQL/MariaDB рекомендуется настроить ежедневные партиции для таблицы `history_json` (используется для хранения значений JSON), так как она может быстро увеличиваться в размере и на длительное время блокировать housekeeper. Если ваш сценарий использования предполагает большие значения JSON (1 MiB и более), см. описание настройки системы для поддержки больших значений JSON.

Когда Zabbix записывает любые данные элемента данных в базу данных, он использует запросы INSERT и журналирует их (если `DebugLevel` установлен в 4 или 5). Большие записи журнала усекаются до 64 KB.

Ограничение пользовательского скрипта

Доступная длина пользовательского скрипта зависит от используемой базы данных:

База данных	Ограничение в символах	Ограничение в байтах
MySQL	65535	65535
PostgreSQL	65535	не ограничено
SQLite (только Zabbix прокси)	65535	не ограничено

Тайм-аут элемента данных

Тайм-аут элемента данных определяет, как долго Zabbix должен ждать перед прерыванием проверки с ошибкой.

Когда достигается тайм-аут, проверка прерывается, даже если получение данных еще не завершено. Если данные получены частично, элемент данных становится неподдерживаемым, а в журнал записывается сообщение об ошибке (например, если в SNMP-проверке данные успешно собраны только для одного из нескольких OID).

Для многих типов элементов данных можно настроить **гибкие** тайм-ауты:

- для отдельного элемента данных
- для типа элемента данных (на уровне прокси)
- для типа элемента данных (на глобальном уровне)

Пользовательский тайм-аут *для отдельного элемента данных* полезен, если вы хотите 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 отдельного элемента данных имеет приоритет над любыми другими тайм-аутами.
2. Тайм-ауты уровня прокси имеют приоритет над глобальными тайм-аутами.

Тайм-аут из конфигурации Zabbix сервера или прокси **не** играет никакой роли для проверок, для которых настроены гибкие тайм-ауты.

Поддержка гибкого тайм-аута

Гибкие тайм-ауты элементов данных поддерживаются для следующих типов элементов данных:

- Агент Zabbix (как для пассивных, так и для активных проверок)
- SNMP агент (кроме устаревших SNMP-проверок[1])
- Простая проверка (кроме `icmpping*` и элементов данных VMware[2])
- SSH агент
- Telnet агент
- Внешняя проверка

- Мониторинг базы данных
- Скрипт
- HTTP агент
- Браузер

[1] Для устаревших SNMP-проверок (опрос одного OID) применяются настройки тайм-аута из сервера или прокси.
 [2] Для элементов данных `icmpping*` значение тайм-аута указывается непосредственно в ключе элемента данных; также применяется жестко заданный максимально возможный тайм-аут в 600 секунд. Элементы данных мониторинга VMware используют собственный параметр VMwareTimeout из конфигурации сервера или прокси.

Неподдерживаемые элементы данных

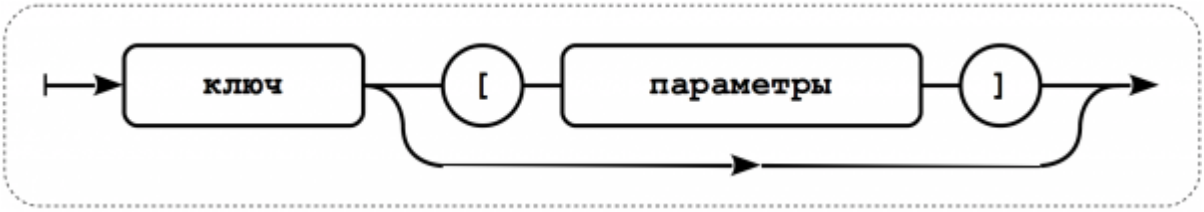
О неподдерживаемых элементах данных сообщается со статусом *Не поддерживается*, и они по-прежнему повторно проверяются с их стандартным *Интервалом обновления*.

Элемент данных становится неподдерживаемым, если по какой-либо причине не удается получить его значение (например, ошибка соединения, не настроены poller-процессы 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 при получении новых корректных исторических данных.

Как и проблемы, неподдерживаемые элементы данных переоцениваются только при получении новых данных — даже если для этого элемента данных исторические данные больше недоступны. Другими 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).

1. Формат ключа элемента данных

Следуйте правилам синтаксиса в формате ключа элемента данных, включая параметры ключа. На рисунках изображен поддерживаемый синтаксис. Допустимые элементы и символы в каждой позиции можно определить, следуя стрелкам — если до какого-то блока можно добраться по линии, то он разрешён, если нет — то не разрешён.



Для построения корректного ключа элемента данных нужно начать с указания имени ключа; затем есть выбор в зависимости от того, имеет ли ключ параметры или нет, — как проиллюстрировано двумя линиями, которым можно следовать.

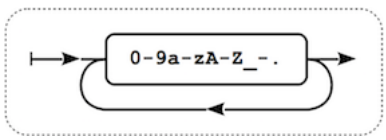
Имя ключа

Имя ключа имеет ограниченный диапазон разрешённых символов, которые просто следуют друг за другом. Разрешённые символы:

`0-9a-zA-Z_-.`

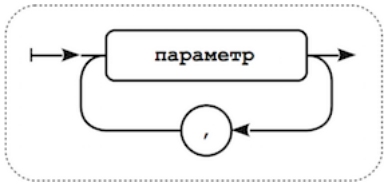
Что означает:

- все цифры;
- все буквы в нижнем регистре;
- все буквы в верхнем регистре;
- подчёркивание;
- дефис;
- точка.



Параметры ключа

Ключ элемента данных может принимать несколько параметров, которые должны быть разделены запятой.



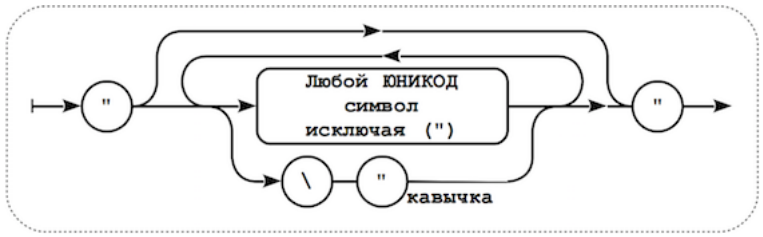
Каждый параметр ключа может быть одним из: заключённая в кавычки строка, строка без кавычек, массив.



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.

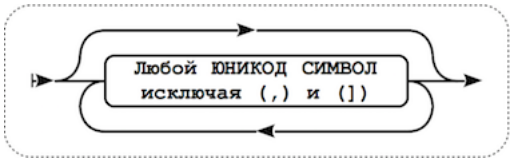


Warning:
Чтобы заключить параметры ключа элемента данных в кавычки, используйте только двойные кавычки. Одинарные кавычки не поддерживаются.

Attention:
Многоуровневые массивы параметров, например `[a,[b,[c,d]],e]`, не разрешены.

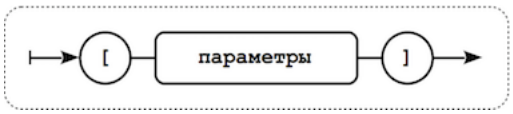
Параметр - строка, не заключённая в кавычки

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.



2. Пользовательские интервалы

Обзор

Можно создавать пользовательские правила, определяющие время, когда проверяется элемент данных. Для этого существуют два метода: *Гибкие интервалы*, которые позволяют переопределить интервал обновления по умолчанию, и *Планирование*, при котором проверка элемента данных может выполняться в определённое время или в последовательности моментов времени.

Гибкие интервалы

Гибкие интервалы позволяют переопределить интервал обновления по умолчанию для определённых периодов времени. Гибкий интервал задаётся с помощью *Интервала* и *Периода*, где:

- *Интервал* – интервал обновления для указанного периода времени. Поддерживаются суффиксы времени, например: 30s, 1m, 2h, 1d.
- *Период* – период времени, в течение которого активен гибкий интервал (подробное описание формата *Периода* см. в разделе периоды времени)

Если несколько гибких интервалов перекрываются, для периода перекрытия используется наименьшее значение *Интервала*. Обратите внимание, что если наименьшее значение перекрывающихся гибких интервалов равно '0', опрос выполняться не будет. Вне гибких интервалов используется интервал обновления по умолчанию.

Обратите внимание, что если гибкий интервал равен длине периода, элемент данных будет проверен ровно один раз. Если гибкий интервал больше периода, элемент данных может быть проверен один раз или не быть проверен вовсе (поэтому такая конфигурация не рекомендуется). Если гибкий интервал меньше периода, элемент данных будет проверен как минимум один раз.

Если для гибкого интервала установлено значение '0', элемент данных не опрашивается в течение периода гибкого интервала и возобновляет опрос в соответствии с *Интервалом обновления* по умолчанию после окончания периода. Примеры:

Interval	Period	Description
10	1-5,09:00-18:00	Элемент данных будет проверяться каждые 10 секунд в рабочее время.
0	1-7,00:00-7:00	Элемент данных не будет проверяться в ночное время.
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 в 12:00 каждый день. Обратите внимание, что это использовалось как обходной способ для проверок по расписанию, и для таких проверок рекомендуется использовать интервалы планирования.

Интервалы планирования

Интервалы планирования используются для проверки элементов данных в определённое время. В то время как гибкие интервалы предназначены для переопределения интервала обновления элемента данных по умолчанию, интервалы планирования используются для указания независимого расписания проверок, которое выполняется параллельно.

Интервал планирования задаётся следующим образом: `md<filter>wd<filter>h<filter>m<filter>s<filter>`, где:

- **md** — дни месяца
- **wd** — дни недели
- **h** — часы
- **m** — минуты
- **s** — секунды

`<filter>` используется для указания значений для своего префикса (дни, часы, минуты, секунды) и определяется следующим образом: `[<from>[-<to>]][/<step>][,<filter>]`, где:

- `<from>` и `<to>` определяют диапазон совпадающих значений (включительно). Если `<to>` опущено, тогда фильтр соответствует диапазону `<from> - <from>`. Если `<from>` также опущено, тогда фильтр соответствует всем возможным значениям.
- `<step>` определяет шаг пропуска числовых значений в пределах диапазона. По умолчанию `<step>` имеет значение 1, что означает, что совпадают все значения заданного диапазона.

Хотя определения фильтров являются необязательными, должен использоваться как минимум один фильтр. Фильтр должен содержать либо диапазон, либо заданное значение *<step>*.

Пустой фильтр соответствует либо ’0’, если не определён фильтр более низкого уровня, либо всем возможным значениям в противном случае. Например, если фильтр часов опущен, тогда будет соответствовать только час ’0’, при условии, что фильтры минут и секунд также опущены; в противном случае пустой фильтр часов будет соответствовать всем значениям часов.

Допустимые значения `<from>` и `<to>` для соответствующих префиксов фильтра:

Prefix	Description	*<from>*	*<to>*
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 `<from>` должно быть меньше или равно значению `<to>`. Значение `<step>` должно быть больше или равно 1 и меньше или равно `<to>` - `<from>`.

Однозначные значения дней месяца, часов, минут и секунд могут иметь префикс 0. Например, `md01-31` и `h/02` являются допустимыми интервалами, а `md01-031` и `wd01-07` — нет.

В веб-интерфейс Zabbix несколько интервалов планирования вводятся в отдельных строках. В Zabbix API они объединяются в одну строку с точкой с запятой `;` в качестве разделителя.

Если одному времени соответствуют несколько интервалов, оно выполняется только один раз. Например, `wd1h9;h9` будет выполнен только один раз в понедельник в 9:00.

Примеры:

Interval	Will be executed
m0-59	каждую минуту
h9-17/2	каждые 2 часа, начиная с 9:00 (9:00, 11:00 ...)
m0,30 or m/30	каждый час в hh:00 и hh:30
m0,5,10,15,20,25,30,35,40,45,50,55 or m/5	каждые пять минут
wd1-5h9	каждый понедельник-пятницу в 9:00
wd1-5h9-18	каждый понедельник-пятницу в 9:00,10:00,...,18:00
h9,10,11 or h9-11	каждый день в 9:00, 10:00 и 11:00
md1h9m30	каждый 1-й день месяца в 9:30
md1wd1h9m30	каждый 1-й день месяца в 9:30, если это 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 (*синтаксис только для API*)	каждый день в 9:00, 9:30, 10:00
h9m/30 h10 (*добавьте это как ещё одну строку в веб-интерфейс*)	каждый день в 9:00, 9:30, 10:00

Выравнивание часовых поясов для прокси и агента

Обратите внимание, что прокси Zabbix и агент используют свои локальные часовые пояса при обработке интервалов планирования.

По этой причине, когда интервалы планирования применяются к элементам данных, которые мониторятся через прокси Zabbix, или к активным элементам данных агента, рекомендуется установить часовой пояс соответствующих прокси или агента таким же, как у сервера Zabbix, иначе очередь может некорректно сообщать о задержках элементов данных.

Часовой пояс для прокси Zabbix или агента можно задать с помощью переменной окружения TZ в файле юнита systemd:

```
[Service]
...
Environment="TZ=Europe/Amsterdam"
```

2. Предобработка значений элементов данных

Обзор

Предобработка позволяет применять преобразования к полученным значениям элемента данных перед их сохранением в базе данных. Эти преобразования/шаги предобработки выполняются сервером Zabbix или прокси (если элементы данных отслеживаются прокси).

Эта возможность поддерживает различные сценарии использования, например:

- преобразование байтов в биты (например, умножение значений сетевого трафика на ”8”);
- вычисление статистики в секунду для значений, которые увеличиваются нарастающим итогом;
- применение регулярных выражений для извлечения или изменения значений;
- выполнение пользовательских скриптов над значениями;
- отбрасывание неизменившихся значений для оптимизации хранения в базе данных.

Для элемента данных можно настроить один или несколько шагов предобработки. Эти шаги выполняются в том порядке, в котором они настроены.

Attention:
Если шаг предобработки завершается ошибкой, элемент данных становится неподдерживаемым. Этого можно избежать с помощью обработки ошибок *Custom on fail* (доступна для большинства преобразований), которая позволяет отбрасывать значения или задавать пользовательские значения.

 Для элементов данных журнала метаданные журнала (без значения) всегда сбрасывают состояние неподдерживаемости элемента данных, снова делая его поддерживаемым. Это происходит, даже если исходная ошибка возникла после получения значения журнала от агент.

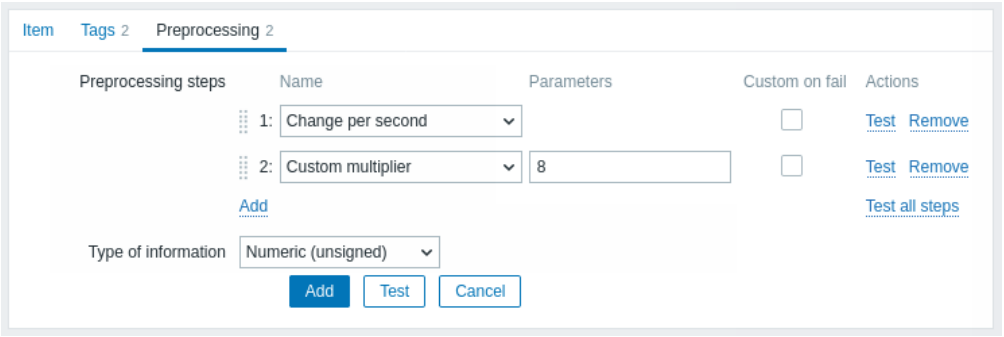
Все значения, передаваемые в предобработку, изначально рассматриваются как строки. Преобразование в требуемый тип значения (как определено в конфигурации элемента данных) выполняется в конце конвейера предобработки. Однако отдельные шаги предобработки могут вызывать более ранние преобразования, если это необходимо. Подробную техническую информацию см. в разделе Подробности предобработки.

Чтобы убедиться, что ваша конфигурация предобработки работает ожидаемым образом, вы можете протестировать её.

См. также: Примеры предобработки

Конфигурация

Шаги предобработки определяются на вкладке **Предобработка** в форме конфигурации элемента данных.



Нажмите *Добавить*, чтобы выбрать поддерживаемое преобразование.

Поле *Тип информации* отображается в нижней части вкладки, когда определен хотя бы один шаг предобработки. При необходимости можно изменить тип информации, не покидая вкладку *Предобработка*. Подробное описание параметров см. в разделе Создание элемента данных.

Поддерживаемые преобразования

Ниже перечислены все поддерживаемые преобразования. Нажмите на название преобразования, чтобы увидеть полную информацию о нём.

Название	Описание	Тип
Регулярное выражение	Сопоставить значение с регулярным выражением и заменить его требуемым результатом.	Текст
Замена	Найти строку поиска и заменить её другой строкой (или удалить).	
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 XML-данных с помощью функциональности XPath.	Структурированные данные
JSON Path	Извлечь значение или фрагмент из JSON-данных 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.	
Значение SNMP get	Применить параметры форматирования к значению SNMP get.	
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 в секунду (разницу между текущим и предыдущим значением).	
Логическое 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	Введите 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 значение не должно соответствовать.	
Проверка ошибки в JSON	Проверить наличие сообщения об ошибке уровня приложения, расположенного по JSONPath.	
Проверка ошибки в XML	Проверить наличие сообщения об ошибке уровня приложения, расположенного по XPath.	
Проверка ошибки с помощью регулярного выражения	Проверить наличие сообщения об ошибке уровня приложения с помощью регулярного выражения.	
Проверка неподдерживаемого значения	Проверить, не удалось ли получить значение элемента данных.	
Отбрасывать неизменённое	Отбросить значение, если оно не изменилось.	Ограничение частоты
Отбрасывать неизменённое с контрольным интервалом	Отбросить значение, если оно не изменилось в течение заданного периода времени.	
Шаблон Prometheus	Использовать следующий запрос для извлечения требуемых данных из метрик Prometheus.	Prometheus
Prometheus в JSON	Преобразовать требуемые метрики Prometheus в JSON.	

Обратите внимание, что для шагов предобработки *Изменение* и *Ограничение частоты* Zabbix должен запоминать

последнее значение, чтобы вычислять/сравнивать новое значение требуемым образом. Эти предыдущие значения обрабатываются менеджером предобработки. Если сервер Zabbix или прокси перезапускается либо в шаги предобработки вносятся какие-либо изменения, последнее значение соответствующего элемента данных сбрасывается, в результате чего:

- для шагов *Простое изменение*, *Изменение в секунду* — следующее значение будет проигнорировано, поскольку отсутствует предыдущее значение, относительно которого можно вычислить изменение;
- для шагов *Отбрасывать неизменённое*, *Отбрасывать неизменённое с контрольным интервалом* — следующее значение никогда не будет отброшено, даже если по правилам отбрасывания оно должно было быть отброшено.

Регулярное выражение

Сопоставьте значение с регулярным выражением и замените его требуемым выводом.

Параметры:

- **pattern** — регулярное выражение;
- **output** — шаблон форматирования вывода. Экранирующая последовательность \N (где N=1…9) заменяется N-й совпавшей группой. Экранирующая последовательность \0 заменяется совпавшим текстом.

Комментарии:

- Если входное значение не совпадает, элемент данных станет неподдерживаемым;
- Регулярное выражение поддерживает извлечение максимум 10 захваченных групп с помощью последовательности \N;
- Если установить флажок *Custom on fail*, можно указать пользовательские параметры обработки ошибок: либо отбросить значение, либо установить указанное значение, либо установить указанное сообщение об ошибке. В случае сбоя на этапе предобработки элемент данных не станет неподдерживаемым, если выбрана опция отбросить значение или установить указанное значение.
- Некоторые существующие примеры приведены в разделе regular expressions.

Заменить

Найдите строку поиска и замените её другой строкой (или ничем).

Параметры:

- **строка поиска** — строка, которую нужно найти и заменить, с учётом регистра (обязательно);
- **замена** — строка, на которую будет заменена строка поиска. Строка замены также может быть пустой, что фактически позволяет удалить найденную строку поиска.

Комментарии:

- Все вхождения строки поиска будут заменены;
- Можно использовать escape-последовательности для поиска или замены переводов строк, возврата каретки, табуляции и пробелов ”\n \r \t \s”; обратную косую черту можно экранировать как ”\\” , а escape-последовательности можно экранировать как ”\\n”;
- Экранирование переводов строк, возврата каретки и табуляции выполняется автоматически во время обнаружения низкого уровня.

Обрезать

Удалить указанные символы с начала и конца значения.

Обрезать справа

Удалить указанные символы с конца значения.

Обрезать слева

Удалить указанные символы с начала значения.

XML XPath

Извлечение значения или фрагмента из XML-данных с использованием функциональности XPath.

Комментарии:

- Чтобы эта опция работала, сервер Zabbix (или прокси Zabbix) должен быть скомпилирован с поддержкой libxml;
- Пространства имён не поддерживаются;
- Если вы установите флажок *Пользовательское поведение при ошибке*, можно указать пользовательские параметры обработки ошибок: либо отбросить значение, установить указанное значение или установить указанное сообщение об ошибке. В случае сбоя шага предобработки элемент данных не станет неподдерживаемым, если выбрана опция отбросить значение или установить указанное значение.

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.

Если установить флажок *Пользовательское действие при ошибке*, можно указать пользовательские варианты обработки ошибок: либо отбросить значение, установить указанное значение или задать указанное сообщение об ошибке. В случае сбоя шага предобработки элемент данных не станет неподдерживаемым, если выбрана опция отбросить значение или установить указанное значение.

CSV в JSON

Сконвертировать данные файла CSV в формат JSON.

Для получения дополнительной информации смотрите: Преобразование CSV в JSON.

XML в JSON

Преобразование данных в формате XML в JSON.

Для получения дополнительной информации см.: Правила сериализации.

Если установить флажок *Пользовательское действие при ошибке*, можно указать пользовательские варианты обработки ошибок: либо отбросить значение, установить указанное значение или установить указанное сообщение об ошибке. В случае сбоя на этапе предобработки элемент данных не станет неподдерживаемым, если выбрана опция отбросить значение или установить указанное значение.

Значение SNMP walk

Извлеките значение по указанному имени OID/MIB и примените параметры форматирования:

- **Без изменений** — вернуть hex-строку как неэкранированную hex-строку (*обратите внимание*, что подсказки отображения всё равно 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, выраженной как последовательность шестнадцатеричных символов (например, ”1A 2B 3C 4D”), в 64-битное беззнаковое целое число. В битовых строках длиной более 8 байт последующие байты будут проигнорированы.

Если вы установите флажок *Пользовательское при ошибке*, можно указать пользовательские параметры обработки ошибок: либо отбросить значение, установить указанное значение или установить указанное сообщение об ошибке. В случае сбоя на этапе предобработки элемент данных не станет неподдерживаемым, если выбрана опция отбросить значение или установить указанное значение.

Преобразование SNMP walk в JSON

Преобразует значения SNMP в JSON.

Укажите имя поля в JSON и соответствующий путь SNMP OID. Значения полей будут заполняться значениями из указанного пути SNMP OID.

Комментарии:

- Доступны параметры форматирования значений, аналогичные шагу *Значение SNMP walk*;
- Вы можете использовать этот шаг предобработки для обнаружения SNMP OID;
- Если установить флажок *Пользовательское действие при ошибке*, можно указать пользовательские параметры обработки ошибок: либо отбросить значение, установить указанное значение или установить указанное сообщение об ошибке. В случае сбоя шага предобработки элемент данных не станет неподдерживаемым, если выбрана опция отбросить значение или установить указанное значение.

Получение значения SNMP

Примените параметры форматирования к значению, полученному через SNMP:

- **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 как последовательность шестнадцатеричных символов (например, ”1A 2B 3C 4D”), в 64-битное целое число без знака. В битовых строках длиной более 8 байт последующие байты будут проигнорированы.

Если установить флажок *Пользовательское при ошибке*, можно указать пользовательские параметры обработки ошибок: либо отбросить значение, установить указанное значение или установить указанное сообщение об ошибке. В случае сбоя на этапе предобработки элемент данных не станет неподдерживаемым, если выбрана опция отбросить значение или установить указанное значение.

Пользовательский множитель

Умножьте значение на указанный целочисленный или вещественный множитель.

Комментарии:

- Используйте эту опцию, чтобы преобразовывать значения, полученные в KB, MBps и т. д., в B, Bps. В противном случае Zabbix не сможет корректно устанавливать префиксы (K, M, G и т. д.).
- Обратите внимание, что если тип информации элемента данных — *Numeric (unsigned)*, входящие значения с дробной частью будут усечены (то есть ’0.9’ станет ’0’) до применения пользовательского множителя;
- Если вы используете пользовательский множитель или сохраняете значение как *Change per second* для элементов данных с типом информации *Numeric (unsigned)*, и итоговое вычисленное значение фактически является числом с плавающей точкой, это вычисленное значение всё равно будет принято как корректное: дробная часть будет отброшена, а значение сохранено как целое число;
- Поддерживается: научная нотация, например 1e+70; пользовательские макросы и LLD-макросы; строки, включающие макросы, например `{#MACRO}e+10`, `{$MACRO1}e+{$MACRO2}`. Макросы должны разрешаться в целое число или число с плавающей точкой.
- Если вы установите флажок *Custom on fail*, можно указать пользовательские параметры обработки ошибок: либо отбросить значение, установить указанное значение или установить указанное сообщение об ошибке. В случае сбоя шага предобработки элемент данных не станет неподдерживаемым, если выбрана опция отброса значения или установки указанного значения.

Простое изменение

Вычисляет разницу между текущим и предыдущим значением.

Комментарии:

- Этот шаг может быть полезен для измерения постоянно растущего значения;
- Вычисляется как **value**-**prev_value**, где *value* — текущее значение; *prev_value* — ранее полученное значение;
- Для одного элемента данных допускается только одна операция изменения (”Simple change” или ”Change per second”);
- Если текущее значение меньше предыдущего, Zabbix отбрасывает эту разницу (ничего не сохраняет) и ожидает другое значение;
- Если вы установите флажок *Custom on fail*, можно указать пользовательские варианты обработки ошибок: либо отбросить значение, либо установить указанное значение, либо установить указанное сообщение об ошибке. В случае сбоя шага предобработки элемент данных не станет неподдерживаемым, если выбрана опция отбросить значение или установить указанное значение.

Изменение в секунду

Вычисляет скорость изменения значения (разницу между текущим и предыдущим значением) в секунду.

Комментарии:

- Этот шаг полезен для вычисления скорости в секунду для постоянно растущего значения;
- Поскольку результатом этого вычисления могут быть числа с плавающей точкой, рекомендуется установить для параметра ’Type of information’ значение *Numeric (float)*, даже если входящие необработанные значения являются целыми числами. Это особенно важно для небольших чисел, где имеет значение дробная часть. Если значения с плавающей точкой велики и могут превысить длину поля ’float’, в результате чего всё значение может быть потеряно, фактически рекомендуется использовать *Numeric (unsigned)* и тем самым отсечь только дробную 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;
- Допускается только одна операция изменения на элемент данных (”Simple change” или ”Change per second”);
- Если текущее значение меньше предыдущего, Zabbix отбрасывает эту разницу (ничего не сохраняет) и ожидает следующее значение. Это помогает корректно работать, например, с переполнением (циклическим возвратом) 32-битных SNMP-счётчиков.
- Если установить флажок *Custom on fail*, можно указать пользовательские варианты обработки ошибок: либо отбросить значение, либо установить указанное значение, либо установить указанное сообщение об ошибке. В

случае сбоя на шаге предобработки элемент данных не станет неподдерживаемым, если выбрана опция отбросить значение или установить указанное значение.

Логическое значение в десятичное

Преобразует значение из логического формата в десятичный.

Комментарии:

- Текстовое представление преобразуется либо в 0, либо в 1. Таким образом, 'TRUE' сохраняется как 1, а 'FALSE' сохраняется 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 TRUE, а ноль считается FALSE.
- Если вы установите флажок *Custom on fail*, можно указать пользовательские параметры обработки ошибок: либо отбросить значение, установить указанное значение или установить указанное сообщение об ошибке. В случае сбоя шага предобработки элемент данных не станет неподдерживаемым, если выбрана опция отбросить значение или установить указанное значение.

Из восьмеричного в десятичный

Преобразует значение из восьмеричного формата в десятичный.

Если установить флажок *Пользовательское действие при ошибке*, можно указать пользовательские варианты обработки ошибок: либо отбросить значение, установить указанное значение или задать указанное сообщение об ошибке. В случае сбоя на этапе предобработки элемент данных не станет неподдерживаемым, если выбрана опция отбросить значение или установить указанное значение.

Шестнадцатеричное в десятичное

Преобразует значение из шестнадцатеричного формата в десятичный.

Если установить флажок *Пользовательское при ошибке*, можно указать пользовательские варианты обработки ошибок: либо отбросить значение, либо установить указанное значение, либо задать указанное сообщение об ошибке. В случае сбоя на этапе предобработки элемент данных не станет неподдерживаемым, если выбрана опция отброса значения или установки указанного значения.

JavaScript

Введите код JavaScript в модальном редакторе, который открывается при щелчке в поле параметра или по значку карандаша рядом с ним.

> **Attention:**
> Не используйте необъявленные присваивания в JavaScript предобработки. Используйте `var` для объявления локальных переменных.

Комментарии:

- Доступная длина JavaScript зависит от используемой базы данных;
- Дополнительную информацию см. в разделе: Предобработка Javascript.

В диапазоне

Определите диапазон, в котором должно находиться значение, указав минимальное/максимальное значения (включительно).

Комментарии:

- Допускаются числовые значения (включая любое количество цифр, необязательную десятичную часть и необязательную экспоненциальную часть, отрицательные значения);
- Минимальное значение должно быть меньше максимального;
- Должно быть указано как минимум одно значение;
- Можно использовать пользовательские макросы и макросы низкоуровневого обнаружения;
- Если установить флажок *Пользовательское действие при ошибке*, можно указать пользовательские параметры обработки ошибок: либо отбросить значение, установить указанное значение или установить указанное сообщение об ошибке. В случае сбоя на этапе предобработки элемент данных не станет неподдерживаемым, если выбрана опция отбросить значение или установить указанное значение.

Соответствует регулярному выражению

Укажите регулярное выражение, которому должно соответствовать значение.

Если вы установите флажок *Пользовательское действие при ошибке*, можно будет указать пользовательские варианты обработки ошибок: либо отбросить значение, установить указанное значение или задать указанное сообщение об ошибке.

В случае сбоя на шаге предобработки элемент данных не станет неподдерживаемым, если выбрана опция отбросить значение или установить указанное значение.

Не соответствует регулярному выражению

Укажите регулярное выражение, которому значение не должно соответствовать.

Если установить флажок *Пользовательское действие при ошибке*, можно указать пользовательские варианты обработки ошибок: либо отбросить значение, установить указанное значение или задать указанное сообщение об ошибке. В случае сбоя на шаге предобработки элемент данных не станет неподдерживаемым, если выбрана опция отбросить значение или установить указанное значение.

Проверка ошибки в JSON

Проверяет наличие сообщения об ошибке на уровне приложения, расположенного по JSONPath. Останавливает обработку, если проверка выполнена успешно и сообщение не пустое; в противном случае обработка продолжается со значением, которое было до этого шага предобработки.

Комментарии:

- Эти ошибки внешнего сервиса сообщаются пользователю как есть, без добавления информации о шаге предобработки;
- Ошибка не будет сообщена в случае неудачной попытки разобрать некорректный JSON;
- Если установить флажок *Custom on fail*, можно указать пользовательские варианты обработки ошибки: либо отбросить значение, установить указанное значение или задать указанное сообщение об ошибке. В случае сбоя шага предобработки элемент данных не станет неподдерживаемым, если выбрана опция отбросить значение или установить указанное значение.

Проверка ошибки в XML

Проверяет наличие сообщения об ошибке на уровне приложения, расположенного по XPath. Останавливает обработку, если проверка выполнена успешно и сообщение не пустое; в противном случае обработка продолжается со значением, которое было до этого шага предобработки.

Комментарии:

- Эти ошибки внешнего сервиса сообщаются пользователю как есть, без добавления информации о шаге предобработки;
- Ошибка не будет сообщена в случае неудачной попытки разобрать некорректный XML;
- Если установить флажок *Custom on fail*, можно указать пользовательские варианты обработки ошибки: либо отбросить значение, установить указанное значение или установить указанное сообщение об ошибке. В случае сбоя шага предобработки элемент данных не станет неподдерживаемым, если выбрана опция отбросить значение или установить указанное значение.

Проверка ошибки с помощью регулярного выражения

Проверьте сообщение об ошибке на уровне приложения с помощью регулярного выражения. Остановите обработку, если проверка выполнена успешно и сообщение не пустое; в противном случае продолжайте обработку со значением, которое было до этого шага предобработки.

Параметры:

- **pattern** — регулярное выражение;
- **output** — шаблон форматирования вывода. Управляющая последовательность \N (где N=1…9) заменяется на N-ю совпавшую группу. Управляющая последовательность \0 заменяется на совпавший текст.

Комментарии:

- Эти ошибки внешнего сервиса сообщаются пользователю как есть, без добавления информации о шаге предобработки;
- Если вы установите флажок *Custom on fail*, можно указать пользовательские варианты обработки ошибок: либо отбросить значение, установить указанное значение или установить указанное сообщение об ошибке. В случае неудачного шага предобработки элемент данных не станет неподдерживаемым, если выбрана опция отбросить значение или установить указанное значение.

Проверка неподдерживаемого значения

Проверьте, что не удалось получить ни одно значение элемента данных. Укажите, как следует обрабатывать сбой, на основе анализа возвращённого сообщения об ошибке.

Параметры:

- **scope** — выберите область обработки ошибки:
any error — любая ошибка;
error matches — только ошибка, соответствующая регулярному выражению, указанному в *pattern*;
error does not match — только ошибка, не соответствующая регулярному выражению, указанному в *pattern*
- **pattern** — регулярное выражение для сопоставления с ошибкой. Если в параметре scope выбрано *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 предобработки позволяет изменить такое поведение. Если установить флажок *Custom on fail* (для этого шага предобработки он всегда установлен и недоступен для изменения), можно указать пользовательские варианты обработки ошибки: отбросить значение, установить указанное значение или установить указанное сообщение об ошибке. В случае сбоя на шаге предобработки элемент данных не станет неподдерживаемым, если выбрана опция отбросить значение или установить указанное значение.
- Этот шаг предобработки проверяет только то, что не удалось получить значение элемента данных. Например, он не проверяет, соответствует ли тип полученного значения (например, строка) типу информации элемента данных (например, числовой); подробности см. в Примерах предобработки. Если имеется несоответствие типов, элемент данных всё равно может стать неподдерживаемым после выполнения всех шагов предобработки. Чтобы проверить несоответствие типов, можно использовать, например, шаг предобработки *Custom multiplier*; см. Примеры предобработки.
- Поддерживается захват групп регулярного выражения в полях *Set value to* или *Set error to*. Используйте \N (где N=1…9) для получения N-й совпавшей группы; используйте \0 для получения совпавшего текста;
- Эти шаги всегда выполняются первыми среди шагов предобработки и после сохранения изменений элемента данных размещаются выше всех остальных;
- Поддерживается несколько шагов *Check for not supported value* в указанном порядке. Шаг для *any error* будет автоматически помещён последним в этой группе.

Отбрасывать неизменившиеся

Отбрасывать значение, если оно не изменилось.

Комментарии:

- Если значение отброшено, оно не сохраняется в базе данных, и сервер Zabbix не знает, что это значение было получено. Выражения триггеров не будут вычисляться, в результате чего для связанных триггеров не будут создаваться/закрываться проблемы. Функции будут работать только на основе данных, которые фактически сохранены в базе данных. Поскольку тренды строятся на основе данных в базе данных, если в течение часа не было сохранено ни одного значения, то и данных трендов за этот час также не будет.
- Для одного элемента данных можно указать только один вариант ограничения частоты.

Отбрасывать неизменившееся с контрольным интервалом

Отбрасывать значение, если оно не изменялось в течение заданного периода времени (в секундах).

Комментарии:

- Поддерживаются положительные целые значения для указания секунд (минимум — 1 секунда);
- Можно использовать суффиксы времени (например, 30s, 1m, 2h, 1d);
- Можно использовать пользовательские макросы и макросы низкоуровневого обнаружения;
- Если значение отброшено, оно не сохраняется в базе данных, и сервер Zabbix не знает, что это значение было получено. Выражения триггеров не будут вычисляться, в результате чего для связанных триггеров не будут создаваться/закрываться проблемы. Функции будут работать только на основе данных, фактически сохранённых в базе данных. Поскольку тренды строятся на основе данных в базе данных, если в течение часа не было сохранено ни одного значения, то и данных трендов за этот час также не будет.
- Для одного элемента данных можно указать только один вариант ограничения частоты.

Шаблон Prometheus

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 шага предобработки, включая код JavaScript;
- пользовательских параметрах обработки ошибок (поля *Установить значение в* и *Установить ошибку в*).

Note:
Контекст макроса игнорируется, когда макрос заменяется своим значением. Значение макроса вставляется в код как есть, невозможно добавить дополнительное экранирование перед помещением значения в код JavaScript. Обратите внимание, что в некоторых случаях это может вызывать ошибки JavaScript.

Тестирование

Смотрите тестирование предварительной обработки.

1. Тестирование предварительной обработки

Тестирование

Тестирование этапов предварительной обработки полезно, чтобы уже на этапе настройки процесса предварительной обработки убедиться, что цепочка шагов предварительной обработки дает ожидаемые результаты, не дожидаясь получения значения элемента данных.

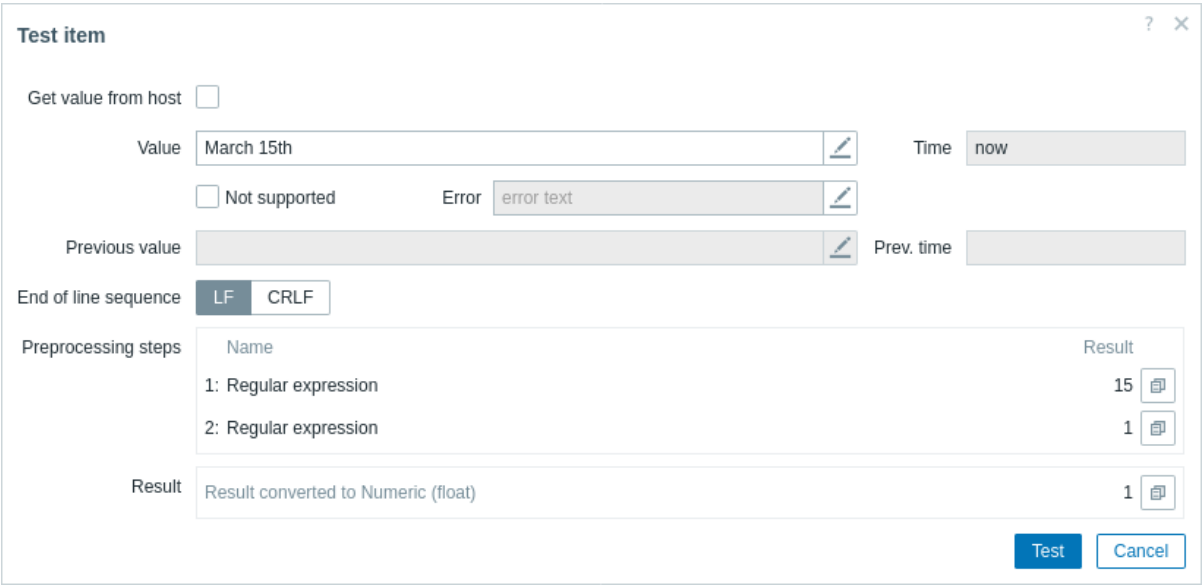


Предварительную обработку можно протестировать:

- используя гипотетическое значение
- используя реальное значение, полученное от узла сети

Каждый шаг предобработки можно протестировать отдельно, также можно проверить все этапы одновременно. Когда вы нажимаете кнопку *Тест* или *Тест всех шагов* соответственно в блоке Действия, открывается окно тестирования.

Тестирование гипотетического значения



Параметр	Описание
Получить значение с узла сети	Если вы хотите протестировать гипотетическое значение, оставьте этот флажок снятым. См. также: Тестирование реального значения.
Значение	Введите входное значение для тестирования. Щелчок в поле параметра или по кнопке просмотра/редактирования откроет окно текстовой области для ввода значения или блока кода.

Параметр	Описание
Не поддерживается	Установите этот флажок, чтобы протестировать неподдерживаемое значение. Этот параметр полезен для тестирования шага предобработки *Проверка на неподдерживаемое значение*.
Ошибка	Введите текст ошибки. Это поле доступно, когда флажок *Получить значение с узла сети* снят, а флажок *Не поддерживается* установлен. Если флажок *Получить значение с узла сети* установлен, это поле заполняется фактическим сообщением об ошибке (только для чтения) от узла сети.
Время	Отображается время входного значения: `now` (только для чтения).
Предыдущее значение	Введите предыдущее входное значение для сравнения. Только для шагов предобработки *Изменение* и *Троттлинг*.
Предыдущее время	Введите время предыдущего входного значения для сравнения. Только для шагов предобработки *Изменение* и *Троттлинг*. Значение по умолчанию основано на значении поля 'Интервал обновления' элемента данных (если '1m', то это поле заполняется значением `now-1m`). Если ничего не указано или у пользователя нет доступа к узлу сети, по умолчанию используется `now-30s`.
Макросы	Если используются какие-либо макросы, они перечисляются вместе со своими значениями. Значения можно редактировать для целей тестирования, но изменения будут сохранены только в контексте тестирования.
Последовательность конца строки	Выберите последовательность конца строки для многострочных входных значений: **LF** — последовательность LF (перевод строки) **CRLF** — последовательность CRLF (возврат каретки + перевод строки).
Шаги предобработки	Перечислены шаги предобработки; результат тестирования отображается для каждого шага после нажатия кнопки *Тест*. Результаты тестирования усекаются до максимального размера 512KB при отправке в веб-интерфейс. Результаты тестирования можно копировать (не более усечённых 512KB). Если результат усечён, отображается значок предупреждения. Описание предупреждения отображается при наведении указателя мыши. Обратите внимание, что данные размером более 512KB всё равно полностью обрабатываются сервером Zabbix. Если шаг завершился ошибкой при тестировании, отображается значок ошибки. Описание ошибки отображается при наведении указателя мыши. Если для шага указано "Пользовательское действие при ошибке" и это действие было выполнено, сразу после строки шага тестирования предобработки появляется новая строка, показывающая, какое действие было выполнено и к какому результату оно привело (ошибка или значение).
Результат	Итоговый результат тестирования шагов предобработки отображается во всех случаях, когда все шаги тестируются вместе (при нажатии кнопки *Тестировать все шаги*). Также отображается тип преобразования к типу значения элемента данных, например `Result converted to Numeric (unsigned)`. Результаты тестирования усекаются до максимального размера 512KB при отправке в веб-интерфейс. Итоговый результат можно копировать (не более усечённых 512KB). Если результат усечён, отображается значок предупреждения. Описание предупреждения отображается при наведении указателя мыши. Обратите внимание, что данные размером более 512KB всё равно полностью обрабатываются сервером Zabbix.

Нажмите *Тест*, чтобы увидеть результат после каждого шага предобработки.

Тестовые значения сохраняются между сеансами тестирования как для отдельных шагов, так и для всех шагов, что позволяет пользователю изменять шаги предобработки или конфигурацию элемента данных, а затем возвращаться в окно тестирования без необходимости повторного ввода информации. Однако при обновлении страницы значения теряются.

Тестирование выполняется сервером Zabbix. Веб-интерфейс отправляет соответствующий запрос на сервер и ожидает результат. Запрос содержит входное значение и шаги предобработки (с раскрытыми пользовательскими макросами). Для шагов *Изменение* и *Троттлинг* можно дополнительно указать предыдущее значение и время. Сервер возвращает результаты для каждого шага предобработки.

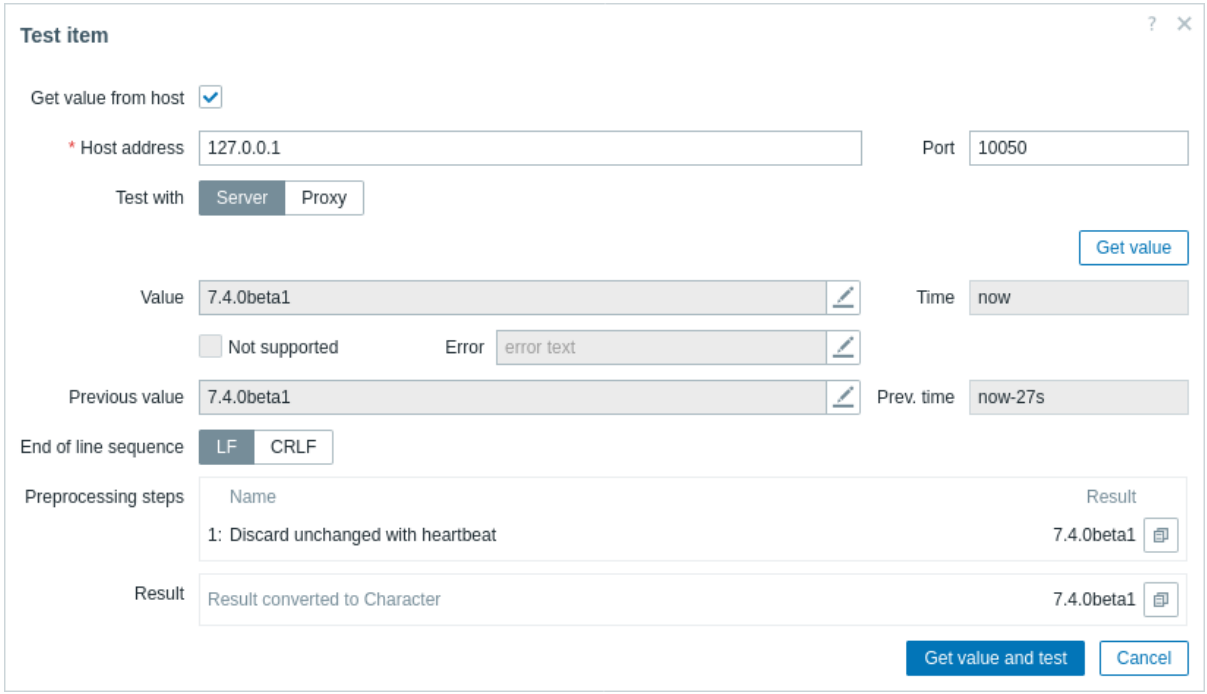
Все технические ошибки или ошибки проверки входных данных отображаются в блоке ошибок в верхней части окна тестирования.

Тестирование реального значения

Чтобы проверить предварительную обработку на реальном значении:

- Отметьте опцию *Получить значение с узла сети*

- Введите или проверьте параметры узла сети (адрес узла сети, порт, имя прокси/отсутствие прокси) и детали, специфичные для элемента данных (такие как SNMPv2 community или учётные данные безопасности SNMPv3). Следующие поля являются контекстно-зависимыми:
  - Значения предварительно заполняются, когда это возможно (например, для элементов данных, требующих агента, информация будет взята из выбранного интерфейса агента узла сети)
  - Для элементов данных с шаблонов значения необходимо заполнить вручную
  - Значения макросов в виде простого текста раскрываются
  - Если значение поля (или часть значения) является скрытым макросом или секретным макросом Хранилища, поле будет пустым и должно быть заполнено вручную. Если какой-либо параметр элемента данных содержит значение скрытого макроса, отобразится следующее сообщение с предупреждением: «Элемент данных содержит пользовательский макрос со скрытыми значениями. Значения таких макросов необходимо ввести вручную».
  - Поля отключаются, когда они не требуются в контексте типа элемента данных (например, поля адреса узла сети и прокси отключены для вычисляемых элементов данных)
- Нажмите *Получить значение и протестировать*, чтобы протестировать предварительную обработку.



Если вы выбрали преобразование значений в диалоге настройки элемента данных (поле «Преобразование значений»), в диалоге тестирования элемента данных появится ещё одна строка после окончательного результата, которая называется «Применён результат с преобразованием значений».

Параметры, специфичные для получения реального значения с узла сети:

Параметр	Описание
Получить значение с узла сети (Get value from host)	Отметьте эту опцию, чтобы получить реальное значение от узла сети.
Адрес узла сети (Host address)	Введите адрес узла сети. Это поле автоматически заполняется адресом интерфейса узла сети этого элемента данных.
Порт (Port)	Введите порт узла сети. Это поле автоматически заполняется номером порта интерфейса узла сети этого элемента данных.
Дополнительные поля для SNMP интерфейсов (Версия SNMP, SNMP community, Имя контекста и т.д.)	Смотрите Настройка SNMP мониторинга для получения дополнительных сведений о настройке SNMP интерфейса (v1, v2 и v3). Эти поля автоматически заполняются с интерфейса узла сети, которому принадлежит элемент данных.
Прокси (Proxy)	Укажите прокси, если узел сети наблюдается через прокси. В это поле автоматически подставляется сервер прокси данного узла сети (если прокси используется).

Параметр	Описание
Значение (Value)	Значение, полученное от узла сети. При щелчке в поле параметра или на кнопку просмотра/редактирования  откроется новое окно для ввода значения или блока кода. Значения усекаются до максимального размера 512КБ и только в веб-интерфейсе. Если результат усечён, отобразится значок предупреждения. Описание предупреждения отобразится при наведении указателя мыши. Обратите внимание, что данные более 512КБ, тем не менее, полностью обрабатываются Zabbix сервером.

Для понимания остальных параметров смотрите Тестирование гипотетического значения выше.

2. Дополнительные сведения о предобработке

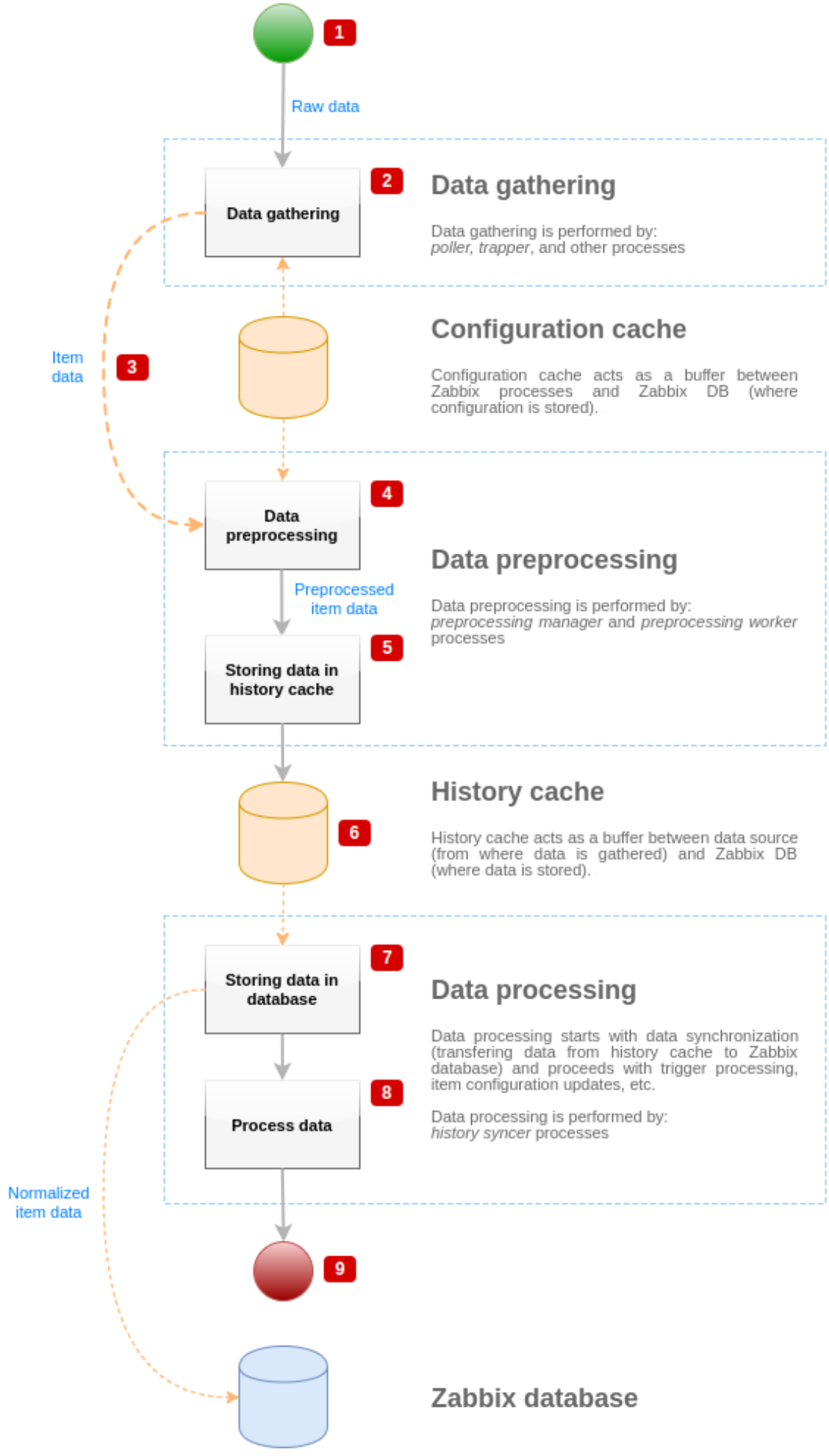
Обзор

В этом разделе приводятся сведения о предобработке значений элементов данных. Предобработка значений элементов данных позволяет определять и выполнять правила преобразования для полученных значений элементов данных.

Предобработка управляется процессом менеджера предобработки вместе с рабочими процессами предобработки, которые выполняют шаги предобработки. Все значения с предобработкой, полученные от различных сборщиков данных, проходят через менеджер предобработки перед добавлением в кэш истории. Для взаимодействия между сборщиками данных (poller, trapper и т. д.) и процессом предобработки используется межпроцессное взаимодействие (IPC) на основе сокетов. Шаги предобработки выполняются либо Zabbix сервером, либо Zabbix прокси (для элементов данных, отслеживаемых прокси).

Обработка значений элемента данных

Чтобы визуализировать поток данных от источника данных до базы данных Zabbix, можно использовать следующую упрощённую схему:


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

На приведённой выше схеме показаны только процессы, объекты и действия, связанные с обработкой значений элемента данных, в **упрощённом** виде. На схеме не показаны условные изменения направления, обработка ошибок или циклы. Также не показан локальный кэш данных менеджера предобработки, поскольку он не влияет напрямую на поток данных. Цель этой схемы — показать процессы, участвующие в обработке значений элемента данных, и способ их взаимодействия.

- Сбор данных начинается с необработанных данных из источника данных. На этом этапе данные содержат только ID, временную метку и значение (также это могут быть несколько значений).
- Независимо от того, какой тип сборщика данных используется, идея остаётся одной и той же для активных или пассивных проверок, для элементов данных trapper и т. д., поскольку меняются только формат данных и инициатор обмена данными (либо сборщик данных ожидает соединения и данные, либо сборщик данных инициирует обмен и запрашивает данные). Необработанные данные проходят проверку, конфигурация элемента данных извлекается из кэша конфигурации (данные дополняются конфигурационными данными).
- Для передачи данных от сборщиков данных менеджеру предобработки используется механизм межпроцессного взаимодействия (IPC) на основе сокетов. На этом этапе сборщик данных продолжает собирать данные, не ожидая ответа от менеджера предобработки.
- Выполняется предобработка данных. Это включает выполнение шагов предобработки и обработку зависимых элементов данных.

Note:
Элемент данных может изменить своё состояние на NOT SUPPORTED во время выполнения предобработки, если какой-либо из шагов предобработки завершится ошибкой.

- Исторические данные из локального кэша данных менеджера предобработки сбрасываются в кэш истории.
- На этом этапе поток данных останавливается до следующей синхронизации кэша истории (когда процесс history syncer выполняет синхронизацию данных).
- Процесс синхронизации начинается с нормализации данных перед сохранением данных в базе данных Zabbix. Нормализация данных выполняет преобразования к требуемому типу элемента данных (тип определяется в конфигурации элемента данных), включая усечение текстовых данных на основе предопределённых размеров, допустимых для этих типов (HISTORY_STR_VALUE_LEN для string, HISTORY_TEXT_VALUE_LEN для text и HISTORY_LOG_VALUE_LEN для значений log). После завершения нормализации данные отправляются в базу данных Zabbix.

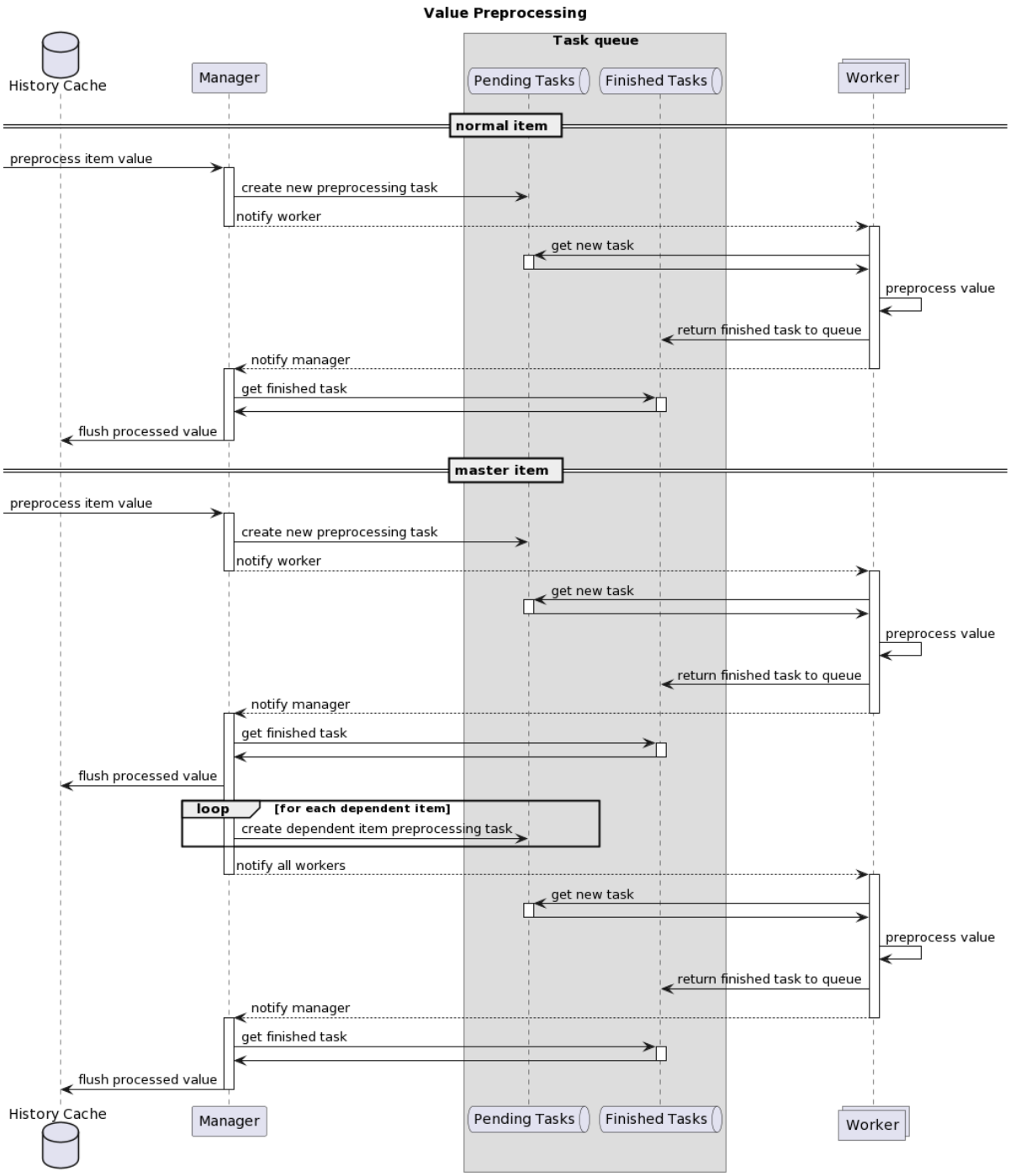
Note:
Элемент данных может изменить своё состояние на NOT SUPPORTED, если нормализация данных завершится ошибкой (например, когда текстовое значение не может быть преобразовано в число).

- Выполняется обработка собранных данных — 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 элемента данных.

Предобработка значений элементов данных

Предобработка данных выполняется в следующие этапы:

- Если у элемента данных нет ни предобработки, ни зависимых элементов данных, его значение либо добавляется в кэш истории, либо отправляется менеджеру LLD. В противном случае значение элемента данных передаётся менеджеру предобработки с использованием механизма IPC на основе UNIX-сокетов.
- Создаётся задача предобработки, 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 берёт следующую задачу из очереди.
- После завершения предобработки (как при неудачном, так и при успешном выполнении этапов предобработки) предобработанное значение добавляется в очередь завершённых задач, а менеджер уведомляется о новой завершённой задаче.
- Менеджер предобработки преобразует результат в требуемый формат (определяемый типом значения элемента данных) и либо добавляет его в кэш истории, либо отправляет менеджеру LLD.
- Если у обработанного элемента данных есть зависимые элементы данных, они добавляются в очередь предобработки с предобработанным значением мастер-элемента данных. Зависимые элементы данных ставятся в очередь в обход обычных запросов на предобработку значений, но только для мастер-элементов данных, у которых значение задано и которые не находятся в состоянии NOT SUPPORTED.



Обратите внимание, что на диаграмме предобработка мастер-элемента данных немного упрощена за счёт 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 == "value")]`)

Потоки предобработки

Файл конфигурации сервера Zabbix позволяет пользователям задавать количество потоков-обработчиков предобработки. Параметр конфигурации StartPreprocessors следует использовать для задания количества заранее запущенных экземпляров обработчиков предобработки, которое должно как минимум соответствовать числу доступных ядер CPU.

Если задачи предобработки не ограничены производительностью CPU и включают частые сетевые запросы, рекомендуется настроить дополнительные обработчики. Оптимальное количество обработчиков предобработки может определяться многими факторами, включая количество "предобрабатываемых" элементов данных (элементов данных, для которых требуется выполнение каких-либо шагов предобработки), количество процессов сбора данных, среднее число шагов предобработки элемента данных и т. д. Недостаточное количество обработчиков может привести к высокому потреблению памяти. Для устранения чрезмерного потребления памяти в вашей установке Zabbix см. Профилирование чрезмерного потребления памяти с помощью tcmalloc.

Однако если предположить, что отсутствуют ресурсоемкие операции предобработки, такие как разбор больших фрагментов XML/JSON, то количество обработчиков предобработки может соответствовать общему числу сборщиков данных. В этом случае в большинстве ситуаций (за исключением случаев, когда данные от сборщика поступают пакетно) для собранных данных будет доступен как минимум один незанятый обработчик предобработки.

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 обработчиками предобработки может исчерпать лимит файловых дескрипторов на процесс для менеджера предобработки.

 Исчерпание лимита файловых дескрипторов на процесс приведет к остановке сервера Zabbix, как правило, вскоре после запуска, хотя иногда это может произойти и позже. Чтобы избежать таких проблем, проверьте файл конфигурации сервера Zabbix, чтобы оптимизировать количество параллельных 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.

#####3. Примеры предобработки {#manual-config-items-preprocessing-examples}

Обзор

В этом разделе представлены примеры использования шагов предобработки для выполнения некоторых практических задач.

Фильтрация записей журнала событий VMware

В этом примере используется шаг предобработки Совпадает с регулярным выражением для фильтрации ненужных событий из журнала событий VMware.

1. На рабочем узле сети VMware Hypervisor проверьте, что элемент данных vmware.eventlog присутствует и работает корректно. Обратите внимание, что элемент данных журнала событий уже может присутствовать на гипервизоре, если при создании узла сети был привязан шаблон VMware.

2. На узле сети VMware Hypervisor создайте зависимый элемент данных типа *Log* и укажите элемент данных журнала событий в качестве его мастер-элемента данных.

3. На вкладке *Preprocessing* зависимого элемента данных нажмите *Add*, чтобы создать шаг предобработки, и выберите *Matches regular expression* из выпадающего списка. Затем укажите один из следующих шаблонов:

- Для фильтрации всех событий журнала:

```
.* logged in .*
```

- Для фильтрации строк, содержащих имена пользователей после "User":

```
\bUser\s+\K\S+
```

Attention:
Если регулярное выражение не совпадает, зависимый элемент данных становится неподдерживаемым с соответствующим сообщением об ошибке. Чтобы избежать этого, установите флажок *Custom on fail* и выберите такой вариант, как отбрасывание значения или установка пользовательского значения. Обратите внимание, что отброшенные значения не сохраняются в базе данных; в результате триггеры не вычисляются, а данные трендов не создаются.

В качестве альтернативы можно использовать шаг предобработки Regular expression для извлечения совпадающих групп и управления выводом:

- Для извлечения и вывода всего события журнала, содержащего "logged in", укажите следующие параметры:

```
Pattern: .*logged in.*
Output: \0
```

- Для извлечения и вывода имен пользователей после "User":

```
Pattern: User (.*?)(?=\ )
Output: \1
```

Проверка типа полученного значения

В этом примере используется шаг предобработки Пользовательский множитель, чтобы проверить, является ли тип полученного значения элемента данных числовым.

На вкладке *Предобработка* элемента данных выберите шаг предобработки *Пользовательский множитель* и укажите следующий параметр (умножает полученное значение на 1):

```
1
```

Attention:
Если предобработка завершается ошибкой (например, входное значение не является числовым), то элемент данных становится неподдерживаемым с соответствующим сообщением об ошибке. Чтобы избежать этого, установите флажок *Пользовательское действие при ошибке* и выберите вариант, например отбрасывание значения или установку пользовательского значения. Обратите внимание, что отброшенные значения не сохраняются в базе данных; в результате триггеры не вычисляются и данные трендов не создаются.

Проверка неподдерживаемого значения

В этом примере используется шаг предобработки Проверка неподдерживаемого значения, чтобы проверить, не удалось ли получить значение элемента данных.

Когда процесс poller сервера/прокси Zabbix пытается собрать значение элемента данных, он может:

- Вернуть корректный результат.
- Вернуть результат, который изначально кажется корректным, но позже может стать неподдерживаемым (например, из-за несоответствия типа значения после предобработки).
- Вернуть ошибку при сборе значения, из-за чего элемент данных становится неподдерживаемым. Распространённые причины включают:
  - Неизвестный ключ элемента данных (для элементов данных Zabbix агент, Simple check или Zabbix internal)
  - Неизвестный OID (SNMP агент), неизвестный сенсор (IPMI агент) или отсутствие метрики JMX (JMX агент)
  - Не удаётся прочитать файл trap (SNMP trap)
  - Скрипт не найден (External check)
  - Такой URL не существует (HTTP агент, Browser)
  - Ошибка входа (SSH агент, TELNET агент)
  - Неверный синтаксис формулы (Calculated), синтаксическая ошибка JavaScript (Script) или неверный SQL-запрос (Database monitor)

Чтобы обнаруживать и обрабатывать ошибки при сборе значений элементов данных, можно использовать шаг предобработки *Проверка неподдерживаемого значения*. Обратите внимание, что этот шаг всегда выполняется первым и обнаруживает только ошибки, возникающие до начала предобработки.

На вкладке *Предобработка* элемента данных выберите шаг предобработки *Проверка неподдерживаемого значения* и укажите один из следующих параметров:

- Для любых ошибок:

```
Parameter: any error
```

- Для ошибок, содержащих "cannot connect":

```
Parameter: error matches
Pattern: (?i)cannot connect
```

Затем используйте опцию *Custom on fail*, чтобы отбросить значение (в данном случае ошибку), задать пользовательское значение или вернуть пользовательское сообщение об ошибке. Обратите внимание, что отброшенные значения не сохраняются в базе данных; в результате триггер не вычисляются и данные трендов не генерируются.

4. Поддержка функционала 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.
`$.['object'].["name"]`	Возвращает содержимое object.name.
`$.object.history.length()`	Возвращает количество элементов массива object.history.

Пример JSONPath	Описание
`$[?(@.name == 'Object')].price.first()`	Возвращает значение свойства `price` первого объекта по имени «Object».
`$[?(@.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 является арифметическим выражением в инфиксной нотации.

Поддерживаемые операнды:

Операнд	Описание

`"< >"`
`'<text>'`
	Текстовая константа.
	
	Пример:
	`'value: \\'1\\''`
	`"value: '1'"`

`< >`	Числовая константа, поддерживая научное представление. Пример: 123
`<jsonpath $>`	Значение, указанное в JSONPath, начиная с корневого узла входного документа; поддерживаются только определённые пути. Пример: `$.object.name`
`<jsonpath @>`	Значение, указанное в JSONPath, начиная с текущего объекта/элемента; поддерживаются только определённые пути. Пример: `@.name`

Поддерживаемые операторы:

Оператор	Тип	Описание	Результат
-	бинарный	Вычитание	Число
+	бинарный	Сложение	Число
/	бинарный	Деление	Число
*	бинарный	Умножение	Число
==	бинарный	Равно	Логический (1 или 0)
!=	бинарный	Не равно	Логический (1 или 0)
	бинарный	Меньше чем	Логический (1 или 0)
<=	бинарный	Меньше чем или равно	Логический (1 или 0)
>	бинарный	Больше чем	Логический (1 или 0)
>=	бинарный	Больше чем или равно	Логический (1 или 0)
=~	бинарный	Соответствие регулярному выражению	Логический (1 или 0)
!	унарный	Логическое НЕ	Логический (1 или 0)
||	бинарный	Логическое ИЛИ	Логический (1 или 0)
&&	бинарный	Логическое И	Логический (1 или 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": "Sayings of the Century",
      "price": 8.95,
      "id": 1
    },
    {
      "category": "fiction",
      "author": "Evelyn Waugh",
      "title": "Sword of Honour",
      "price": 12.99,
      "id": 2
    },
    {
      "category": "fiction",
      "author": "Herman Melville",
      "title": "Moby Dick",
      "isbn": "0-553-21311-3",
      "price": 8.99,
      "id": 3
    },
    {
      "category": "fiction",
      "author": "J. R. R. Tolkien",
      "title": "The Lord of the Rings",
      "isbn": "0-395-19395-8",
      "price": 22.99,
      "id": 4
    }
  ],
  "services": {
    "delivery": {
      "servicegroup": 1000,
      "description": "Next day delivery in local town",
      "active": true,
      "price": 5
    },
    "bookbinding": {
      "servicegroup": 1001,
      "description": "Printing and assembling book in A5 format",
      "active": true,
      "price": 154.99
    },
    "restoration": {
      "servicegroup": 1002,
      "description": "Various restoration methods",
      "active": false,
      "methods": [
        {
          "description": "Chemical cleaning",
          "price": 46
        },
```

```
        {
          "description": "Pressing pages damaged by moisture",
          "price": 24.5
        },
        {
          "description": "Rebinding torn book",
          "price": 99.49
        }
      ]
    }
  },
  "filters": {
    "price": 10,
    "category": "fiction",
    "no filters": "no \"filters\""
  },
  "closed message": "Store is closed",
  "tags": [
    "a",
    "b",
    "c",
    "d",
    "e"
  ]
}
```

JSONPath	Type	Result
`$.filters.price`	определённый	10
`$.filters.category`	определённый	fiction
`$.filters['no filters']`	определённый	no "filters"

`$.filters`	определённый	{
		"price": 10,
		"category": "fiction",
		"no filters": "no \"filters\""
		}

`$.books[1].title`	определённый	Sword of Honour
`$.books[-1].author`	определённый	J. R. R. Tolkien
`$.books.length()`	определённый	4
`$.tags[:]`	неопределённый	["a", "b", "c", "d", "e"]
`$.tags[2:]`	неопределённый	["c", "d", "e"]
`$.tags[:3]`	неопределённый	["a", "b", "c"]
`$.tags[1:4]`	неопределённый	["b", "c", "d"]
`$.tags[-2:]`	неопределённый	["d", "e"]
`$.tags[:-3]`	неопределённый	["a", "b"]
`$.tags[:-3].length()`	определённый	2
`$.books[0, 2].title`	неопределённый	["Moby Dick", "Sayings of the Century"]
`$.books[1]['author', "title"]`	неопределённый	["Sword of Honour", "Evelyn Waugh"]
`$..id`	неопределённый	[1, 2, 3, 4]
`$.services..price`	неопределённый	[14.99, 5, 46, 24.5, 99.49]
`$.books[?(@.id == 4 - 0.4 * 5)].title`	неопределённый	["Sword of Honour"] Примечание: Этот запрос показывает, что в запросах можно использовать математические операции. Конечно, этот запрос можно упростить до `$.books[?(@.id == 2)].title`
`$.books[?(@.id == 2 || @.id == 4)].title`	неопределённый	["Sword of Honour", "The Lord of the Rings"]
`$.books[?(!(@.id == 2))].title`	неопределённый	["Sayings of the Century", "Moby Dick", "The Lord of the Rings"]
`$.books[?(@.id != 2)].title`	неопределённый	["Sayings of the Century", "Moby Dick", "The Lord of the Rings"]

JSONPath	Type	Result
`$.books[?(@.title =~ " of ")].title`	неопределённый	[”Sayings of the Century”, ”Sword of Honour”, ”The Lord of the Rings”]
`$.books[?(@.price > 12.99)].title`	неопределённый	[”The Lord of the Rings”]
`$.books[?(@.author > "Herman Melville")].title`	неопределённый	[”Sayings of the Century”, ”The Lord of the Rings”]
`$.books[?(@.price > $.filters.price)].title`	неопределённый	[”Sword of Honour”, ”The Lord of the Rings”]
`$.books[?(@.category == $.filters.category)].title`	неопределённый	[”Sword of Honour”,”Moby Dick”,”The Lord of the Rings”]
`$.books[?(@.category == "fiction" && @.price < 10)].title`	неопределённый	[”Moby Dick”]

`$..[?(@.id)]`	неопределённый	[
		{
		”price”: 8.95,
		”id”: 1,
		”category”: ”reference”,
		”author”: ”Nigel Rees”,
		”title”: ”Sayings of the Century”
		},
		{
		”price”: 12.99,
		”id”: 2,
		”category”: ”fiction”,
		”author”: ”Evelyn Waugh”,
		”title”: ”Sword of Honour”
		},
		{
		”price”: 8.99,
		”id”: 3,
		”category”: ”fiction”,
		”author”: ”Herman Melville”,
		”title”: ”Moby Dick”,
		”isbn”: ”0-553-21311-3”
		},
		{
		”price”: 22.99,
		”id”: 4,
		”category”: ”fiction”,
		”author”: ”J. R. R. Tolkien”,
		”title”: ”The Lord of the Rings”,
		”isbn”: ”0-395-19395-8”
		}
		]

`$.services..[?(@.price > 50)].description`	неопределённый	[”Printing and assembling book in A5 format”, ”Rebinding torn book”]
`$..id.length()`	определённый	4
`$.books[?(@.id == 2)].title.first()`	определённый	Sword of Honour
`$..tags.first().length()`	определённый	5 Примечание: `$..tags` неопределённый путь, поэтому он вернёт массив соответствующих элементов, т.е. `[["a", "b", "c", "d", "e" ]]`; `first()` вернёт первый элемент, т.е., `["a", "b", "c", "d", "e"]`; `length()` вычислит его размер, т.е.5.
`$.books[*].price.min()`	определённый	8.95
`$..price.max()`	определённый	154.99
`$.books[?(@.category == "fiction")].price.avg()`	определённый	14.99
`$.books[?(@.category == $.filters.xyz)].title`	неопределённый	Примечание: Запрос без совпадения вернёт NULL по определённому и неопределённому путям.

JSONPath	Type	Result
`$.services[?(@.active=="true")].serviceid`	Не предупреждать	[1001,1000]
		Примечание: В сравнениях логических значений необходимо использовать текстовые константы.
`$.services[?(@.active=="false")].serviceid`	Не предупреждать	[1002]
		Примечание: В сравнениях логических значений необходимо использовать текстовые константы.
`$.services[?(@.servicegroup=="1000")].serviceid`	Пользовательский	first(1)

1. Экранирование спецсимволов из значений LLD макросов в JSONPath

Когда в предварительной обработке JSONPath используются макросы низкоуровневого обнаружения и их значения раскрываются, применяются следующие правила экранирования спецсимволов:

- для экранирования учитываются только символы обратной косой черты (\) и двойной кавычки (”);
- если раскрытое значение макроса содержит эти символы, каждый из них экранируется при помощи обратной косой черты;
- если эти символы уже экранированы при помощи обратной косой черты, такая запись не считается экранированием и как косая черта, так и последующие спецсимволы будут экранированы еще раз.

Например:

JSONPath	Значение LLD макроса	После подстановки
$.[?(@.value == ”{#MACRO}”)]	special ”value”	$.[?(@.value == ”special \”value\””)]
	c:\temp	$.[?(@.value == ”c:\\temp”)]
	a\\b	$.[?(@.value == ”a\\\\b”)]

Макрос, который может иметь спецсимволы, при использовании в выражении необходимо заключить в двойные кавычки:

JSONPath	Значение LLD макроса	После подстановки	Результат
$.[?(@.value == ”{#MACRO}”)]	special ”value”	$.[?(@.value == ”special \”value\””)]	OK
$.[?(@.value == {#MACRO})]		$.[?(@.value == special \”value\”)]	**Плохое JSONPath выражение**

Макрос который может иметь спецсимволы, при использовании в пути необходимо заключить в квадратные скобки **и** двойные кавычки:

JSONPath	Значение LLD макроса	После подстановки	Результат
$.[”{#MACRO}”].value	c:\temp	$.[”c:\\temp”].value	OK
$.{#MACRO}.value		$.c:\\temp.value	**Плохое JSONPath выражение**

5. Предобработка Javascript

Обзор

В этом разделе приводятся сведения о предобработке с помощью JavaScript.

Attention:
Не используйте необъявленные присваивания в JavaScript предобработки. 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 параметры предобработки JavaScript, после чего сервер обернет это 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, ошибка возвращается как строковое значение), за несколькими исключениями:

- возврат значения undefined приведет к ошибке;
- возврат значения null приведет к отбрасыванию входного значения, аналогично предобработке 'Discard value' при действии 'Custom on fail'.

Ошибки могут возвращаться путем генерации значений/объектов (обычно это строки или объекты Error).

Например:

```
if (value == 0)
    throw "Zero input value"
return 1/value
```

Для каждого скрипта действует тайм-аут выполнения 10 секунд (в зависимости от скрипта срабатывание тайм-аута может занять больше времени); при его превышении будет возвращена ошибка. Применяется ограничение heap-памяти в 512 мегабайт.

Байткод шага предобработки JavaScript кэшируется и повторно используется при следующем применении этого шага. Любые изменения в шагах предобработки элемента данных приведут к сбросу кэшированного скрипта и его последующей повторной компиляции.

Последовательные ошибки выполнения (3 подряд) приведут к повторной инициализации движка, чтобы снизить вероятность того, что один скрипт нарушит среду выполнения для следующих скриптов (это действие журналируется с DebugLevel 4 и выше).

Предобработка JavaScript реализована с использованием движка JavaScript Duktape.

См. также: Дополнительные объекты JavaScript и глобальные функции

Использование макросов в скриптах

В коде JavaScript можно использовать пользовательские макросы (а также LLD-макросы в контексте обнаружения низкого уровня). Если скрипт содержит пользовательские макросы, эти макросы разрешаются сервером/прокси перед выполнением конкретных шагов предобработки. Обратите внимание, что при тестировании шагов предобработки в веб-интерфейсе значения макросов не будут подставлены и их необходимо вводить вручную.

> **Note:**
> Контекст игнорируется, когда макрос заменяется своим значением. Значение макроса вставляется в код как есть; невозможно добавить дополнительное экранирование перед помещением значения в код JavaScript. Обратите внимание, что в некоторых случаях это может вызывать ошибки JavaScript.

В приведенном ниже примере, если полученное значение превышает значение макроса {$THRESHOLD}, вместо него будет возвращено пороговое значение (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

Преобразуйте значение, полученное из элемента данных агента Zabbix web.page.get (например, web.page.get[http://127.0.0.1:80/serv
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, принимаемое 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
    }
}
```

1. Дополнительные объекты Javascript

Обзор

В этом разделе описаны дополнения Zabbix к языку JavaScript, реализованные с помощью Duktape, а также поддерживаемые глобальные функции JavaScript.

Attention:
Не используйте необъявленные присваивания в JavaScript предобработки. Используйте `var` для объявления локальных переменных.

Встроенные объекты

Zabbix

Объект Zabbix обеспечивает взаимодействие с внутренней функциональностью Zabbix.

Method	Description
`log(loglevel, message)`	Записывает <message> в журнал Zabbix с использованием уровня журналирования <loglevel> (см. параметр DebugLevel в файле конфигурации).

Пример:

```
Zabbix.log(3, "this is a log entry written with 'Warning' log level")
```

Вы можете использовать следующие псевдонимы:

Alias	Alias to
console.log(object)	Zabbix.log(4, JSON.stringify(object))
console.warn(object)	Zabbix.log(3, JSON.stringify(object))
console.error(object)	Zabbix.log(2, JSON.stringify(object))

Attention:
Общий размер всех записываемых в журнал сообщений ограничен 8 МБ на одно выполнение скрипта.

Method	Description
`sleep(delay)`	Задерживает выполнение JavaScript на `delay` миллисекунд.

Пример (задержка выполнения на 15 секунд):

```
Zabbix.sleep(15000)
```

HttpRequest

Этот объект инкапсулирует дескриптор cURL, позволяя выполнять простые HTTP-запросы. Ошибки генерируются как исключения.

Attention:
Инициализация нескольких объектов `HttpRequest` ограничена 10 на одно выполнение скрипта.

Method	Description
`addHeader(value)`	Добавляет поле HTTP-заголовка. Это поле используется для всех последующих запросов, пока не будет очищено методом `clearHeader()`. Суммарная длина полей заголовка, которые можно добавить в один объект `HttpRequest`, ограничена 128 Кбайтами (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 — text/plain.
`connect(url)`	Отправляет HTTP-запрос CONNECT на URL и возвращает ответ.
`customRequest(method, url, data)`	Позволяет указать любой HTTP-метод в первом параметре. Отправляет запрос указанным методом на URL с необязательной полезной нагрузкой *data* и возвращает ответ.
`delete(url, data)`	Отправляет HTTP-запрос DELETE на URL с необязательной полезной нагрузкой *data* и возвращает ответ.
`getHeaders(<asArray>)`	Возвращает объект полученных полей HTTP-заголовка. Параметр `asArray` может быть установлен в "true" (например, `getHeaders(true)`), "false" или быть неопределённым. Если установлено значение "true", значения полученных полей HTTP-заголовка будут возвращены как массивы; это следует использовать для получения значений полей нескольких заголовков с одинаковыми именами. Если параметр не задан или установлен в "false", значения полученных полей HTTP-заголовка будут возвращены как строки.
`get(url, data)`	Отправляет HTTP-запрос GET на URL с необязательной полезной нагрузкой *data* и возвращает ответ.
`head(url)`	Отправляет HTTP-запрос HEAD на URL и возвращает ответ.
`options(url)`	Отправляет HTTP-запрос OPTIONS на URL и возвращает ответ.
`patch(url, data)`	Отправляет HTTP-запрос PATCH на URL с необязательной полезной нагрузкой *data* и возвращает ответ.
`put(url, data)`	Отправляет HTTP-запрос PUT на URL с необязательной полезной нагрузкой *data* и возвращает ответ.
`post(url, data)`	Отправляет HTTP-запрос POST на URL с необязательной полезной нагрузкой *data* и возвращает ответ.
`getStatus()`	Возвращает код состояния последнего HTTP-запроса.
`setProxy(proxy)`	Устанавливает HTTP-прокси в значение "proxy". Если этот параметр пуст, прокси не используется.

`setHttpAuth(bitmask, username, password)`	Устанавливает включённые 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)`

Method	Description
`trace(url, data)`	Отправляет HTTP-запрос TRACE на URL с необязательной полезной нагрузкой *data* и возвращает ответ.

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 при предобработке элемента данных, низкоуровневом обнаружении и в вебхуках.

Attention:
Для использования объекта XML сервер/прокси должны быть скомпилированы с поддержкой libxml2.

Method	Description
`XML.query(data, expression)`	Извлекает содержимое узла с помощью XPath. Возвращает null, если узел не найден. **expression** - выражение XPath; **data** - XML-данные в виде строки.
`XML.toJson(data)`	Преобразует данные в формате XML в JSON.
`XML.fromJson(object)`	Преобразует данные в формате JSON в XML.

Пример:

Входные данные:

```
<menu>
    <food type = "breakfast">
        <name>Chocolate</name>
        <price>$5.95</price>
        <description></description>
        <calories>650</calories>
    </food>
</menu>
```

Выходные данные:

```
{
    "menu": {
        "food": {
            "@type": "breakfast",
            "name": "Chocolate",
            "price": "$5.95",
            "description": null,
            "calories": "650"
        }
    }
}
```

Правила сериализации

Преобразование XML в JSON будет выполняться в соответствии со следующими правилами (для преобразования JSON в XML применяются обратные правила):

1. Атрибуты XML будут преобразованы в ключи, к именам которых добавляется префикс '@'.

Пример:

Входные данные:

```
<xml foo="FOO">
  <bar>
    <baz>BAZ</baz>
  </bar>
</xml>
```

Выходные данные:

```
{
  "xml": {
    "@foo": "FOO",
    "bar": {
      "baz": "BAZ"
    }
  }
}
```

2. Самозакрывающиеся элементы (<foo/>) будут преобразованы как имеющие значение 'null'.

Пример:

Входные данные:

```
<xml>
  <foo/>
</xml>
```

Выходные данные:

```
{
  "xml": {
    "foo": null
```

```
  }
}
```

3. Пустые атрибуты (со значением "") будут преобразованы как имеющие значение пустой строки ('').

Пример:

Входные данные:

```
<xml>
  <foo bar="" />
</xml>
```

Выходные данные:

```
{
  "xml": {
    "foo": {
      "@bar": ""
    }
  }
}
```

4. Несколько дочерних узлов с одинаковым именем элемента будут преобразованы в один ключ, значением которого будет массив значений.

Пример:

Входные данные:

```
<xml>
  <foo>BAR</foo>
  <foo>BAZ</foo>
  <foo>QUX</foo>
</xml>
```

Выходные данные:

```
{
  "xml": {
    "foo": ["BAR", "BAZ", "QUX"]
  }
}
```

5. Если текстовый элемент не имеет атрибутов и дочерних элементов, он будет преобразован в строку.

Пример:

Входные данные:

```
<xml>
    <foo>BAZ</foo>
</xml>
```

Выходные данные:

```
{
  "xml": {
    "foo": "BAZ"
   }
}
```

6. Если текстовый элемент не имеет дочерних элементов, но имеет атрибуты, текстовое содержимое будет преобразовано в элемент с ключом '#text', а содержимое — в значение; атрибуты будут преобразованы, как описано в правиле сериализации 1.

Пример:

Входные данные:

```
<xml>
  <foo bar="BAR">
    BAZ
```

```
  </foo>
</xml>
```

Выходные данные:

```
{
  "xml": {
    "foo": {
      "@bar": "BAR",
      "#text": "BAZ"
    }
  }
}
```

Глобальные функции JavaScript

С помощью Duktape были 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-хеш данных.
- sha256(data) — вычисляет SHA256-хеш данных.
- hmac('<hash type>',key,data) — возвращает HMAC-хеш 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 сгенерирована ошибка.
 key — закрытый ключ. Он должен соответствовать стандарту PKCS#1 или PKCS#8. Ключ может быть предоставлен в разных формах:

  - с пробелами вместо переводов строк
  - с экранированными или неэкранированными \n вместо переводов строк
  - без переводов строк, как однострочная строка
  - как строка в формате JSON

  Ключ также может быть загружен из пользовательского макроса/секретного макроса/хранилища секретов.

  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 библиотек шифрования, будет сгенерирована ошибка (’missing OpenSSL or GnuTLS library’).

2. Объекты JavaScript элемента данных «Браузер»

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 сессиями WebDriver, инициализируя сессию при создании и завершая её при уничтожении. Один скрипт может поддерживать до четырёх объектов `Browser`.

Чтобы создать объект `Browser`, используйте синтаксис `new Browser(options)`. Параметр `options` (*JSON object*) задаёт параметры браузера, обычно это результат метода параметров WebDriver (например, `Browser.chromeOptions()`).

Для объекта `Browser` поддерживаются следующие методы.

Method	Description
`navigate(url)`	Перейти по указанному URL. Параметры: `url` - (string) URL для перехода.
`getUrl()`	Вернуть строку с URL открытой страницы.
`getPageSource()`	Вернуть строку с исходным кодом открытой страницы.
`findElement(strategy, selector)`	Вернуть объект `Element` с одним элементом на открытой странице (или вернуть `null`, если ни один элемент не соответствует `strategy` и `selector`). Параметры: `strategy` - (string, css selector/link text/partial link text/tag name/xpath) Стратегия поиска; `selector` - (string) Селектор элемента с использованием указанной стратегии поиска.
`findElements(strategy, target)`	Вернуть массив объектов `Element` с несколькими элементами на открытой странице (или вернуть пустой массив, если ни один элемент не соответствует стратегии поиска и target). Параметры: `strategy` - (string, css selector/link text/partial link text/tag name/xpath) Стратегия поиска; `target` - (string) Селектор элемента с использованием указанной стратегии поиска.
`getCookies()`	Вернуть массив объектов `Cookie`.
`addCookie(cookie)`	Установить cookie. Параметры: `cookie` - (`Cookie` object) Cookie для установки.
`getScreenshot()`	Вернуть строку в кодировке base64, представляющую изображение области просмотра браузера.
`setScreenSize(x,y)`	Установить размер области просмотра браузера. Параметры: `x` - (string) Ширина в пикселях; `y` - (string) Высота в пикселях.
`setScriptTimeout(timeout)`	Установить тайм-аут загрузки скрипта. Параметры: `timeout` - (integer) Значение тайм-аута в миллисекундах.
`setSessionTimeout(timeout)`	Установить тайм-аут сессии (загрузки страницы). Параметры: `timeout` - (integer) Значение тайм-аута в миллисекундах.
`setElementWaitTimeout(timeout)`	Установить неявный тайм-аут стратегии поиска элемента. Параметры: `timeout` - (integer) Значение тайм-аута в миллисекундах.

Method	Description
`collectPerfEntries(mark)`	Собрать записи производительности для последующего получения методом `getResult()`. Параметры: `mark` - (string, optional) Метка снимка производительности.
`getRawPerfEntries()`	Вернуть массив объектов записей производительности.
`getResult()`	Вернуть объект `Result` со статистикой сессии браузера (информация об ошибках, снимки производительности и т. д.).
`getError()`	Вернуть объект `BrowserError` с ошибками браузера (или вернуть `null`, если ошибок браузера нет).
`setError(message)`	Установить пользовательское сообщение об ошибке для включения в объект `Result`. Параметры: `message` - (string) Сообщение об ошибке.
`discardError()`	Отбросить ошибку, которая должна быть возвращена в объекте `Result`.
`getAlert()`	Вернуть объект `Alert` с предупреждениями браузера (или вернуть `null`, если предупреждений браузера нет).
`chromeOptions()`	Вернуть объект `chromeOptions` с предопределёнными параметрами браузера Chrome.
`firefoxOptions()`	Вернуть объект `firefoxOptions` с предопределёнными параметрами браузера Firefox.
`safariOptions()`	Вернуть объект `safariOptions` с предопределёнными параметрами браузера Safari.
`edgeOptions()`	Вернуть объект `edgeOptions` с предопределёнными параметрами браузера Edge.
`switchFrame(target)`	Переключиться на указанный фрейм. Параметры: `target` - (browser element or integer, optional) Целевой фрейм. Чтобы выбрать фрейм по элементу, передайте элемент. Чтобы выбрать фрейм по индексу, передайте число. Если параметр не указан, произойдёт переключение на контекст просмотра верхнего уровня.

Все методы Browser могут вызывать следующие ошибки:

- `BrowserError` - производный от объекта `Error`, выбрасывается, если конструктор `Browser` завершается неудачей; содержит дополнительное свойство `browser` с объектом `Browser`, который вызвал этот `BrowserError`.
- `WebdriverError` - производный от `BrowserError`; содержит те же свойства, что и объект `BrowserError`, которые указывают, 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.

Объектом `Element` поддерживаются следующие методы.

Метод	Описание
`getAttribute( )`	Вернуть строку значения атрибута для атрибута элемента (или вернуть `null`, если указанный атрибут не найден). Параметры: — (строка) Имя атрибута.
`getProperty( )`	Вернуть строку значения свойства элемента property (или вернуть `null`, если указанное свойство не найдено). Параметры: — (строка) Имя свойства.
`getText()`	Вернуть строку текста значения текстового элемента.

Метод	Описание
`click()`	Щелчок мышью по элементу.
`clear()`	Очистить содержимое редактируемого элемента.
`sendKeys( )`	Отослать ключи. Параметры: — (строка) Ключи для отправки.

Cookie

Объект `Cookie` возвращается методом `getCookies()` объекта `Browser` и передаётся методу `addCookie()`.

Хотя объект `Cookie` не имеет методов, он может содержать следующие свойства:

Свойство	Тип	Описание
name	string	Имя cookie.
value	string	Значение cookie.
path	string	Путь, для которого cookie является действительным. По умолчанию используется `"/"`, если значение не указано при добавлении cookie.
domain	string	Домен, для которого cookie является видимым. По умолчанию используется домен URL активного документа текущего контекста просмотра сеанса, если значение не указано при добавлении cookie.
secure	boolean	Указывает, является ли cookie защищённым. По умолчанию используется `false`, если значение не указано при добавлении cookie.
httpOnly	boolean	Указывает, является ли cookie доступным только по HTTP. По умолчанию используется `false`, если значение не указано при добавлении cookie.
expiry	integer	Время истечения срока действия cookie (в секундах с начала эпохи Unix). Не должно быть задано, если значение не указано при добавлении cookie.
sameSite	string	Атрибут `sameSite` cookie, который определяет, должен ли cookie быть ограничен контекстом основного сайта или того же сайта. Может быть установлен в `"Lax"` или `"Strict"`. По умолчанию используется `"None"`, если значение не указано при добавлении cookie.

Alert

Объект `Alert` представляет оповещение 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	Error information.
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	object	Статистика производительности.
summary	объект	Сводка производительности.
navigation	объект	Сводка навигации.
resource	объект	Сводка ресурсов.
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.

6. Предобработка CSV в JSON

Обзор

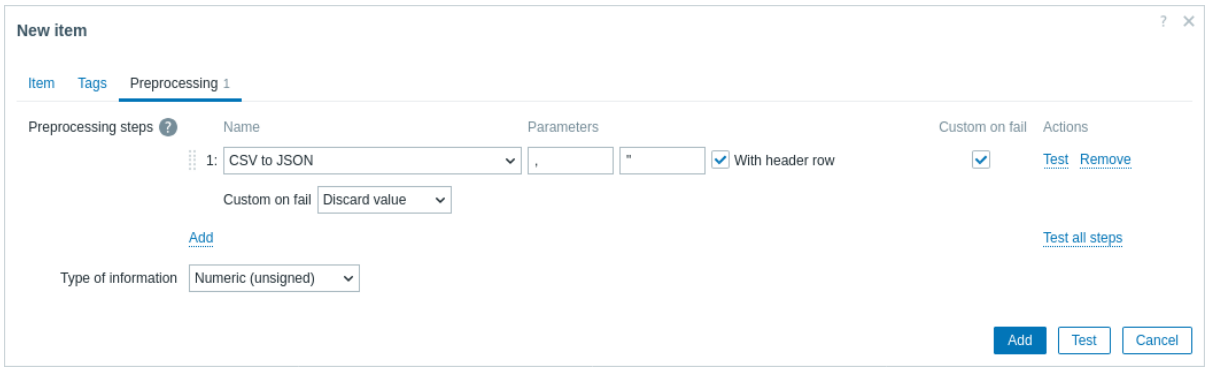
В этом шаге предварительной обработки имеется возможность преобразования данных CSV файла в формат JSON. Поддерживается в:

- элементах данных (прототипах элементов данных)
- правилах низкоуровневого обнаружения

Настройка

Для настройки шага предварительной обработки CSV в JSON:

- Перейдите на вкладку Предобработка в настройках элемента данных / правила обнаружения
- Нажмите на *Добавить*
- Выберите опцию *CSV в JSON*



Первый параметр позволяет задать пользовательский разделитель. Обратите внимание, если первая строка входящих данных CSV начинается с «Sep=» и затем следует один UTF-8 символ, тогда этот символ будет использоваться в качестве разделителя в случае, если первый параметр не задан. Если первый параметр не задан и разделитель не извлекается из строки «Sep=», то в качестве разделителя будет использоваться запятая.

Второй опциональный параметр позволяет задать символ кавычек.

Если отмечен флажок *Со строкой заголовка*, значения строки заголовка будут интерпретированы как имена столбцов (смотрите Обработка заголовка для получения более подробной информации).

Если отмечен флажок *Другое при ошибке*, элемент данных не станет неподдерживаемым в случае ошибки на шаге предварительной обработки. Кроме того, можно задать пользовательскую обработку ошибок: отбросить значение, задать нужное значение или задать нужное 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 заключенного 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"
    }
]
```

3. 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:

- Проверки агента Zabbix
- Проверки SNMP-агента
- SNMP-трапы
- Проверки IPMI
- Простые проверки
  - Мониторинг VMware
- Мониторинг файлов журналов
- Вычисляемые элементы данных
  - Агрегированные вычисления
- Внутренние проверки Zabbix
- Проверки SSH
- Проверки Telnet
- Внешние проверки
- Элементы данных trapper
- Мониторинг JMX
- Проверки ODBC
- Зависимые элементы данных
- Проверки HTTP
- Проверки Prometheus
- Элементы данных script
- Элементы данных browser

Подробная информация обо всех типах элементов данных приведена на вложенных страницах этого раздела. Хотя типы элементов данных предоставляют множество возможностей для сбора данных, существуют и дополнительные возможности через пользовательские параметры или загружаемые модули.

Некоторые проверки выполняются только сервером Zabbix (как безагентный мониторинг), тогда как для других требуется агент Zabbix или даже Zabbix Java gateway (при мониторинге JMX).

> **Attention:**
> Если для определённого типа элемента данных требуется определённый интерфейс (например, для проверки IPMI на узле сети необходим интерфейс IPMI), этот интерфейс должен существовать в определении узла сети.

В определении узла сети можно задать несколько интерфейсов: агент Zabbix, SNMP-агент, JMX и IPMI. Если элемент данных может использовать более одного интерфейса, он будет искать среди доступных интерфейсов узла сети (в порядке: Агент→SNMP→JMX→IPMI) первый подходящий для привязки.

Все элементы данных, которые возвращают текст (типы информации character, log, text), также могут возвращать только пробельные символы (где это применимо), устанавливая возвращаемое значение в пустую строку (поддерживается с версии 2.0).

1. Zabbix агент

Обзор

В этом разделе приводится подробная информация 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 требуемый тип:

- *Zabbix агент* — для пассивных проверок
- *Zabbix агент (active)* — 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

Ключи элементов данных, которые можно использовать с Zabbix агент, перечислены ниже.

Ключи элементов данных приведены без параметров и дополнительной информации. Нажмите на ключ элемента данных, чтобы увидеть полные сведения.

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 файла журнала с ротацией.	
logrt.count	Количество совпавших строк в отслеживаемом файле журнала с ротацией.	
modbus.get	Читает данные Modbus.	Modbus
net.dns	Проверяет состояние службы DNS.	Сеть
net.dns.perf	Проверяет производительность службы DNS.	
net.dns.record	Выполняет DNS-запрос.	
net.if.collisions	Количество коллизий вне окна.	
net.if.discovery	Список сетевых интерфейсов.	
net.if.in	Статистика входящего трафика на сетевом интерфейсе.	
net.if.out	Статистика исходящего трафика на сетевом интерфейсе.	
net.if.total	Суммарная статистика входящего и исходящего трафика на сетевом интерфейсе.	
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.	
net.tcp.service.perf	Проверяет производительность TCP-службы.	
net.tcp.socket.count	Возвращает количество TCP-сокетов, соответствующих параметрам.	
net.udp.listen	Проверяет, находится ли этот UDP-порт в состоянии LISTEN.	
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.	
proc.cpu.util	Процент использования CPU процессом.	Процессы
proc.get	Список процессов ОС и их параметров.	
proc.mem	Память, используемая процессом, в байтах.	
proc.num	Количество процессов.	
sensor	Показания аппаратного датчика.	Датчики
system.boottime	Время загрузки системы.	Система
system.cpu.discovery	Список обнаруженных CPU/ядер CPU.	
system.cpu.intr	Прерывания устройств.	
system.cpu.load	Нагрузка CPU.	
system.cpu.num	Количество CPU.	
system.cpu.switches	Количество переключений контекста.	
system.cpu.util	Процент использования CPU.	

Ключ элемента данных	Описание	Группа элементов данных
system.hostname	Имя узла системы.	
system.hw.chassis	Информация о шасси.	
system.hw.cpu	Информация о CPU.	
system.hw.devices	Список устройств PCI или USB.	
system.hw.macaddr	Список MAC-адресов.	
system.localtime	Системное время.	
system.run	Выполняет указанную команду на узле сети.	
system.stat	Системная статистика.	
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, имя дистрибутива, младшая и старшая версии и т. д.).	
system.sw.packages	Список установленных пакетов.	
system.sw.packages.get	Подробный список установленных пакетов.	
system.swap.in	Статистика подкачки в память (с устройства в память).	
system.swap.out	Статистика подкачки из памяти (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 непрерывной работы системы в секундах.	
system.users.num	Количество вошедших в систему пользователей.	
vfs.dev.discovery	Список блочных устройств и их тип.	Виртуальные файловые системы
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 по алгоритму UNIX cksum.	
vfs.file.contents	Получение содержимого файла.	
vfs.file.exists	Проверяет, существует ли файл.	
vfs.file.get	Возвращает информацию о файле.	
vfs.file.md5sum	Контрольная сумма MD5 файла.	
vfs.file.owner	Получает владельца файла.	
vfs.file.permissions	Возвращает 4-значную строку, содержащую восьмеричное число с правами UNIX.	
vfs.file.regexp	Извлекает строку из файла.	
vfs.file.regmatch	Находит строку в файле.	
vfs.file.size	Размер файла.	
vfs.file.time	Информация о времени файла.	
vfs.fs.discovery	Список смонтированных файловых систем с их типом и параметрами монтирования.	
vfs.fs.get	Список смонтированных файловых систем с их типом, доступным дисковым пространством, статистикой inode и параметрами монтирования.	
vfs.fs.inode	Количество inode или их процент.	
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	Получает содержимое веб-страницы.	Веб-мониторинг
web.page.perf	Время загрузки полной веб-страницы.	
web.page.regexp	Находит строку на веб-странице.	
agent.hostmetadata	Метаданные узла сети агента.	Zabbix
agent.hostname	Имя узла сети агента.	
agent.ping	Проверка доступности агента.	
agent.variant	Вариант Zabbix агент (Zabbix агент или Zabbix агент 2).	
agent.version	Версия Zabbix агент.	
zabbix.stats	Возвращает набор внутренних метрик Zabbix сервера или прокси. Если указаны `ip` и `port`, метрики извлекаются с удаленного экземпляра; в противном случае — с локального экземпляра.	

Ключ элемента данных	Описание	Группа элементов данных
zabbix.stats	Возвращает количество отслеживаемых элементов данных в очереди, которые задерживаются на Zabbix сервере или прокси. Если указаны `ip` и `port`, метрики извлекаются с удаленного экземпляра; в противном случае — с локального экземпляра.	

Поддерживаемые платформы

Если иное не указано в сведениях об элементе данных, элементы данных агента (и все параметры) поддерживаются на:

- **Linux**
- **FreeBSD**
- **Solaris**
- **HP-UX**
- **AIX**
- **MacOS X**
- **OpenBSD**
- **NetBSD**

Многие элементы данных агента также поддерживаются в **Windows**. Подробности смотрите на странице Элементы данных агента Windows.

Подробности ключа элемента данных

Параметры без угловых скобок являются обязательными. Параметры, отмеченные угловыми скобками **< >**, являются необязательными.

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 количество 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) или *skip* — пропустить обработку старых данных (влияет только на вновь созданные элементы данных);
- **output** — необязательный шаблон форматирования вывода. Управляющая последовательность **\0** заменяется совпавшей частью текста (от первого символа, где начинается совпадение, до символа, где совпадение заканчивается), а управляющая последовательность **\N** (где N=1...9) заменяется N-й совпавшей группой (или пустой строкой, если N превышает количество 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 анализировать самые новые строки в пределах "maxdelay" секунд. Перед использованием прочитайте примечания о maxdelay!
- **options** — дополнительные параметры:
mtime-noreread — неуникальные записи, перечитывать только при изменении размера файла (игнорировать изменение времени модификации). (Этот параметр устарел начиная с 5.0.2, поскольку теперь mtime игнорируется.)
- **persistent dir** (только в zabbix_agentd в Unix-системах; не поддерживается в Zabbix agent 2) — абсолютный путь к каталогу, где следует хранить persistent-файлы. См. также дополнительные примечания о persistent-файлах.

Комментарии:

- Элемент данных должен быть настроен как активная проверка.
- Если файл отсутствует или права доступа не позволяют получить к нему доступ, элемент данных становится неподдерживаемым.
- Если `output` оставлен пустым, возвращается вся строка, содержащая совпавший текст. Обратите внимание, что все типы глобальных регулярных выражений, кроме 'Result is TRUE', всегда возвращают всю совпавшую строку, а 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] #
```

Пример использования параметра `output` для переписывания 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 отслеживаемом 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, которое агент будет анализировать (не может превышать 10000). Значение по умолчанию — 10*'MaxLinesPerSecond' в zabbix_agentd.conf.
- **mode** — возможные значения: *all* (по умолчанию) или *skip* — пропустить обработку 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 обеспечить анализ самых последних строк в пределах "maxdelay" секунд. Перед использованием прочитайте примечания о maxdelay!
- **options** — дополнительные параметры:
mtime-noreread — неуникальные записи, перечитывать только при изменении размера файла (игнорировать изменение времени модификации). (Этот параметр устарел начиная с 5.0.2, поскольку теперь mtime игнорируется.)
- **persistent dir** (только в zabbix_agentd в Unix-системах; не поддерживается в Zabbix agent 2) — абсолютный путь к каталогу, где следует хранить 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 поэтому зависят от интервала обновления элемента данных.
- Если файл отсутствует или права доступа не позволяют получить к нему доступ, элемент данных становится неподдерживаемым.

logrt[file regexp,<regexp>,<encoding>,<maxlines>,<mode>,<output>,<maxdelay>,<options>,<persistent dir>]

 Мониторинг ротируемого лог-файла.
 Возвращаемое значение: *Log*.
 См. поддерживаемые платформы.

Параметры:

- **file regexp** — абсолютный путь к файлу, где имя файла задаётся с помощью регулярного выражения. Обратите внимание, что регулярное выражение применяется только к имени файла и не должно соответствовать имени целиком (например, /path/to/agent будет соответствовать zabbix_agentd.log).
- **regexp** — регулярное выражение, описывающее требуемый шаблон содержимого.
- **encoding** — идентификатор кодовой страницы.
- **maxlines** — максимальное количество 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) или *skip* — пропустить обработку старых данных (влияет только на вновь созданные элементы данных).

- **output** — необязательный шаблон форматирования вывода. Управляющая последовательность **\0** заменяется совпавшей частью текста (от первого символа, где начинается совпадение, до символа, где совпадение заканчивается), а управляющая последовательность **\N** (где N=1...9) заменяется N-й совпавшей группой (или пустой строкой, если N превышает количество захваченных групп).
- **maxdelay** — максимальная задержка в секундах. Тип: float. Значения: 0 — (по умолчанию) никогда не игнорировать строки лог-файла; > 0.0 — игнорировать более старые строки, чтобы анализировать самые новые строки в пределах ”maxdelay” секунд. Перед использованием прочитайте примечания о maxdelay!
- **options** — тип ротации лог-файла и другие параметры. Возможные значения:
rotate (по умолчанию),
copytruncate — обратите внимание, что *copytruncate* нельзя использовать вместе с *maxdelay*. В этом случае *maxdelay* должен быть равен 0 или не указан; см. примечания о copytruncate,
mtime-reread — неуникальные записи, перечитывать при изменении времени модификации или размера (используется по умолчанию, если ни один параметр `mtime-*` явно не задан),
mtime-noreread — неуникальные записи, перечитывать только при изменении размера (игнорировать изменение времени модификации).
- **persistent dir** (только в zabbix_agentd в Unix-системах; не поддерживается в Zabbix agent 2) — абсолютный путь к каталогу, где следует хранить persistent-файлы. См. также дополнительные примечания о persistent-файлах.

Комментарии:

- Элемент данных должен быть настроен как активная проверка.
- Ротация логов основана на времени последней модификации файлов.
- Обратите внимание, что logrt предназначен для работы с одним текущим активным лог-файлом и несколькими другими совпадающими неактивными ротированными файлами. Если, например, каталог содержит много активных лог-файлов, для каждого из них следует создать отдельный элемент данных logrt. В противном случае, если один элемент данных logrt обрабатывает слишком много файлов, это может привести к исчерпанию памяти и сбою мониторинга.
- Если `output` оставлен пустым, возвращается вся строка, содержащая совпавший текст. Обратите внимание, что все типы глобальных регулярных выражений, кроме ’Result is TRUE’, всегда возвращают всю совпавшую строку, и параметр `output` игнорируется.
- Извлечение содержимого с помощью параметра `output` выполняется на агенте.
- В параметре `file regexp` путь к каталогу логов и регулярное выражение имени лог-файла должны быть разделены правильным разделителем каталогов:
  - В Windows разделителем должна быть обратная косая черта (\); прямые косые черты могут допускаться в других позициях, кроме той, которая разделяет путь к каталогу логов и регулярное выражение имени лог-файла (см. примеры ниже).
  - В других системах это должна быть прямая косая черта (/).

Примеры для Windows:

```
logrt["c:/dir1/dir2/dir3\filename.*\.log","pattern_to_match"] #                                -      "c:/dir
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 лога:

```
logrt[/app1/^test.*log$,"task run [0-9.]+ sec, processed ([0-9]+) records, [0-9]+ errors",,,,\1] #
```

Пример использования параметра `output` для переписывания записи лога 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 отслеживаемом журнале, для которого используется ротация.
 Возвращаемое значение: *Integer*.
 См. поддерживаемые платформы.

Параметры:

- **file regexp** — абсолютный путь к файлу, где имя файла задано с помощью регулярного выражения. Обратите внимание, что регулярное выражение применяется только к имени файла и не обязано соответствовать имени целиком (например, /path/to/agent будет соответствовать zabbix_agentd.log).
- **regexp** — регулярное выражение, описывающее требуемый шаблон.
- **encoding** — идентификатор кодовой страницы.
- **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) или *skip* — пропустить обработку 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 обеспечить анализ самых новых строк в пределах "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 равен 0 или не должен быть указан; см. примечания о 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 Unix-системах; не поддерживается в Zabbix агент 2) — абсолютный путь к каталогу, где следует хранить persistent-файлы. См. также дополнительные примечания о persistent-файлах.

Комментарии:

- Элемент данных должен быть настроен как активная проверка.
- Совпавшие строки подсчитываются среди новых строк с момента последней проверки журнала агентом, и поэтому зависят от интервала обновления элемента данных.
- Ротация журнала определяется по времени последней модификации файлов.
- В параметре `file regexp` путь к каталогу журнала и регулярное выражение для имени файла журнала должны быть разделены правильным разделителем каталогов:
  - В Windows разделителем должна быть обратная косая черта (\); прямые косые черты могут допускаться в других позициях, кроме той, которая разделяет путь к каталогу журнала и регулярное выражение для имени файла журнала (см. примеры ниже).
  - В других системах это должна быть прямая косая черта (/).

Примеры для Windows:

```
logrt.count["c:/dir1/dir2/dir3\filename.*\.log","pattern_to_match"] #
logrt.count["//example.com/share/dir1/dir2/dir3\filename.*\.log","pattern_to_match"] #
```

Примеры для других систем:

```
logrt.count["/home/zabbix/logs/.*","^logfile[0-9]{1,3}$",,100] #
logrt.count["/home/user/^logfile_.*_[0-9]{1,3}$","pattern_to_match","UTF-8",100] #
```

modbus.get[endpoint,<slave id>,<function>,<address>,<count>,<type>,<endianness>,<offset>]

 Считывает данные Modbus.
 Возвращаемое значение: *строка JSON*.
 Поддерживаемые платформы: Linux, Windows.

Параметры:

- **endpoint** — конечная точка, заданная в формате `protocol://connection_string`;
- **slave id** — ID ведомого устройства;
- **function** — функция Modbus;
- **address** — адрес первого регистра, coil или input;
- **count** — количество записей для чтения;
- **type** — тип данных;
- **endianness** — конфигурация порядка байтов;
- **offset** — количество регистров, начиная с 'address', результаты которых будут отброшены.

В Windows этот элемент данных обрабатывает UNC-пути в параметре **endpoint**, что может представлять угрозу 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 агент 2) — IP-адрес DNS-сервера (оставьте пустым для DNS-сервера по умолчанию);
- **name** — DNS-имя для запроса;
- **type** — тип записи для запроса (по умолчанию *SOA*);

- **timeout** (игнорируется в Windows, если не используется Zabbix агент 2) — тайм-аут запроса в секундах (по умолчанию 1 секунда);
- **count** (игнорируется в Windows, если не используется Zabbix агент 2) — количество попыток выполнения 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 Zabbix агентом в Windows, Zabbix агентом 2 во всех ОС), *HINFO*, *MINFO*, *TXT*, *SRV*.
- Для обратных DNS-запросов (когда `type` установлен в *PTR*) можно указывать DNS-имя как в обратном, так и в обычном формате (см. примеры ниже). Обратите внимание, что при запросе PTR-записи DNS-имя фактически является IP-адресом.
- Интернационализированные доменные имена не поддерживаются; используйте 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.0.0.2.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.8.0.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 DNS-службы.
 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 агент 2) — IP-адрес DNS-сервера (оставьте пустым для DNS-сервера по умолчанию);
- **name** — DNS-имя для запроса;
- **type** — тип записи для запроса (по умолчанию *SOA*);
- **timeout** (игнорируется в Windows, если не используется Zabbix агент 2) — тайм-аут запроса в секундах (по умолчанию 1 секунда);
- **count** (игнорируется в Windows, если не используется Zabbix агент 2) — количество попыток выполнения 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 Zabbix агентом в Windows, Zabbix агентом 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 обычном формате (см. примеры ниже). Обратите внимание, что при запросе PTR-записи DNS-имя фактически является IP-адресом.
- Интернационализированные доменные имена не поддерживаются; используйте имена, закодированные в IDNA.
- Элемент данных возвращает время отклика вместо 0, когда DNS-сервер отвечает с 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.0.0.2.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.8.0.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.

Zabbix агент 2 также предоставляет net.dns.get, который предлагает дополнительные возможности, такие как большее количество типов записей и более гибкое управление DNS-мониторингом. Возвращаемое значение: символьная строка с требуемым типом информации.
 См. поддерживаемые платформы.

Параметры:

- **ip** (игнорируется в Windows, если не используется Zabbix агент 2) — IP-адрес DNS-сервера (оставьте пустым для DNS-сервера по умолчанию);
- **name** — DNS-имя для запроса;
- **type** — тип записи, который необходимо запросить (по умолчанию *SOA*);
- **timeout** (игнорируется в Windows, если не используется Zabbix агент 2) — тайм-аут запроса в секундах (по умолчанию 1 секунда);
- **count** (игнорируется в Windows, если не используется Zabbix агент 2) — количество попыток выполнения 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 Zabbix агентом в Windows и Zabbix агентом 2 во всех ОС), *HINFO*, *MINFO*, *TXT*, *SRV*.
- Для обратных DNS-запросов (когда type установлен в *PTR*) можно указать DNS-имя в обратном или обычном формате (см. примеры ниже). Обратите внимание, что при запросе PTR-записи DNS-имя фактически является IP-адресом.
- Интернационализированные доменные имена 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 вне окна.
 Возвращаемое значение: *Целое число*.
 Поддерживаемые платформы: Linux, FreeBSD, Solaris, AIX, MacOS X, OpenBSD, NetBSD. В NetBSD требуются привилегии root.

Параметр:

- **if** — имя сетевого интерфейса

net.if.discovery

 Список сетевых интерфейсов. Используется для низкоуровневого обнаружения.
 Возвращаемое значение: *строка JSON*.
 Поддерживаемые платформы: Linux, FreeBSD, Solaris, HP-UX, AIX, OpenBSD, NetBSD, Windows.

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; 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 FIFO-буфера
frame — количество ошибок кадрирования пакетов
compressed — количество сжатых пакетов, полученных драйвером устройства
multicast — количество кадров multicast, полученных драйвером устройства

Комментарии:

- Вы можете использовать этот ключ с шагом предобработки *Change per second*, чтобы получить статистику байт 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 подробную информацию об интерфейсах 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: *Целое число*.
 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; 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 несущей, обнаруженных драйвером устройства
compressed — количество сжатых пакетов, переданных драйвером устройства

Комментарии:

- Вы можете использовать этот ключ с шагом предобработки *Изменение в секунду*, чтобы получить статистику в байтах в секунду.
- Режим *dropped* поддерживается только в Linux и HP-UX.
- Режимы *overruns*, *collision*, *carrier*, *compressed* поддерживаются только в Linux.
- В HP-UX этот элемент данных не предоставляет подробную информацию об интерфейсах loopback (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: *Целое число*.
 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; 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
compressed — количество сжатых пакетов, переданных или полученных драйвером устройства

Комментарии:

- Вы можете использовать этот ключ с шагом предобработки *Изменение в секунду*, чтобы получить статистику в байтах в секунду.
- Режим *dropped* поддерживается только в Linux и HP-UX. Отброшенные пакеты поддерживаются только в том случае, если на вашей платформе для отброшенных пакетов работают оба ключа `net.if.in` и `net.if.out`.
- Режимы *overruns*, *collision*, *compressed* поддерживаются только в Linux.
- В HP-UX этот элемент данных не предоставляет подробную информацию по loopback-интерфейсам (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 прослушиваемых TCP-сокетах по возможности 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 — подключение невозможно; 1 — подключение возможно.
 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 лог-файлах системных демонов (обычно журналируются SMTP- и SSH-сеансы).

Пример:

```
net.tcp.port[,80] #                                    -    ,              80
```

net.tcp.service[service,<ip>,<port>]

 Проверяет, запущен ли сервис и принимает ли он TCP-соединения.
 Возвращаемые значения: 0 — сервис недоступен; 1 — сервис запущен.
 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:

- Эти проверки могут приводить к появлению дополнительных сообщений в лог-файлах системных демонов (обычно журналируются SMTP- и SSH-сеансы).
- Проверка зашифрованных протоколов (например, IMAP на порту 993 или POP на порту 995) в настоящее время не поддерживается. В качестве обходного решения используйте для таких проверок `net.tcp.port[]`.
- Проверка LDAP и HTTPS в Windows поддерживается только Zabbix агент 2.
- Проверка telnet ищет приглашение для входа в систему (':' в конце).

Пример:

```
net.tcp.service[ftp,,45] #                                  FTP-       TCP-    45
```

net.tcp.service.perf[service,<ip>,<port>]

 Проверяет производительность TCP-сервиса.
 Возвращаемые значения: *Float* (0 — сервис недоступен; секунды — 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 подобных проверок.
- Проверка telnet ищет приглашение для входа в систему (':' в конце).

Пример:

```
net.tcp.service.perf[ssh] #                                                 SSH-
```

net.tcp.socket.count[<laddr>,<lport>,<raddr>,<rport>,<state>]

 Возвращает количество TCP-сокетов, соответствующих параметрам.
 Возвращаемое значение: *Integer*.
 Поддерживаемые платформы: Linux, Windows.

Параметры:

- **laddr** — локальный IPv4/6-адрес или подсеть CIDR;
- **lport** — номер локального порта или имя сервиса;
- **raddr** — удалённый IPv4/6-адрес или подсеть CIDR;
- **rport** — номер удалённого порта или имя сервиса;
- **state** — 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[port]

 Проверяет, находится ли этот 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:

- **port** — 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-службы.
 Возвращаемые значения: *Float* (0 — служба недоступна; секунды — количество секунд, затраченных на ожидание ответа от службы).
 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.perf[ntp] #                                   NTP
```

net.udp.socket.count[<laddr>,<lport>,<raddr>,<rport>,<state>]

 Возвращает количество UDP-сокетов, соответствующих параметрам.
 Возвращаемое значение: *Целое число*.
 Поддерживаемые платформы: Linux, Windows.

Параметры:

- **laddr** — локальный IPv4/6-адрес или подсеть CIDR;
- **lport** — номер локального порта или имя сервиса;
- **raddr** — удалённый IPv4/6-адрес или подсеть CIDR;
- **rport** — номер удалённого порта или имя сервиса;
- **state** — состояние соединения (*established*, *unconn*).

Пример:

```
net.udp.socket.count[,,,,established] #             UDP-            connected
```

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 *все процессы*);
- **user** — имя пользователя (по умолчанию *все пользователи*);
- **type** — тип использования CPU: *total* (по умолчанию), *user* или *system*;
- **cmdline** — фильтр 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 задействующим два ядра, составит 200%.
- Данные об использовании CPU процессом собираются сборщиком, который поддерживает максимум 1024 уникальных запроса (по имени, пользователю и командной строке). Запросы, к которым не было обращений в течение последних 24 часов, удаляются из сборщика.
- Если параметр zone установлен в *current* (или используется значение по умолчанию), а агент был скомпилирован в Solaris без поддержки зон, но работает на более новой версии Solaris, где зоны поддерживаются, тогда агент вернёт NOTSUPPORTED (агент не может ограничить результаты только текущей зоной). Однако в этом случае *all* поддерживается.

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 *все процессы*);
- **user** — имя пользователя (по умолчанию *все пользователи*);
- **cmdline** — фильтр по командной строке (это регулярное выражение). Этот параметр не поддерживается в Windows; на других платформах он не поддерживается, если для mode установлено 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 не может быть получено, 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* — если mode имеет значение *max*, *min*, *sum*; *Float* — если mode имеет значение *avg*
 Поддерживаемые платформы: Linux, FreeBSD, Solaris, AIX, OpenBSD, NetBSD.

Параметры:

- **name** — имя процесса (по умолчанию — *все процессы*);
- **user** — имя пользователя (по умолчанию — *все пользователи*);
- **mode** — возможные значения: *avg*, *max*, *min* или *sum* (по умолчанию);
- **cmdline** — фильтр по командной строке (это регулярное выражение);
- **memtype** — тип памяти, используемой процессом

Комментарии:

- Параметр memtype поддерживается только в Linux, FreeBSD, Solaris[6], AIX.
- Когда несколько процессов используют общую память, суммарный объём памяти, используемой процессами, может приводить к большим, нереалистичным значениям.

См. примечания по выбору процессов с помощью параметров name и cmdline (только для Linux).

Если этот элемент данных вызывается из командной

строки и содержит параметр командной строки (например, при использовании режима тестирования агента: `zabbix_agentd -t proc.mem[,,,apache2]`), будет учтён один дополнительный процесс, так как агент будет учитывать сам себя.

Примеры:

```
proc.mem[,root] #    ,                 ,                    "root"
proc.mem[zabbix_server,zabbix] #    ,                  zabbix_server,                    zabbix
proc.mem[,oracle,max,oracleZABBIX] #    ,             ,                    ,            Oracle        ora
```

proc.num[<name>,<user>,<state>,<cmdline>,<zone>]

 Количество процессов.
 Возвращаемое значение: *Целое число*.
 Поддерживаемые платформы: Linux, FreeBSD, Solaris[6], HP-UX, AIX, OpenBSD, NetBSD, Windows.

Параметры:

- **name** — имя процесса (по умолчанию *все процессы*);
- **user** — имя пользователя (по умолчанию *все пользователи*);
- **state** — возможные значения:
all (по умолчанию),
disk — непрерываемый сон,
run — выполняется,
sleep — прерываемый сон,
trace — остановлен,
zomb — зомби;
- **cmdline** — фильтр 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 режима тестирования агента: `zabbix_agentd -t proc.num[,,,apache2]`), будет подсчитан один дополнительный процесс, так как агент посчитает сам себя.
- Если параметр `zone` установлен в `current` (или используется значение по умолчанию), а агент был скомпилирован в Solaris без поддержки зон, но работает в более новой версии Solaris, где зоны поддерживаются, тогда агент вернет NOTSUPPORTED (агент не может ограничить результаты только текущей зоной). Однако в этом случае поддерживается *all*.
- См. примечания о выборе 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        oracl
```

sensor[device,sensor,<mode>]

 Показание аппаратного датчика.
 Возвращаемое значение: *Float*.
 Поддерживаемые платформы: Linux, OpenBSD.

Параметры:

- **device** — имя устройства;
- **sensor** — имя датчика;
- **mode** — возможные значения: *avg*, *max* или *min*.

Комментарии:

- Считывает /proc/sys/dev/sensors в Linux 2.4.
- Считывает /sys/class/hwmon в Linux 2.6+. См. более подробное описание элемента данных sensor в Linux.
- Считывает MIB *hw.sensors* в OpenBSD.
- В Linux 2.4, если указан mode, то device и sensor обрабатываются как регулярные выражения.
- В Linux 2.6+ и OpenBSD, либо если mode не указан, device и sensor обрабатываются как буквальные значения.

Примеры:

```
sensor[w83781d-i2c-0-2d,temp1]
sensor[cpu0,temp0] #the temperature of one CPU
sensor["cpu[0-2]$",temp,avg] #the average temperature of the first three CPUs
```

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 низкоуровневого обнаружения.
 Возвращаемое значение: *JSON string*.
 Смотрите поддерживаемые платформы.

system.cpu.intr

 Прерывания устройств.
 Возвращаемое значение: *Целое число*.
 Поддерживаемые платформы: Linux, FreeBSD, Solaris, AIX, OpenBSD, NetBSD.

system.cpu.load[<cpu>,<mode>]

 Загрузка CPU.
 Возвращаемое значение: *Float*.
 См. поддерживаемые платформы.

Параметры:

- **cpu** — возможные значения: *all* (по умолчанию) или *percpu* (общая загрузка, делённая на количество CPU в сети);
- **mode** — возможные значения: *avg1* (среднее за одну минуту, по умолчанию), *avg5* или *avg15*.

Пример:

```
system.cpu.load[,avg5]
```

system.cpu.num[<type>]

 Количество процессоров.
 Возвращаемое значение: *Целое число*.
 Поддерживаемые платформы: Linux, FreeBSD, Solaris, HP-UX, AIX, MacOS X, OpenBSD, NetBSD, Windows.

Параметр:

- **type** — возможные значения: *online* (по умолчанию) или *max*

Параметр типа *max* 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 ЦП.
 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 более поздних версиях, Solaris, AIX.
- Параметр типа *interrupt* поддерживается только в Linux 2.6 и более поздних версиях, FreeBSD, OpenBSD.
- Параметры типа *softirq*, *steal*, *guest*, *guest_nice* поддерживаются только в Linux 2.6 и более поздних версиях.
- 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 сети 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 сети до первой точки, полную строку для имён без точек), *fqdn* (возвращает полное доменное имя — Fully Qualified Domain Name);
- **transform** — возможные значения: *none* (по умолчанию) или *lower* (преобразовать в нижний регистр).

Значение получается из nodename в выводе 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[<info>]

 Информация о шасси.
 Возвращаемое значение: *Строка*.
 Поддерживаемые платформы: Linux.

Параметр:

- **info** — возможные значения: *full* (по умолчанию), *model*, *serial*, *type* или *vendor*

Комментарии:

- Этот ключ элемента данных зависит от доступности таблицы SMBIOS.
- Будет предпринята попытка прочитать таблицу DMI из sysfs; если доступ к sysfs завершится неудачей, будет предпринята попытка чтения напрямую из памяти.
- Требуются **права root**, поскольку значение получается путем чтения из sysfs или памяти.

Пример:

```
system.hw.chassis[full] → Hewlett-Packard HP Pro 3010 Small Form Factor PC CZXXXXXXXX Desktop
```

system.hw.cpu[<cpu>,<info>]

 Информация о CPU.
 Возвращаемое значение: *String* или *Integer*.
 Поддерживаемые платформы: Linux.

Параметры:

- **cpu** — <*номер CPU*> или *all* (по умолчанию);
- **info** — возможные значения: *full* (по умолчанию), *curfreq*, *maxfreq*, *model* или *vendor*.

Комментарии:

- Собирает информацию из /proc/cpuinfo и /sys/devices/system/cpu/[cpunum]/cpufreq/cpuinfo_max_freq.
- Если указан номер CPU и *curfreq* или *maxfreq*, возвращается числовое значение (Гц).

Пример:

```
system.hw.cpu[0,vendor] → AuthenticAMD
```

system.hw.devices[<type>]

 Список устройств PCI или USB.
 Возвращаемое значение: *Текст*.
 Поддерживаемые платформы: Linux.

Параметр:

- **type** — *pci* (по умолчанию) или *usb*

Возвращает вывод утилиты lspci или lsusb (выполняемой 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: *Строка*.
 Поддерживаемые платформы: Linux.

Параметры:

- **interface** — *all* (по умолчанию) или регулярное выражение;
- **format** — *full* (по умолчанию) или *short*

Комментарии:

- Выводит список MAC-адресов интерфейсов, имя которых соответствует указанному регулярному выражению `interface` (*all* выводит список для всех интерфейсов).
- Если `format` указан как *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* — если `type` имеет значение *utc*; *String* — если `type` имеет значение *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, или *local* — время в формате 'yyyy-mm-dd,hh:mm:ss.nnn,+hh:mm'

Должен использоваться как пассивная проверка для Zabbix агент; в Zabbix агент 2 может быть настроен как активная проверка.

Пример:

```
system.localtime[local] #                    ,                                    *Clo
```

system.run[command,<mode>]

 Выполнить указанную команду на узле сети.
 Возвращаемое значение: *Текстовый* результат команды или 1 — при `mode`, равном *nowait* (независимо от результата команды).
 См. поддерживаемые платформы.

Параметры:

- **command** — команда для выполнения;
- **mode** — возможные значения: *wait* — ждать завершения выполнения (по умолчанию) или *nowait* — не ждать.

Комментарии:

- Этот элемент данных по умолчанию отключен. Узнайте, как включить их.
- Возвращаемое значение элемента данных представляет собой стандартный вывод вместе со стандартным выводом ошибок, сформированными командой. Проверка кода завершения не выполняется.
- Для корректной обработки возвращаемое значение команды должно иметь тип данных `text`. Также допускается пустой результат.
- Возвращаемое значение ограничено 16 МБ (включая конечные пробельные символы, которые усекаются); также применяются ограничения базы данных.
- См. также: Выполнение команд.

Пример:

```
system.run[ls -l /] #
```

system.stat[ресурс,<тип>]

 Системная статистика.
 Возвращаемое значение: *Целое число* или *число с плавающей точкой*.
 Поддерживаемые платформы: AIX.

Параметры:

- **ent** — количество процессорных единиц, выделенных на ресурс (число с плавающей точкой);
- **kthr,<тип>** — информация о состояниях потоков ядра:
r — среднее количество запускаемых потоков ядра (число с плавающей точкой)
b — среднее количество потоков ядра, помещённых в очередь ожидания Менеджера Виртуальный Памяти (число с плавающей точкой)
- **memory,<тип>** — информация об использовании виртуальной и реальной памяти:
avm — активные виртуальные страницы (целое число)
fre — размер свободного списка (целое число)
- **page,<тип>** — информация об ошибках страниц и активности страниц:
fi — количество загрузок страниц в секунду (число с плавающей точкой)
fo — количество выгрузок страниц в секунду (число с плавающей точкой)
pi — страницы, загруженные из пространства подкачки (число с плавающей точкой)
po — страницы, выгруженные в пространство подкачки (число с плавающей точкой)
fr — освобождённые страницы (замена страниц) (число с плавающей точкой)
sr — количество страниц, сканированных алгоритмом замены страниц (число с плавающей точкой)

- **faults,<тип>** — соотношение ловушек и прерываний:
in — прерывания устройства (число с плавающей точкой)
sy — системные вызовы (число с плавающей точкой)
cs — переключения контекста потока ядра (число с плавающей точкой)
- **cpu,<тип>** — распределение процессорного времени в процентах:
us — пользовательское время (число с плавающей точкой)
sy — системное время (число с плавающей точкой)
id — время простоя (число с плавающей точкой)
wa — время простоя, в ходе которого система обрабатывала запросы I/O диска/NFS (число с плавающей точкой)
pc — количество использованных физических процессоров (число с плавающей точкой)
ec — процент потреблённой выделенной ёмкости (число с плавающей точкой)
lbusy — отображает утилизацию в процентах для логических процессор(ов), которая произошла при выполнении уровней пользователя и системы (число с плавающей точкой)
app — отображает количество доступных физических процессоров в общем пуле (число с плавающей точкой)
- **disk,<тип>** — статистика о дисках:
bps — показывает объём данных, переданных (прочитанных или записанных) на диск в байтах в секунду (целое число)
tps — показывает количество передач за секунду, которые произошли на физическом диске/ленте (число с плавающей точкой)

Комментарии:

- Обратите внимание на следующие ограничения в этих элементах данных:
 `system.stat[cpu,app]` — поддерживается только на AIX LPAR с типом «Shared»
 `system.stat[cpu,ec]` — поддерживается на AIX LPAR с типом «Shared» и «Dedicated» («Dedicated» всегда возвращает 100 (процентов))
 `system.stat[cpu,lbusy]` — поддерживается только на AIX LPAR с типом «Shared»
 `system.stat[cpu,pc]` — поддерживается на AIX LPAR с типом «Shared» и «Dedicated»
 `system.stat[ent]` — поддерживается на AIX LPAR с типом «Shared» и «Dedicated»

system.sw.arch

 Информация об архитектуре программного обеспечения.
 Возвращаемое значение: *String*.
 См. поддерживаемые платформы.

Информация получается с помощью функции `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 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 Linux-системах, которые его поддерживают, или `/etc/issue.net` (*name*);
- раздела 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, имя 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: *Текст*.
 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 указанному регулярному выражению (*all* выводит все).
- Поддерживаемые менеджеры пакетов (выполняемая команда):
  - dpkg (dpkg --get-selections)
  - pkgtool (ls /var/log/packages)
  - rpm (rpm -qa)
  - pacman (pacman -Q)
portage
- Если `format` указан как *full*, пакеты группируются по менеджерам пакетов (каждый менеджер на отдельной строке, начинающейся с его имени в квадратных скобках).
- Если `format` указан 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. подробнее).

system.swap.in[<device>,<type>]

 Статистика подкачки в память (с устройства в память).
 Возвращаемое значение: *Целое число*.
 Поддерживаемые платформы: Linux, FreeBSD, OpenBSD.

Параметры:

- **device** — укажите устройство, используемое для подкачки (только Linux), или *all* (по умолчанию);
- **type** — возможные значения: *count* (количество операций подкачки в память, по умолчанию на платформах, отличных от Linux), *sectors* (секторы, загруженные в память), или *pages* (страницы, загруженные в память, по умолчанию в 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 в Linux.

Пример:

```
system.swap.in[,pages]
```

system.swap.out[<device>,<type>]

 Статистика выгрузки в swap (из памяти на устройство).
 Возвращаемое значение: *Целое число*.
 Поддерживаемые платформы: Linux, FreeBSD, OpenBSD.

Параметры:

- **device** — укажите устройство, используемое для подкачки (только Linux), или *all* (по умолчанию);
- **type** — возможные значения: *count* (количество выгрузок в swap, по умолчанию на платформах, отличных от Linux), *sectors* (секторы, выгруженные в swap) или *pages* (страницы, выгруженные в swap, по умолчанию в 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 в Linux.

Пример:

```
system.swap.out[,pages]
```

system.swap.size[<device>,<type>]

 Размер пространства подкачки в байтах или в процентах от общего объёма.
 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) или *used* (используемое пространство подкачки).

Комментарии:

- Обратите внимание, что *pfree*, *pused* не поддерживаются в Windows, если размер подкачки равен 0.
- Если устройство не указано, агент Zabbix будет учитывать только устройства подкачки (файлы), физическая память будет игнорироваться. 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*.
 См. поддерживаемые платформы.

Комментарии:

- В UNIX значение этого элемента данных получается с помощью системного вызова uname().
- В Windows этот элемент данных 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 непрерывной работы системы в секундах.
 Возвращаемое значение: *Integer*.
 Поддерживаемые платформы: Linux, FreeBSD, Solaris, AIX, MacOS X, OpenBSD, NetBSD, Windows.

В настройке элемента данных используйте единицы **s** или **uptime**, чтобы получить значения в удобочитаемом виде.

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 типа. Используется для низкоуровневого обнаружения.
 Возвращаемое значение: *JSON string*.
 Поддерживаемые платформы: Linux.

vfs.dev.read[<device>,<type>,<mode>]

 Статистика чтения диска.
 Возвращаемое значение: *Integer* — с `type` в *sectors*, *operations*, *bytes*; *Float* — с `type` в *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: секторов, операций, байт в секунду);
- **mode** — возможные значения: *avg1* (среднее за одну минуту, по умолчанию), *avg5* или *avg15*. Этот параметр поддерживается только при `type` в: sps, ops, bps.

Комментарии:

- При использовании интервала обновления в 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 параметра 'type' по умолчанию для разных ОС:
  - *AIX* — operations
  - *FreeBSD* — bps
  - *Linux* — sps
  - *OpenBSD* — operations
  - *Solaris* — bytes
- *sps*, *ops* и *bps* на поддерживаемых платформах ограничены 1024 устройствами (1023 отдельных и одно для *all*).

Пример:

```
vfs.dev.read[,operations]
```

vfs.dev.write[<device>,<type>,<mode>]

 Статистика записи на диск.
 Возвращаемое значение: *Integer* — при `type` в *sectors*, *operations*, *bytes*; *Float* — при `type` в *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: секторов, операций, байт в секунду);
- **mode** — возможные значения: *avg1* (среднее за одну минуту, по умолчанию), *avg5* или *avg15*. Этот параметр поддерживается только при `type` в: sps, ops, bps.

Комментарии:

- При использовании интервала обновления в 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 параметра 'type' по умолчанию для разных ОС:
  - *AIX* - operations
  - *FreeBSD* - bps
  - *Linux* - sps
  - *OpenBSD* - operations
  - *Solaris* - bytes
- *sps*, *ops* и *bps* на поддерживаемых платформах ограничены 1024 устройствами (1023 отдельных и одно 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 (файл, каталог, символическая ссылка), которую нужно включить; если пусто, включаются все (значение по умолчанию);
- **regex excl** - регулярное выражение, описывающее шаблон имени сущности (файл, каталог, символическая ссылка), которую нужно исключить; если пусто, ничего не исключается (значение по умолчанию);

- **types incl** - типы записей каталога для подсчета, возможные значения: *file* - обычный файл, *dir* - подкаталог, *sym* - символическая ссылка, *sock* - сокет, *bdev* - блочное устройство, *cdev* - символьное устройство, *fifo* - FIFO, *dev* - синоним для "bdev,cdev", *all* - все типы (по умолчанию), то есть "file,dir,sym,sock,bdev,cdev,fifo". Несколько типов должны быть разделены запятыми и заключены в кавычки.
- **types excl** - типы записей каталога (см. `types incl`), которые НЕ нужно подсчитывать. Если какой-либо тип записи указан и в `types incl`, и в `types excl`, записи каталога этого типа НЕ подсчитываются.
- **max depth** - максимальная глубина обхода подкаталогов:
-1 (по умолчанию) - без ограничений,
0 - без спуска в подкаталоги.
- **min size** - минимальный размер (в байтах) файла, который будет подсчитан. Файлы меньшего размера не будут подсчитываться. Можно использовать суффиксы памяти.
- **max size** - максимальный размер (в байтах) файла, который будет подсчитан. Файлы большего размера не будут подсчитываться. Можно использовать суффиксы памяти.
- **min age** - минимальный возраст (в секундах) записи каталога, которая будет подсчитана. Более новые записи не будут подсчитываться. Можно использовать суффиксы времени.
- **max age** - максимальный возраст (в секундах) записи каталога, которая будет подсчитана. Записи такого возраста и старше не будут подсчитываться (время изменения). Можно использовать суффиксы времени.
- **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 представлять риск безопасности.
- Переменные окружения, например %APP_HOME%, $HOME и %TEMP%, не поддерживаются.
- Псевдокаталоги "." и ".." никогда не подсчитываются.
- Символические ссылки никогда не используются при обходе каталогов.
- И `regex incl`, и `regex excl` применяются к файлам и каталогам при вычислении количества записей, но игнорируются при выборе подкаталогов для обхода (если `regex incl` имеет значение "(?i)^.+\.zip$" и `max depth` не задан, будут обойдены все подкаталоги, но подсчитаны будут только файлы типа zip).
- Время выполнения ограничено значением timeout в конфигурации агента (3 сек). Поскольку обход больших каталогов может занять больше времени, данные не будут возвращены, а элемент данных станет неподдерживаемым. Частичный результат подсчета возвращен не будет.
- При фильтрации по размеру только обычные файлы имеют значимые размеры. В Linux и BSD каталоги также имеют ненулевой размер (обычно несколько КБ). Устройства имеют нулевой размер, 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 неожиданностей.

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: *JSON string*.
 См. поддерживаемые платформы.

Параметры:

- **dir** - абсолютный путь к каталогу;
- **regex incl** - регулярное выражение, описывающее шаблон имени сущности (файл, каталог, символическая ссылка), которую следует включить; если пусто, включаются все (значение по умолчанию);
- **regex excl** - регулярное выражение, описывающее шаблон имени сущности (файл, каталог, символическая ссылка), которую следует исключить; если пусто, не исключается ничего (значение по умолчанию);
- **types incl** - типы записей каталога для вывода, возможные значения: *file* - обычный файл, *dir* - подкаталог, *sym* - символическая ссылка, *sock* - сокет, *bdev* - блочное устройство, *cdev* - символьное устройство, *fifo* - FIFO, *dev* - синоним для "bdev,cdev", *all* - все типы (по умолчанию), то есть "file,dir,sym,sock,bdev,cdev,fifo". Несколько типов должны быть разделены запятыми и заключены в кавычки.
- **types excl** - типы записей каталога (см. `types incl`), которые НЕ следует выводить. Если какой-либо тип записи указан и в `types incl`, и в `types excl`, записи каталога этого типа НЕ выводятся.
- **max depth** - максимальная глубина обхода подкаталогов:
-1 (по умолчанию) - без ограничений,
0 - без спуска в подкаталоги.
- **min size** - минимальный размер (в байтах) файла для включения в список. Файлы меньшего размера не будут выведены. Можно использовать суффиксы памяти.
- **max size** - максимальный размер (в байтах) файла для включения в список. Файлы большего размера не будут выведены. Можно использовать суффиксы памяти.
- **min age** - минимальный возраст (в секундах) записи каталога для включения в список. Более новые записи не будут выведены. Можно использовать суффиксы времени.

- **max age** - максимальный возраст (в секундах) записи каталога для включения в список. Записи такого возраста и старше не будут выведены (по времени изменения). Можно использовать суффиксы времени.
- **regex excl dir** - регулярное выражение, описывающее шаблон имени каталога, который следует исключить. Все содержимое каталога будет исключено (в отличие от `regex excl`)

Комментарии:

- В Windows этот элемент данных следует по UNC-путям, что может представлять угрозу безопасности.
- Переменные окружения, например %APP_HOME%, $HOME и %TEMP%, не поддерживаются.
- Псевдокаталоги ”.” и ”..” никогда не выводятся.
- Символические ссылки никогда не используются при обходе каталогов.
- И `regex incl`, и `regex excl` применяются к файлам и каталогам при формировании списка записей, но игнорируются при выборе подкаталогов для обхода (если `regex incl` равно ”(?i)^.+\.zip$” и `max depth` не задан, будут обойдены все подкаталоги, но учтены будут только файлы типа zip).
- Время выполнения ограничено значением timeout в конфигурации агент. Поскольку обход больших каталогов может занять больше времени, данные не будут возвращены, и элемент данных станет неподдерживаемым. Частичный список возвращен не будет.
- При фильтрации по размеру только обычные файлы имеют значимый размер. В Linux и BSD каталоги также имеют ненулевой размер (обычно несколько КБ). Устройства имеют нулевой размер, 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 неожиданностей.

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 сущности (файла, каталога, символической ссылки), которую следует включить; если пусто, включаются все (значение по умолчанию);
- **regex excl** — регулярное выражение, описывающее шаблон имени сущности (файла, каталога, символической ссылки), которую следует исключить; если пусто, не исключается ничего (значение по умолчанию);
- **mode** — возможные значения: *apparent* (по умолчанию) — получает видимый размер файлов, а не использование диска (аналогично `du -sb dir`), *disk* — получает использование диска (аналогично `du -s -B1 dir`). В отличие от команды du, элемент данных vfs.dir.size учитывает скрытые файлы при вычислении размера каталога (аналогично `du -sb .[^.]* *` внутри dir).
- **max depth** — максимальная глубина обхода подкаталогов: **-1** (по умолчанию) — без ограничений, **0** — без спуска в подкаталоги.
- **regex excl dir** — регулярное выражение, описывающее шаблон имени каталога, который следует исключить. Будет исключено всё содержимое каталога (в отличие от `regex excl`)

Комментарии:

- В Windows этот элемент данных следует по UNC-путям, что может представлять угрозу безопасности.
- Вычисляются только каталоги, для которых у пользователя *zabbix* есть как минимум права на чтение. Для каталогов только с правами на чтение вычисляется размер самого каталога. Для каталогов с правами на чтение и выполнение вычисляется размер вместе с содержимым.
- Для больших каталогов или медленных дисков этот элемент данных может завершиться по тайм-ауту из-за настройки Timeout в конфигурационных файлах агент и сервер/прокси. При необходимости увеличьте значения тайм-аута.
- Ограничение на размер файла зависит от поддержки больших файлов.

Примеры:

```
vfs.dir.size[/tmp,log] #calculates the size of all files in /tmp containing 'log' in their names
vfs.dir.size[/tmp,log,^.+\.old$] #calculates the size of all files in /tmp containing 'log' in their names
```

vfs.file.cksum[file,<mode>]

 Контрольная сумма файла, вычисленная алгоритмом UNIX cksum.
 Возвращаемое значение: *Integer* — если `mode` имеет значение *crc32*, *String* — если `mode` имеет значение *md5*, *sha256*.
 См. поддерживаемые платформы.

Параметры:

- **file** — полный путь к файлу;
- **mode** — *crc32* (по умолчанию), *md5* или *sha256*.

Комментарии:

- В Windows этот элемент данных поддерживает UNC-пути, что может представлять угрозу безопасности.
- Ограничение на размер файла зависит от поддержки больших файлов.

Пример:

```
vfs.file.cksum[/etc/passwd]
```

Примеры возвращаемых значений (crc32/md5/sha256 соответственно):

```
675436101
9845acf68b73991eb7fd7ee0ded23c44
ae67546e4aac995e5c921042d0cf0f1f7147703aa42bfbfb65404b30f238f2dc
```

vfs.file.contents[file,<encoding>]

 Получение содержимого файла[7].
 Возвращаемое значение: *Текст*.
 См. поддерживаемые платформы.

Параметры:

- **file** — полный путь к файлу;
- **encoding** — идентификатор кодовой страницы.

Комментарии:

- В Windows этот элемент данных поддерживает UNC-пути, что может представлять угрозу безопасности.
- Возвращаемое значение ограничено 16 МБ (включая конечные пробельные символы, которые усекаются); также применяются ограничения базы данных.
- Если файл пуст или содержит только символы LF/CR, возвращается пустая строка.
- Метка 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 — существует файл указанного типа.
 См. поддерживаемые платформы.

Параметры:

- **file** — полный путь к файлу;
- **types incl** — список включаемых типов файлов, 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 types_excl задан).
- **types excl** — список исключаемых типов файлов, возможные значения см. в types_incl (по умолчанию никакие типы не исключаются)

Комментарии:

- В Windows этот элемент данных обрабатывает UNC-пути, что может представлять угрозу безопасности.
- Несколько типов должны разделяться запятой, а весь набор должен быть заключён в двойные кавычки "".
- Если один и тот же тип указан и в <types_incl>, и в <types_excl>, файлы этого типа исключаются.
- Ограничение на размер 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 UNC-путям, что может представлять угрозу безопасности.
- Поддерживаемые типы файлов в UNIX-подобных системах: обычный файл, каталог, символическая ссылка, сокет, блочное устройство, символьное устройство, FIFO.
- Ограничение на размер файла зависит от поддержки больших файлов.

Пример:

```
vfs.file.get[/etc/passwd] #      JSON              /etc/passwd (  ,        ,         , SID, uid    .  .)
```

vfs.file.md5sum[file]

 MD5-контрольная 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 UNC-путям, что может представлять риск безопасности.
- Ограничение на размер 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>]

 Возвращает владельца файла.
 Возвращаемое значение: *Строка*.
 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 представлять угрозу 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[file]

 Возвращает 4-значную строку, содержащую восьмеричное число с правами доступа UNIX.
 Возвращаемое значение: *String*.
 Поддерживаемые платформы: Linux. Элемент данных может работать и на других UNIX-подобных платформах.

Параметры:

- **file** — полный путь к файлу

Ограничение на размер 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>]

 Получение строки в файле[7].
 Возвращаемое значение: строка, содержащая совпавшую подстроку, или значение, заданное необязательным параметром `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 совпадение заканчивается), а последовательность экранирования **\N** (где N=1...9) заменяется N-й совпавшей группой (или пустой строкой, если N превышает количество захваченных групп).

Комментарии:

- В Windows этот элемент данных поддерживает UNC-пути, что может представлять угрозу безопасности.
- Ограничение на размер 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;
- **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 поддерживает UNC-пути, что может представлять угрозу безопасности.
- Ограничение на размер 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) или *lines* (пустые строки также учитываются).

Комментарии:

- В Windows этот элемент данных следует по UNC-путям, что может представлять угрозу безопасности.
- Файл должен иметь права на чтение для пользователя *zabbix*.
- Ограничение на размер 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,
access — время последнего чтения файла,
change — время последнего изменения свойств файла

Комментарии:

- В Windows этот элемент данных поддерживает UNC-пути, что может представлять угрозу безопасности.
- Ограничение на размер 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 низкоуровневого обнаружения.
 Возвращаемое значение: *JSON string*.
 Поддерживаемые платформы: Linux, FreeBSD, Solaris, HP-UX, AIX, MacOS X, OpenBSD, NetBSD, Windows.

vfs.fs.get

 Список смонтированных файловых систем с их типом, доступным дисковым пространством, статистикой inode и параметрами монтирования. Может использоваться для низкоуровневого обнаружения.
 Возвращаемое значение: *JSON string*.
 Поддерживаемые платформы: Linux, FreeBSD, Solaris, HP-UX, AIX, MacOS X, OpenBSD, NetBSD, Windows.

Комментарии:

- Файловые системы с количеством inode, равным нулю, что возможно для файловых систем с динамическими inode (например, btrfs), также включаются в отчет.
- См. также: Обнаружение смонтированных файловых систем.

vfs.fs.inode[fs,<mode>]

 Количество inode или их процент.
 Возвращаемое значение: *Integer* — для количества; *Float* — для процентов.
 См. поддерживаемые платформы.

Параметры:

- **fs** — файловая система;
- **mode** — возможные значения: *total* (по умолчанию), *free*, *used*, *pfree* (свободно, в процентах) или *pused* (использовано, в процентах).

Если количество inode равно нулю, что возможно для файловых систем с динамическими inode (например, btrfs), значения pfree/pused будут возвращены как ”100” и ”0” соответственно.

Пример:

```
vfs.fs.inode[/,pfree]
```

vfs.fs.size[fs,<mode>]

 Дисковое пространство в байтах или в процентах от общего объема.
 Возвращаемое значение: *Integer* — для байтов; *Float* — для процентов.
 См. поддерживаемые платформы.

Параметры:

- **fs** — файловая система;
- **mode** — возможные значения: *total* (по умолчанию), *free*, *used*, *pfree* (свободно, в процентах) или *pused* (использовано, в процентах).

Комментарии:

- Если файловая система не смонтирована, возвращается размер локальной файловой системы, в которой находится точка монтирования.
- Зарезервированное пространство файловой системы учитывается и не включается 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* (используется, в процентах), *available* или *pavailable* (доступно, в процентах).

Комментарии:

- Этот элемент данных принимает три категории параметров:
  - *total* — общий объем памяти;
  - специфичные для платформы типы памяти: *active*, *anon*, *buffers*, *cached*, *exec*, *file*, *free*, *inactive*, *pinned*, *shared*, *slab*, *wired*;
  - оценки пользовательского уровня того, 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>]

 Получить содержимое веб-страницы.
 Возвращаемое значение: исходный код веб-страницы в виде текста (включая заголовки).
 См. поддерживаемые платформы.

Параметры:

- **host** — имя узла сети или URL (в виде `scheme://host:port/path`, где обязательным является только *host*). Допустимые схемы URL: *http*, *https*[4]. Если схема не указана, будет использоваться *http*. Если указан URL, параметры `path` и `port` должны быть пустыми. Указание имени пользователя/пароля при подключении к серверам, требующим аутентификации, например: `http://user:password@www.example.com`, возможно только при наличии поддержки cURL [4]. В именах узлов сети поддерживается Punycode.
- **path** — путь к HTML-документу (по умолчанию /);
- **port** — номер порта (по умолчанию 80 для HTTP)

Комментарии:

- Этот элемент данных становится неподдерживаемым, если ресурс, указанный в `host`, не существует или недоступен.
- `host` может быть именем узла сети, доменным именем, адресом IPv4 или IPv6. Но для адреса IPv6 агент Zabbix должен быть скомпилирован с включенной поддержкой IPv6.

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.perf[host,<path>,<port>]

 Время загрузки полной веб-страницы (в секундах).
 Возвращаемое значение: *Float*.
 См. поддерживаемые платформы.

Параметры:

- **host** — имя узла или URL (в виде `scheme://host:port/path`, где обязательным является только *host*). Допустимые схемы URL: *http*, *https*[4]. Если схема не указана, будет использоваться *http*. Если указан URL, `path` и `port` должны быть пустыми. Указание имени пользователя/пароля при подключении к серверам, требующим аутентификации, например: `http://user:password@www.example.com`, возможно только при наличии поддержки cURL [4]. Punycode поддерживается в именах узлов.
- **path** — путь к HTML-документу (по умолчанию /);
- **port** — номер порта (по умолчанию 80 для HTTP)

Комментарии:

- Этот элемент данных становится неподдерживаемым, если ресурс, указанный в `host`, не существует или недоступен.
- `host` может быть именем узла, доменным именем, адресом IPv4 или IPv6. Но для адреса IPv6 агент Zabbix должен быть скомпилирован с включенной поддержкой IPv6.

Примеры:

```
web.page.perf[www.example.com,index.php,80]
web.page.perf[https://www.example.com]
```

web.page.regexp[host,<path>,<port>,regexp,<length>,<output>]

 Найти строку на веб-странице.
 Возвращаемое значение: совпавшая строка или значение, заданное необязательным параметром `output`.
 См. поддерживаемые платформы.

Параметры:

- **host** — имя узла сети или URL (в виде `scheme://host:port/path`, где обязателен только *host*). Допустимые схемы URL: *http*, *https*[4]. Если схема не указана, будет использоваться *http*. Если указан URL, `path` и `port` должны быть пустыми. Указание имени пользователя/пароля при подключении к серверам, требующим аутентификации, например: `http://user:password@www.example.com`, возможно только при наличии поддержки 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 вывода. Управляющая последовательность **\0** заменяется совпавшей частью текста (от первого символа, с которого начинается совпадение, до символа, на котором совпадение заканчивается), а управляющая последовательность **\N** (где N=1...9) заменяется N-й совпавшей группой (или пустой строкой, если N превышает количество захваченных групп).

Комментарии:

- Этот элемент данных становится неподдерживаемым, если ресурс, указанный в `host`, не существует или недоступен.
- `host` может быть именем узла сети, доменным именем, адресом IPv4 или IPv6. Но для адреса IPv6 агент Zabbix должен быть скомпилирован с включенной поддержкой IPv6.
- Извлечение содержимого с использованием параметра `output` выполняется на агенте.

Примеры:

```
web.page.regexp[www.example.com,index.php,80,OK,2]
web.page.regexp[https://www.example.com,,,OK,2]
```

agent.hostmetadata

 Метаданные узла сети агента.
 Возвращаемое значение: *Строка*.
 Смотрите Поддерживаемые платформы.

Возвращает значение параметров HostMetadata или HostMetadataItem, или пустую строку, если ничего не задано.

agent.hostname

 Имя узла сети агента.
 Возвращаемое значение: *Строка*.
 Смотрите Поддерживаемые платформы.

Возвращает:

- Пассивной проверкой — имя первого узла сети, указанного в параметре Hostname в файле конфигурации агента;
- Активной проверкой — текущее значение имени хоста.

agent.ping

 Проверка доступности агента.
 Возвращаемое значение: Ничего — недоступен; 1 — доступен.
 Смотрите Поддерживаемые платформы.

Используйте триггерную функцию **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 агента Zabbix.
 Возвращаемое значение: *String*.
 См. 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 удалённого экземпляра; в противном случае — с локального экземпляра.
 Возвращаемое значение: *строка JSON*.
 См. поддерживаемые платформы.

Параметры:

- **ip** — список IP/DNS/сетевых масок серверов/прокси, к которым выполняется удалённый запрос (по умолчанию 127.0.0.1);
- **port** — порт сервера/прокси, к которому выполняется удалённый запрос (по умолчанию 10051)

Комментарии:

- Этот элемент данных возвращает выбранный набор внутренних метрик. Подробности см. в разделе Удалённый мониторинг статистики Zabbix.
- Обратите внимание, что запрос статистики будет принят только от адресов, перечисленных в параметре 'StatsAllowedIP' сервера/прокси на целевом экземпляре.

zabbix.stats[<ip>,<port>,queue,<from>,<to>]

 Возвращает количество отслеживаемых элементов данных в очереди, обработка которых задерживается на сервере Zabbix или прокси. Если указаны `ip` и `port`, метрики извлекаются из удалённого экземпляра; в противном случае — из локального экземпляра.
 Возвращаемое значение: *JSON string*.
 См. поддерживаемые платформы.

Параметры:

- **ip** - список IP/DNS/сетевых масок серверов/прокси, к которым выполняется удалённый запрос (по умолчанию 127.0.0.1);
- **port** - порт сервера/прокси, к которому выполняется удалённый 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 статистики будет принят только с адресов, перечисленных в параметре 'StatsAllowedIP' server/proxy на целевом экземпляре.

Сноски

[1] Примечание только для Linux. Агент Zabbix должен иметь доступ только на чтение к файловой системе */proc*. Патчи ядра с www.grsecurity.org ограничивают права доступа непривилегированных пользователей.

[2] `vfs.dev.read[]`, `vfs.dev.write[]`: агент Zabbix будет завершать "устаревшие" подключения к устройствам, если к значениям элементов данных не обращались 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 '0'.

[3] `vfs.dev.read[]`, `vfs.dev.write[]`: если для первого параметра используется значение по умолчанию *all*, то ключ будет возвращать сводную статистику, включая все блочные устройства, такие как sda, sdb, и их разделы (sda1, sda2, sdb3...), а также составные устройства (MD raid), основанные на этих блочных устройствах/разделах, и логические тома (LVM), основанные на этих блочных устройствах/разделах. В таких случаях возвращаемые значения следует рассматривать только как относительные величины (изменяющиеся во времени), а не как абсолютные значения.

[4] SSL (HTTPS) поддерживается только в том случае, если агент скомпилирован с поддержкой cURL. В противном случае элемент данных станет неподдерживаемым.

[5] Значения `bytes` и `errors` не поддерживаются для loopback-интерфейсов в системах Solaris вплоть до Solaris 10 6/06 включительно, так как статистика байтов, ошибок и использования не сохраняется и/или не предоставляется ядром. Однако, если вы выполняете мониторинг системы Solaris через net-snmp, значения могут возвращаться, поскольку net-snmp содержит устаревший код из cmu-snmp, датируемый 1997 годом, который при невозможности прочитать значения байтов из статистики интерфейса возвращает счётчик пакетов (который существует на loopback-интерфейсах), умноженный на произвольное значение 308. При этом предполагается, что средняя длина пакета составляет 308 октетов, что является очень грубой оценкой, поскольку предел MTU для loopback-интерфейсов в системах Solaris составляет 8892 байта. Не следует считать эти значения корректными или даже приблизительно точными. Это лишь приблизительные оценки. Агент Zabbix не делает никаких предположений, но 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 на момент запуска процесса.

[7] Элементы данных `vfs.file.contents[]`, `vfs.file.regexp[]`, `vfs.file.regmatch[]` можно использовать для получения содержимого файлов. Если вы хотите ограничить доступ к определённым файлам с конфиденциальной информацией, запустите агент Zabbix от имени пользователя, у которого нет прав доступа на просмотр этих файлов.

Использование с утилитами командной строки

Обратите внимание, что при тестировании или использовании ключей элементов данных с `zabbix_agentd` или `zabbix_get` из командной строки необходимо также учитывать синтаксис оболочки.

Например, если определённый параметр ключа должен быть заключён в двойные кавычки, необходимо явно экранировать двойные кавычки, иначе оболочка обработает их как специальные символы и они 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, вы можете 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 "Code Page Identifiers". Обратите внимание, что Microsoft иногда помечает некоторые кодовые страницы как "available only to managed applications" — то есть доступные только через среду выполнения .NET и недоступные через собственные API Win32. Агент Zabbix реализует собственную логику преобразования кодировок, поэтому эти кодовые страницы поддерживаются агентом Zabbix, даже если 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 UTF-8, данные обрабатываются "как есть";
- Анализ BOM — применимо для элементов данных 'vfs.file.contents', 'vfs.file.regexp', 'vfs.file.regmatch'. Предпринимается попытка определить правильную кодировку с использованием метки порядка байтов (BOM) в начале файла. Если BOM отсутствует, вместо этого применяется стандартное определение (см. выше).

Устранение неполадок элементов данных агента

В случае пассивных проверок, чтобы предотвратить ситуацию, когда элемент данных не получает никакого 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 глобальный тайм-аут) 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.

1. Zabbix агент 2

Zabbix агент 2 поддерживает все ключи элементов данных, поддерживаемые для Zabbix агент на Unix и Windows. На этой странице приведены сведения о дополнительных ключах элементов данных, которые можно использовать только с Zabbix агент 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 по пулам.	Ceph
ceph.osd.stats	Агрегированная статистика и статистика по каждому OSD.	
ceph.osd.discovery	Список обнаруженных OSD.	
ceph.osd.dump	Пороговые значения 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.	

Ключ элемента данных	Описание	Плагин
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+
memcached.ping	Проверяет, активно ли соединение.	Memcached
memcached.stats	Получает вывод команды STATS.	
mongodb.collection.stats	Возвращает различные статистические данные хранилища для указанной коллекции.	MongoDB
mongodb.cfg.discovery	Возвращает список обнаруженных серверов конфигурации.	MongoDB
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-чанков.	
mongodb.oplog.stats	Возвращает состояние набора реплик, используя данные, полученные 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 указанную тему или темы (с подстановочными символами) предоставленного брокера и ожидает публикации.	MQTT
mssql.availability.group.get	Возвращает группы доступности.	MSSQL
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, участвующие в группе доступности Always On и реплике (основной или вторичной) и расположенные на сервере, к которому было установлено соединение.	
mssql.mirroring.get	Возвращает информацию о зеркалировании.	
mssql.nonlocal.db.get	Возвращает базы данных, участвующие в группе доступности Always On и реплике (основной или вторичной), расположенные на других серверах (база данных не является локальной для экземпляра SQL Server, к которому было установлено соединение).	
mssql.perfcounter.get	Возвращает счётчики производительности.	
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 устройств GPU.	NVIDIA GPU
nvml.device.decoder.utilization	Использование декодера устройства GPU.	
nvml.device.ecc.mode	Режим ECC устройства GPU.	

Ключ элемента данных	Описание	Плагин
nvml.device.encoder.stats.get	Статистика кодировщика устройства GPU.	
nvml.device.encoder.utilization	Использование кодировщика устройства GPU.	
nvml.device.energy.consumption	Энергопотребление устройства GPU.	
nvml.device.errors.memory	Информация об ошибках ECC памяти устройства GPU.	
nvml.device.errors.register	Информация об ошибках ECC регистров устройства GPU.	
nvml.device.fan.speed.avg	Средняя скорость вентилятора устройства GPU.	
nvml.device.get	Обнаруженные GPU с UUID и именем.	
nvml.device.graphics.frequency	Частота графического процессора устройства GPU.	
nvml.device.memory.bar1.get	Информация о памяти BAR1 устройства GPU.	
nvml.device.memory.fb.get	Информация о памяти framebuffer устройства GPU.	
nvml.device.memory.frequency	Частота памяти устройства GPU.	
nvml.device.pci.utilization	Использование PCI устройства GPU.	
nvml.device.performance.state	Состояние производительности устройства GPU.	
nvml.device.power.limit	Предел энергопотребления устройства GPU.	
nvml.device.power.usage	Энергопотребление устройства GPU.	
nvml.device.serial	Серийный номер устройства GPU.	
nvml.device.sm.frequency	Частота потокового мультипроцессора устройства GPU.	
nvml.device.temperature	Температура устройства GPU.	
nvml.device.utilization	Статистика использования устройства GPU.	
nvml.device.video.frequency	Частота видеопроцессора устройства GPU.	
nvml.system.driver.version	Версия драйвера NVIDIA.	
nvml.version	Версия библиотеки NVML.	
oracle.diskgroups.stats	Возвращает статистику групп дисков Automatic Storage Management (ASM).	Oracle
oracle.diskgroups.discovery	Возвращает список групп дисков ASM.	
oracle.archive.info	Статистика архивных журналов.	
oracle.cdb.info	Информация о Container Databases (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 Pluggable Databases (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 файле журнала из управляющего файла.	
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.	
oracle.ts.discovery	Возвращает список табличных пространств.	
oracle.user.info	Возвращает информацию о пользователях Oracle.	
oracle.version	Возвращает версию сервера базы данных.	
pgsql.autovacuum.count	Количество рабочих процессов autovacuum.	PostgreSQL
pgsql.archive	Информация об архивированных файлах.	
pgsql.bgwriter	Общее количество контрольных точек для кластера базы 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 на базу 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 выданных блокировках по каждой базе данных.	
pgsql.oldest.xid	Возраст самого старого XID.	
pgsql.ping	Проверяет, активно ли соединение.	

Ключ элемента данных	Описание	Плагин
pgsql.queries	Метрики запросов по времени выполнения.	
pgsql.replication.count	Количество резервных серверов.	
pgsql.replication.process	Задержка flush, write и replay для каждого процесса отправителя.	
pgsql.replication.process.discovery	Обнаружение имён 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 непрерывной 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 в 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 модуля systemd.	Systemd
systemd.unit.info	Информация о модуле systemd.	
systemd.unit.discovery	Список модулей systemd и сведения о них.	
web.certificate.get	Проверяет сертификаты и возвращает сведения о сертификате.	Web certificates

См. также:

- Встроенные плагины
- Загружаемые плагины

Подробности ключа элемента данных

Параметры без угловых скобок являются обязательными. Параметры, отмеченные угловыми скобками **< >**, являются необязательными.

ceph.df.details[connString,<user>,<apikey>]

 Использование данных кластера и их распределение между пулами.
 Возвращаемое значение: *объект JSON*.

Параметры:

- **connString** — URI или имя сессии;
- **user, apikey** — имя пользователя и ключ API для подключения к Ceph.

ceph.osd.stats[connString,<user>,<apikey>]

 Агрегированная статистика и статистика по каждому OSD.
 Возвращаемое значение: *объект JSON*.

Параметры:

- **connString** - URI или имя сессии;
- **user, apikey** - имя пользователя и API-ключ для подключения к Ceph.

ceph.osd.discovery[connString,<user>,<apikey>]

 Список обнаруженных OSD. Используется для обнаружения низкого уровня.
 Возвращаемое значение: *объект JSON*.

Параметры:

- **connString** - URI или имя сессии;
- **user, apikey** - имя пользователя и ключ API для подключения к Ceph.

ceph.osd.dump[connString,<user>,<apikey>]

 Пороговые значения использования и статусы OSD.
 Возвращаемое значение: *объект JSON*.

Параметры:

- **connString** — URI или имя сессии;
- **user, apikey** — имя пользователя и ключ API для подключения к Ceph.

ceph.ping[connString,<user>,<apikey>]

 Проверяет, можно ли установить соединение с Ceph.
 Возвращаемое значение: *0* — соединение разорвано (при наличии любой ошибки, включая проблемы AUTH и конфигурации); *1* — соединение успешно установлено.

Параметры:

- **connString** — URI или имя сессии;
- **user, apikey** — имя пользователя и API key для подключения к Ceph.

ceph.pool.discovery[connString,<user>,<apikey>]

 Список обнаруженных пулов. Используется для low-level discovery.
 Возвращаемое значение: *JSON object*.

Параметры:

- **connString** — URI или имя сессии;
- **user, apikey** — имя пользователя и API key для подключения к Ceph.

ceph.status[connString,<user>,<apikey>]

 Общий статус кластера.
 Возвращаемое значение: *объект JSON*.

Параметры:

- **connString** — URI или имя сессии;
- **user, apikey** — имя пользователя и ключ API для подключения к Ceph.

docker.container_info[<ID>,<info>]

 Низкоуровневая информация о контейнере.
 Возвращаемое значение: вывод вызова API ContainerInspect, сериализованный в JSON.

Параметры:

- **ID** — ID или имя контейнера;
- **info** — объём возвращаемой информации. Поддерживаемые значения: *short* (по умолчанию) или *full*.

Пользователь Agent 2 (’zabbix’) должен быть добавлен в группу ’docker’ для получения достаточных привилегий. В противном случае проверка завершится ошибкой.

docker.container_stats[<ID>]

 Статистика использования ресурсов контейнера.
 Возвращаемое значение: вывод вызова API ContainerStats и процент использования CPU, сериализованные в JSON.

Параметр:

- **ID** — ID или имя контейнера.

Пользователь Agent 2 (’zabbix’) должен быть добавлен в группу ’docker’ для получения достаточных привилегий. В противном случае проверка завершится ошибкой.

docker.containers

 Список контейнеров.
 Возвращаемое значение: вывод вызова API ContainerList, сериализованный в JSON.

Пользователь Agent 2 (’zabbix’) должен быть добавлен в group ’docker’ для получения достаточных привилегий. В противном случае проверка завершится ошибкой.

docker.containers.discovery[<options>]

 Возвращает список контейнеров. Используется для low-level discovery.
 Возвращаемое значение: *объект JSON*.

Параметр:

- **options** — указывает, должны ли обнаруживаться все контейнеры или только запущенные. Поддерживаемые значения: *true* — вернуть все контейнеры; *false* — вернуть только запущенные контейнеры (по умолчанию).

Пользователь Agent 2 (’zabbix’) должен быть добавлен в group ’docker’ для получения достаточных привилегий. В противном случае проверка завершится ошибкой.

docker.data.usage

 Информация о текущем использовании данных.
 Возвращаемое значение: вывод вызова API SystemDataUsage, сериализованный в JSON.

Пользователь Agent 2 (’zabbix’) должен быть добавлен в group ’docker’ для получения достаточных привилегий. В противном случае проверка завершится ошибкой.

docker.images

 Возвращает список образов.
 Возвращаемое значение: вывод вызова API ImageList, сериализованный в JSON.

Пользователь Agent 2 (’zabbix’) должен быть добавлен в группу ’docker’ для получения достаточных привилегий. В противном случае проверка завершится ошибкой.

docker.images.discovery

 Возвращает список образов. Используется для low-level discovery.
 Возвращаемое значение: *объект JSON*.

Пользователь Agent 2 (’zabbix’) должен быть добавлен в группу ’docker’ для получения достаточных привилегий. В противном случае проверка завершится ошибкой.

docker.info

 Системная информация.
 Возвращаемое значение: вывод вызова API SystemInfo, сериализованный в JSON.

Пользователь Agent 2 (’zabbix’) должен быть добавлен в группу ’docker’ для получения достаточных привилегий. В противном случае проверка завершится ошибкой.

docker.ping

 Проверяет, запущен ли демон Docker.
 Возвращаемое значение: *1* — соединение активно; *0* — соединение разорвано.

Пользователь Agent 2 (’zabbix’) должен быть добавлен в group ’docker’ для получения достаточных привилегий. В противном случае проверка завершится неудачей.

ember.get[<uri>,<path>]

 Возвращает результат от требуемого устройства.
 Возвращаемое значение: *объект JSON*.

Параметры:

- **uri** — URI устройства Ember+. По умолчанию: 127.0.0.1:9998;
- **path** — путь OID 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.

Parameters:

- **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[connString,<user>,<password>]

 Возвращает список обнаруженных коллекций. Используется для обнаружения низкого уровня.
 Возвращаемое значение: *объект JSON*.

Параметры:

- **connString** — URI или имя сессии;
- **user, password** — учетные данные для входа в 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[connString,<user>,<password>]

 Возвращает информацию об открытых исходящих соединениях от текущего экземпляра базы данных к другим участникам шардированного кластера или набора реплик.
 Возвращаемое значение: *объект JSON*.

Параметры:

- **connString** — URI или имя сессии;
- **user, password** — учетные данные для входа в 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[connString,<user>,<password>]

 Возвращает список обнаруженных баз данных. Используется для low-level discovery.
 Возвращаемое значение: *объект JSON*.

Параметры:

- **connString** — URI или имя сессии;
- **user, password** — учетные данные для входа в 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 видимости 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 определённый топик или топики (с подстановочными символами) указанного брокера и ожидает публикации.
 Возвращаемое значение: зависит от содержимого топика. Если используются подстановочные символы, возвращает содержимое топика в формате JSON.

Параметры:

- **broker url** — URL MQTT-брокера 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 не указаны, будут использованы протокол (`tcp`) или порт (1883) по умолчанию;
- **topic** — MQTT-топик (обязательный параметр). Поддерживаются подстановочные символы (+,#);
- **user, password** — учётные данные для аутентификации (если требуется).

Комментарии:

- Элемент данных должен быть настроен как активная проверка (тип элемента данных «Zabbix агент (активный)»);
- Сертификаты TLS-шифрования можно использовать, сохранив их в расположение по умолчанию (например, каталог `/etc/ssl/certs/` в Ubuntu). Для TLS используйте схему `tls://`.

mssql.availability.group.get[URI,<user>,<password>]

 Возвращает группы доступности.
 Возвращаемое значение: *объект JSON*.

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.

Для получения дополнительной информации смотрите readme плагина MSSQL.

mssql.custom.query[URI,<user>,<password>,queryName,<args...>]

 Возвращает результат пользовательского запроса.
 Возвращаемое значение: *объект JSON*.

Параметры:

- **URI** — URI сервера MSSQL (поддерживается только 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 в запрос.

Для получения дополнительной информации см. readme плагина MSSQL.

mssql.db.get

 Возвращает все доступные базы данных MSSQL.
 Возвращаемое значение: *Объект JSON*.

Для получения дополнительной информации см. readme плагина MSSQL [en].

mssql.job.status.get

 Возвращает состояние джобов.
 Возвращаемое значение: *Объект JSON*.

Для получения дополнительной информации см. readme плагина MSSQL [en].

mssql.last.backup.get

 Возвращает для всех баз данных время последнего резервного копирования.
 Возвращаемое значение: *Объект JSON*.

Для получения дополнительной информации см. readme плагина MSSQL [en].

mssql.local.db.get

 Возвращает базы данных, которые участвуют в группе доступности Always On, и реплики (основная или вторичная) расположены на сервере, к которому установлено подключение.
 Возвращаемое значение: *Объект JSON*.

Для получения дополнительной информации см. readme плагина MSSQL [en].

mssql.mirroring.get

 Возвращает информацию о зеркалировании.
 Возвращаемое значение: *Объект JSON*.

Для получения дополнительной информации см. readme плагина MSSQL [en].

mssql.nonlocal.db.get

 Возвращает базы данных, которые участвуют в группе доступности Always On и реплики (основная или вторичная) расположены на других серверах (база данных не является локальной на экземпляре сервера SQL, к которому установлено подключение).
 Возвращаемое значение: *Объект JSON*.

Для получения дополнительной информации см. readme плагина MSSQL [en].

mssql.perfcounter.get

 Возвращает счётчики производительности.
 Возвращаемое значение: *Объект JSON*.

Для получения дополнительной информации см. readme плагина MSSQL [en].

mssql.ping

 Проверка доступности базы данных. Проверяет, правильно ли настроено подключение.
 Возвращаемое значение: *1* — доступна, *0* — недоступна.

Для получения дополнительной информации смотрите readme плагина MSSQL.

mssql.quorum.get

 Возвращает информацию о кворуме.
 Возвращаемое значение: *Объект JSON*.

Для получения дополнительной информации см. readme плагина MSSQL [en].

mssql.quorum.member.get

 Возвращает информацию о членах кворума.
 Возвращаемое значение: *Объект JSON*.

Для получения дополнительной информации см. readme плагина MSSQL [en].

mssql.replica.get

 Возвращает реплики.
 Возвращаемое значение: *Объект JSON*.

Для получения дополнительной информации см. readme плагина MSSQL [en].

mssql.version

 Возвращает версию MSSQL.
 Возвращаемое значение: *Строка*.

Для получения дополнительной информации см. readme плагина MSSQL [en].

mysql.custom.query[connString,<user>,<password>,queryName,<args...>]

 Возвращает результат пользовательского запроса.
 Возвращаемое значение: *объект JSON*.

Параметры:

- **connString** — URI или имя сессии;
- **user, password** — учетные данные для входа в MySQL;
- **queryName** — имя пользовательского запроса; должно совпадать с именем SQL-файла без расширения;
- **args** — один или несколько аргументов, разделенных запятыми, для передачи в запрос.

Для получения дополнительной информации смотрите readme плагина MySQL, раздел *Пользовательские запросы*.

mysql.db.discovery[connString,<user>,<password>]

 Возвращает список баз данных MySQL. Используется для обнаружения низкого уровня.
 Возвращаемое значение: результат SQL-запроса ”show databases” в формате LLD JSON.

Параметры:

- **connString** — URI или имя сессии;
- **user, password** — учетные 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[connString,<user>,<password>]

 Возвращает список репликаций MySQL. Используется для обнаружения низкого уровня.
 Возвращаемое значение: результат SQL-запроса `SHOW SLAVE STATUS` или `SHOW REPLICA STATUS` в формате LLD JSON.

Параметры:

- **connString** — URI или имя сессии;
- **user, password** — учетные данные для входа в MySQL.

mysql.replication.get_slave_status[connString,<user>,<password>,<master host>]

 Статус репликации.
 Возвращаемое значение: результат SQL-запроса `SHOW SLAVE STATUS` или `SHOW REPLICA STATUS` в формате JSON.

Параметры:

- **connString** — URI или имя сессии;
- **user, password** — учетные данные для входа в MySQL;
- **master host** — имя узла сети мастера репликации. Если ничего не найдено, возвращается ошибка. Если этот параметр не указан, возвращаются все узлы сети.

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 агента Zabbix `net.dns.record` 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 для запроса (по умолчанию *SOA*);
- **timeout** — тайм-аут 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, разделённых запятыми, для передачи в запрос.

Комментарии:

- Возможные значения для `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 или смешанном регистре не поддерживаются.
- Для обратных DNS-запросов (когда `type` установлен в *PTR*) можно указывать DNS-имя как в обратном, так и в обычном формате (см. примеры ниже). Обратите внимание, что при запросе PTR-записи DNS-имя фактически является IP-адресом.
- Возможные значения для `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 `flags` *dnssec* и *nsid* нельзя использовать вместе с *noedns0*, так как оба требуют *edns0*. Обратите внимание, что значения должны быть только в нижнем регистре; значения в верхнем или смешанном регистре не поддерживаются.
- Интернационализированные доменные имена 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[192.0.2.0,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: *объект JSON*.

Параметры:

- **connString** — может быть:
  - URI — если сеанс с указанным именем не найден и `Plugins.Oracle.ResolveTNS` установлено 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 обрабатываться как URI;
  - именем сеанса — если такое имя найдено в файле конфигурации плагина;
  - ключом TNS — если параметр ResolveTNS установлен в true и ни одно из указанных выше условий не выполняется;
  - значением 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 нечувствителен к регистру и не должен содержать завершающий пробел).
- **password** — пароль Oracle;
- **service** — имя сервиса Oracle;
- **diskgroup** — имя группы дисков ASM для запроса.

oracle.diskgroups.discovery[connString,<user>,<password>,<service>]

 Возвращает список групп дисков ASM. Используется для обнаружения низкого уровня.
 Возвращаемое значение: *объект JSON*.

Параметры:

- **connString** - может быть:
  - URI — если сеанс с указанным именем не найден и `Plugins.Oracle.ResolveTNS` установлено 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 обрабатываться как URI;
  - именем сеанса — если такое имя найдено в файле конфигурации плагина;
  - ключом TNS — если параметр ResolveTNS установлен в true и ни одно из указанных выше условий не выполняется;
  - значением 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 нечувствителен к регистру и не должен содержать завершающий пробел).
- **password** - пароль Oracle;
- **service** - имя сервиса Oracle.

oracle.archive.info[connString,<user>,<password>,<service>,<destination>]

 Статистика журналов архивации.
 Возвращаемое значение: *объект JSON*.

Параметры:

- **connString** — может быть:
  - URI — если сеанс с указанным именем не найден и `Plugins.Oracle.ResolveTNS` установлено в false. Если он содержит схему (например, "tcp://"), порт (например, 1521) или и то, и другое, параметр ResolveTNS не будет учитываться, и значение в любом случае будет обрабатываться как URI;
  - именем сеанса — если такое имя найдено в файле конфигурации плагина;
  - ключом TNS — если параметр ResolveTNS установлен в true и ни одно из указанных выше условий не выполняется;
  - значением 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 нечувствителен к регистру и не должен содержать завершающий пробел).
- **password** — пароль Oracle;
- **service** — имя сервиса Oracle;
- **destination** — имя назначения для запроса.

oracle.cdb.info[connString,<user>,<password>,<service>,<database>]

 Информация о Container Databases (CDB).
 Возвращаемое значение: *объект JSON*.

Параметры:

- **connString** — может быть:
  - URI — если сеанс с указанным именем не найден и `Plugins.Oracle.ResolveTNS` установлено 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 рассматриваться как URI;
  - именем сеанса — если такое имя найдено в файле конфигурации плагина;
  - ключом TNS — если параметр ResolveTNS установлен в true и ни одно из указанных выше условий не выполняется;
  - значением 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 нечувствителен к регистру и не должен содержать пробел в конце).
- **password** — пароль Oracle;
- **service** — имя сервиса Oracle;
- **database** — имя базы данных для запроса.

oracle.custom.query[connString,<user>,<password>,<service>,queryName,<args...>]

 Результат пользовательского запроса.
 Возвращаемое значение: *объект JSON*.

Параметры:

- **connString** — может быть:
  - URI — если сеанс с указанным именем не найден и `Plugins.Oracle.ResolveTNS` установлено 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 обрабатываться как URI;
  - именем сеанса — если такое имя найдено в файле конфигурации плагина;
  - ключом TNS — если параметр ResolveTNS установлен в true и ни одно из указанных выше условий не выполняется;
  - значением 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 нечувствителен к регистру и не должен содержать завершающий пробел).
- **password** — пароль Oracle;
- **service** — имя сервиса Oracle;
- **queryName** — имя пользовательского запроса; должно совпадать с именем SQL-файла без расширения;
- **args** — один или несколько аргументов, разделённых запятыми, для передачи в запрос.

Комментарии: - Возвращаемые данные автоматически преобразуются в JSON. - Не рекомендуется возвращать JSON непосредственно из запросов, так как он будет повреждён, когда плагин попытается снова преобразовать его в JSON.

oracle.datafiles.stats[connString,<user>,<password>,<service>]

 Возвращает статистику файлов данных.
 Возвращаемое значение: *объект JSON*.

Параметры:

- **connString** — может быть:
  - URI — если сеанс с указанным именем не найден и `Plugins.Oracle.ResolveTNS` установлено 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 обрабатываться как URI;
  - именем сеанса — если такое имя найдено в файле конфигурации плагина;
  - ключом TNS — если параметр ResolveTNS установлен в true и ни одно из указанных выше условий не выполняется;
  - значением 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 нечувствителен к регистру и не должен содержать завершающий пробел).
- **password** — пароль Oracle;
- **service** — имя сервиса Oracle.

oracle.db.discovery[connString,<user>,<password>,<service>]

 Возвращает список баз данных. Используется для обнаружения низкого уровня.
 Возвращаемое значение: *объект JSON*.

Параметры:

- **connString** — может быть:
  - URI — если сеанс с указанным именем не найден и `Plugins.Oracle.ResolveTNS` установлено 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 обрабатываться как URI;
  - именем сеанса — если такое имя найдено в файле конфигурации плагина;
  - ключом TNS — если параметр ResolveTNS установлен в true и ни одно из указанных выше условий не выполняется;
  - значением 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 нечувствителен к регистру и не должен содержать завершающий пробел).
- **password** — пароль Oracle;
- **service** — имя сервиса Oracle.

oracle.fra.stats[connString,<user>,<password>,<service>]

 Возвращает статистику Fast Recovery Area (FRA).
 Возвращаемое значение: *объект JSON*.

Параметры:

- **connString** — может быть:
  - URI — если сеанс с указанным именем не найден и `Plugins.Oracle.ResolveTNS` установлено 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 рассматриваться 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; 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 нечувствителен к регистру и не должен содержать завершающий пробел).
- **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 `Plugins.Oracle.ResolveTNS` установлено в false. Если он содержит схему (например, "tcp://"), порт (например, 1521) или и то, и другое, параметр ResolveTNS не будет учитываться, и значение в любом случае будет обрабатываться как URI;
  - именем сеанса — если такое имя найдено в файле конфигурации плагина;
  - ключом TNS — если параметр ResolveTNS установлен в true и ни одно из указанных выше условий не выполняется;
  - значением 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 нечувствителен к регистру и не должен содержать завершающий пробел).
- **password** — пароль Oracle;
- **service** — имя сервиса Oracle.

oracle.pdb.info[connString,<user>,<password>,<service>,<database>]

 Информация о подключаемых базах данных (PDB).
 Возвращаемое значение: *объект JSON*.

Параметры:

- **connString** — может быть:
  - URI — если сеанс с указанным именем не найден и `Plugins.Oracle.ResolveTNS` установлено 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 рассматриваться как URI;
  - именем сеанса — если такое имя найдено в файле конфигурации плагина;
  - ключом TNS — если параметр ResolveTNS установлен в true и ни одно из указанных выше условий не выполняется;
  - значением 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 нечувствителен к регистру и не должен содержать пробел в конце).
- **password** — пароль Oracle;
- **service** — имя сервиса Oracle;
- **database** — имя базы данных для выполнения запроса.

oracle.pdb.discovery[connString,<user>,<password>,<service>]

 Возвращает список PDB. Используется для обнаружения низкого уровня.
 Возвращаемое значение: *объект JSON*.

Параметры:

- **connString** — может быть:
  - URI — если сеанс с указанным именем не найден и `Plugins.Oracle.ResolveTNS` установлено 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 рассматриваться как URI;
  - именем сеанса — если такое имя найдено в файле конфигурации плагина;
  - ключом TNS — если параметр ResolveTNS установлен в true и ни одно из указанных выше условий не выполняется;
  - значением 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 нечувствителен к регистру и не должен содержать пробел в конце).
- **password** — пароль Oracle;
- **service** — имя сервиса Oracle.

oracle.pga.stats[connString,<user>,<password>,<service>]

 Возвращает статистику Program Global Area (PGA).
 Возвращаемое значение: *объект JSON*.

Параметры:

- **connString** — может быть:
  - URI — если сеанс с указанным именем не найден и `Plugins.Oracle.ResolveTNS` установлено 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 обрабатываться как URI;
  - именем сеанса — если такое имя найдено в файле конфигурации плагина;
  - ключом TNS — если параметр ResolveTNS установлен в true и ни одно из указанных выше условий не выполняется;
  - значением 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 нечувствителен к регистру и не должен содержать пробел в конце).
- **password** — пароль Oracle;
- **service** — имя сервиса Oracle.

oracle.ping[connString,<user>,<password>,<service>]

 Проверяет, может ли быть установлено соединение с Oracle.
 Возвращаемое значение: *1* — соединение успешно установлено; *0* — соединение разорвано (при наличии любой ошибки, включая проблемы AUTH и конфигурации).

Параметры:

- **connString** — может быть:
  - URI — если сеанс с указанным именем не найден и `Plugins.Oracle.ResolveTNS` установлено в false. Если он содержит схему (например, ”tcp://”), порт (например, 1521) или и то, и другое, параметр ResolveTNS не будет учитываться, и значение в любом случае будет обрабатываться как URI;

    - именем сеанса — если такое имя найдено в файле конфигурации плагина;
    - ключом TNS — если параметр ResolveTNS установлен в true и ни одно из указанных выше условий не выполняется;
    - значением 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 нечувствителен к регистру и не должен содержать завершающий пробел).
- **password** — пароль Oracle;
- **service** — имя сервиса 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: *объект JSON*.

Параметры:

- **connString** — может быть:
    - URI — если сеанс с указанным именем не найден и `Plugins.Oracle.ResolveTNS` установлено 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 рассматриваться как URI;
    - именем сеанса — если такое имя найдено в файле конфигурации плагина;
    - ключом TNS — если параметр ResolveTNS установлен в true и ни одно из указанных выше условий не выполняется;
    - значением 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 нечувствителен к регистру и не должен содержать завершающий пробел).
- **password** — пароль Oracle;
- **service** — имя сервиса Oracle.

oracle.redolog.info[connString,<user>,<password>,<service>]

 Информация о файле журнала из управляющего файла.
 Возвращаемое значение: *объект JSON*.

Параметры:

- **connString** — может быть:
    - URI — если сеанс с указанным именем не найден и `Plugins.Oracle.ResolveTNS` установлено 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 значение будет рассматриваться как URI;
    - именем сеанса — если такое имя найдено в файле конфигурации плагина;
    - ключом TNS — если параметр ResolveTNS установлен в true и ни одно из указанных выше условий не выполняется;
    - значением 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 нечувствителен к регистру и не должен содержать пробел в конце).
- **password** — пароль Oracle;
- **service** — имя сервиса Oracle.

oracle.sga.stats[connString,<user>,<password>,<service>]

 Возвращает статистику System Global Area (SGA).
 Возвращаемое значение: *объект JSON*.

Параметры:

- **connString** — может быть:
    - URI — если сеанс с указанным именем не найден и `Plugins.Oracle.ResolveTNS` установлено 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 рассматриваться как URI;

    - именем сеанса — если такое имя найдено в файле конфигурации плагина;
    - ключом TNS — если параметр ResolveTNS установлен в true и ни одно из указанных выше условий не выполняется;
    - значением 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 нечувствителен к регистру и не должен содержать завершающий пробел).
- **password** — пароль Oracle;
- **service** — имя сервиса 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 `Plugins.Oracle.ResolveTNS` установлено 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 обрабатываться как URI;
    - именем сеанса — если такое имя найдено в файле конфигурации плагина;
    - ключом TNS — если параметр ResolveTNS установлен в true и ни одно из указанных выше условий не выполняется;
    - значением 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 нечувствителен к регистру и не должен содержать завершающий пробел).
- **password** — пароль Oracle;
- **service** — имя сервиса Oracle;
- **lockMaxTime** — максимальная длительность блокировки сеанса в секундах, при превышении 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 быть:
    - URI — если сеанс с указанным именем не найден и `Plugins.Oracle.ResolveTNS` установлено 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 обрабатываться как URI;
    - именем сеанса — если такое имя найдено в файле конфигурации плагина;
    - ключом TNS — если параметр ResolveTNS установлен в true и ни одно из вышеперечисленных условий не выполняется;
    - значением 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 нечувствителен к регистру и не должен содержать завершающий пробел).
- **password** — пароль Oracle;
- **service** — имя сервиса Oracle;
- **duration** — интервал сбора (в секундах) значений системных метрик. Возможные значения: *60* — длительный интервал (по умолчанию), *15* — короткий интервал.

oracle.sys.params[connString,<user>,<password>,<service>]

 Возвращает набор значений системных параметров.
 Возвращаемое значение: *объект JSON*.

Параметры:

- **connString** — может быть:
    - URI — если сеанс с указанным именем не найден и `Plugins.Oracle.ResolveTNS` установлено 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 обрабатываться как URI;
    - именем сеанса — если такое имя найдено в файле конфигурации плагина;

    - ключом TNS — если параметр ResolveTNS установлен в true и ни одно из указанных выше условий не выполняется;
    - значением 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 нечувствителен к регистру и не должен содержать пробел в конце).
- **password** — пароль Oracle;
- **service** — имя сервиса Oracle.

oracle.ts.stats[connString,<user>,<password>,<service>,<tablespace>,<type>,<conname>]

 Возвращает статистику табличных пространств.
 Возвращаемое значение: *объект JSON*.

Параметры:

- **connString** — может быть:
    - URI — если сеанс с указанным именем не найден и `Plugins.Oracle.ResolveTNS` установлено 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 обрабатываться как URI;
    - именем сеанса — если такое имя найдено в файле конфигурации плагина;
    - ключом TNS — если параметр ResolveTNS установлен в true и ни одно из указанных выше условий не выполняется;
    - значением 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 нечувствителен к регистру и не должен содержать завершающий пробел).
- **password** — пароль Oracle;
- **service** — имя сервиса Oracle;
- **tablespace** — имя табличного пространства для запроса. По умолчанию (если не указано и задан `type`):
    - ”TEMP” (если `type` задан как ”TEMPORARY”);
    - ”USERS” (если `type` задан как ”PERMANENT”).
- **type** — тип табличного пространства для запроса. По умолчанию (если задан `tablespace`): ”PERMANENT”.
- **conname** — имя контейнера, для которого требуется информация.

Если `tablespace`, `type` или `conname` не указаны, элемент данных вернет статистику табличных пространств для всех соответствующих контейнеров (включая PDB и CDB).

oracle.ts.discovery[connString,<user>,<password>,<service>]

 Возвращает список табличных пространств. Используется для обнаружения низкого уровня.
 Возвращаемое значение: *объект JSON*.

Параметры:

- **connString** — может быть:
    - URI — если сеанс с указанным именем не найден и `Plugins.Oracle.ResolveTNS` установлено 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 обрабатываться как URI;
    - именем сеанса — если такое имя найдено в файле конфигурации плагина;
    - ключом TNS — если параметр ResolveTNS установлен в true и ни одно из указанных выше условий не выполняется;
    - значением 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 нечувствителен к регистру и 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: *объект JSON*.

Параметры:

- **connString** — может быть:

  - URI — если сеанс с указанным именем не найден и `Plugins.Oracle.ResolveTNS` установлено 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 значение будет обрабатываться 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; 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 нечувствителен к регистру и не должен содержать завершающий пробел).
- **password** — пароль Oracle;
- **service** — имя сервиса Oracle;
- **username** — имя пользователя, для которого требуется 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` установлено 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 рассматриваться как URI;
  - именем сеанса — если такое имя найдено в файле конфигурации плагина;
  - ключом TNS — если параметр ResolveTNS установлен в true и ни одно из указанных выше условий не выполняется;
  - значением 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 нечувствителен к регистру и не должен содержать пробел в конце).
- **password** — пароль Oracle;
- **service** — имя сервиса Oracle.

pgsql.autovacuum.count[uri,<username>,<password>,<database name>]

 Количество процессов autovacuum.
 Возвращаемое значение: *Целое число*.

Параметры:

- **uri** — URI или имя сессии;
- **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: *объект JSON*.

Параметры:

- **uri** — URI или имя сессии;
- **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 с разбивкой по типу контрольной точки.
 Возвращаемое значение: *объект JSON*.

Параметры:

- **uri** — URI или имя сессии;
- **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 сессии;
- **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: *объект JSON*.

Параметры:

- **uri** — URI или имя сессии;
- **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,<имя_пользователя>,<пароль>,имя_запроса,<аргументы...>]

 Возвращает результат пользовательского запроса.
 Возвращаемое значение: *Объект JSON*.

Параметры:

- **uri** — URI или имя сессии;
- **имя_пользователя, пароль** — учётные данные PostgreSQL;
- **имя_запроса** — имя пользовательского запроса, должно совпадать с именем sql файла без расширения;
- **аргументы** — один или несколько аргументов (через запятую), которые будут переданы запросу.

pgsql.db.age[uri,<username>,<password>,<database name>]

 Возраст самого старого FrozenXID базы данных.
 Возвращаемое значение: *Целое число*.

Параметры:

- **uri** — URI или имя сессии;
- **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 таблиц с разрастанием в базе данных.
 Возвращаемое значение: *Integer*.

Параметры:

- **uri** — URI или имя сессии;
- **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 низкого уровня.
 Возвращаемое значение: *объект JSON*.

Параметры:

- **uri** — URI или имя сессии;
- **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 сессии;
- **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 низкого уровня.
 Возвращаемое значение: *объект JSON*.

Параметры:

- **uri** — URI или имя сессии;
- **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 по всем базам данных в кластере.
 Возвращаемое значение: *объект JSON*.

Параметры:

- **uri** — URI или имя сессии;
- **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 низкого уровня.
 Возвращаемое значение: *объект JSON*.

Параметры:

- **uri** — URI или имя сессии;
- **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.
 Возвращаемое значение: *Integer*.

Параметры:

- **uri** - URI или имя сессии;
- **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 (при наличии любой ошибки, включая проблемы AUTH и конфигурации).

Параметры:

- **uri** — URI или имя сессии;
- **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]

 Метрики запросов по времени выполнения.
 Возвращаемое значение: *объект JSON*.

Параметры:

- **uri** — URI или имя сессии;
- **username, password** — учетные данные PostgreSQL;
- **database name** — имя базы данных (если не указано, используется значение Plugins.PostgreSQL.Default.Database из postgresql.conf);
- **time period** — ограничение 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 непрерывной работы PostgreSQL в миллисекундах.
 Возвращаемое значение: *Float*.

Параметры:

- **uri** — URI или имя сессии;
- **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: *Строка*.

Параметры:

- **uri** — URI или имя сессии;
- **username, password** — учетные данные PostgreSQL;
- **database name** — имя базы данных (если опущено, используется значение Plugins.PostgreSQL.Default.Database из postgresql.conf).

pgsql.wal.stat[uri,<username>,<password>,<database name>]

 Статистика WAL.
 Возвращаемое значение: *объект JSON*.

Параметры:

- **uri** — URI или имя сессии;
- **username, password** — учетные данные PostgreSQL;
- **database name** — имя базы данных (если не указано, используется значение Plugins.PostgreSQL.Default.Database из postgresql.conf).

redis.config[connString,<password>,<pattern>]

 Получает параметры конфигурации экземпляра Redis, соответствующие шаблону.
 Возвращаемое значение: *JSON* — если использовался шаблон в стиле glob; одиночное значение — если шаблон не содержал подстановочных символов.

Параметры:

- **connString** — URI или имя сеанса;
- **password** — пароль Redis;
- **pattern** — 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 секретные данные). Использование шаблона по умолчанию (`*`) или широких шаблонов может раскрыть эти секретные данные любому, у кого есть разрешение на просмотр значений элементов данных в Zabbix.

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: *объект JSON*.

Комментарии:

- Возвращаются следующие макросы и их значения: {#NAME}, {#DISKTYPE}, {#ID}, {#ATTRNAME}, {#THRESH};
- Поддерживаются типы накопителей HDD, SSD и NVME. Накопители могут быть как отдельными, так и объединёнными в RAID. {#NAME} будет иметь дополнение в случае RAID, например: {"{#NAME}": "/dev/sda cciss,2"}.

smart.disk.discovery[<type>]

 Возвращает список устройств S.M.A.R.T.
 Возвращаемое значение: *объект JSON*.

Параметр:

- **type** — указывает значение для сканирования 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 принадлежит RAID, {#RAIDTYPE} будет пустым. {#NAME} будет содержать 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: *объект JSON*.

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 использоваться отдельно или быть объединены 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 модуля,<интерфейс>]

 Возвращает все свойства модуля systemd.
 Возвращаемое значение: *Объект JSON*.

Параметры:

- **имя модуля** — имя модуля (вы, возможно, захотите использовать {#UNIT.NAME} макрос в прототипе элементов данных для обнаружения имени);
- **интерфейс** — тип интерфейса модуля, 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 модуля возвращаются в виде текста и целого числа: `"ActiveState":{"state":1,"text":"active"}`.

systemd.unit.info[имя модуля,<свойство>,<интерфейс>]

 Информация о модуле systemd.
 Возвращаемое значение: *String*.

Параметры:

- **имя модуля** — имя модуля (для обнаружения 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).

Комментарии:

- Этот элемент данных поддерживается только на платформе Linux;
- Этот элемент данных позволяет получать определённое свойство из определённого типа интерфейса, как описано в dbus API.

Примеры:

```
systemd.unit.info["{#UNIT.NAME}"] #collect active state (active, reloading, inactive, failed, activating,
systemd.unit.info["{#UNIT.NAME}",LoadState] #collect load state info on discovered systemd units
systemd.unit.info[mysqld.service,Id] #retrieve the service technical name (mysqld.service)
systemd.unit.info[mysqld.service,Description] #retrieve the service description (MySQL Server)
systemd.unit.info[mysqld.service,ActiveEnterTimestamp] #retrieve the last time the service entered the act
systemd.unit.info[dbus.socket,NConnections,Socket] #collect the number of connections from this socket uni
```

systemd.unit.discovery[<type>]

 Список модулей systemd и их подробных сведений. Используется для обнаружения низкого уровня.
 Возвращаемое значение: *объект JSON*.

Параметр:

- **type** — 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 как IP-адресом, так и 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 имени узла;
- **port** — номер порта (по умолчанию для HTTPS используется 443);
- **address** — может быть как IP-адресом, так и DNS-именем. Если указан, он будет использоваться для подключения, а hostname (1-й параметр) будет использоваться для SNI и проверки узла. Если 1-й параметр является IP-адресом, а 3-й параметр — DNS-именем, 1-й параметр будет использоваться для подключения, а 3-й параметр — для SNI и проверки узла.

Комментарии:

- Этот элемент данных становится неподдерживаемым, если назначение, указанное в конфигурации узла сети, не существует, недоступно или если TLS-рукопожатие завершается с любой ошибкой, кроме недействительного сертификата;
- В настоящее время не поддерживаются расширение X.509 AIA (Authority Information Access), CRL и OCSP (включая OCSP stapling), а также Certificate Transparency;
- Поля ответа JSON:
  - *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 срока действия сертификата.
    - *not_after*: дата окончания срока действия сертификата.
    - *subject*: субъект сертификата.
    - *public_key_algorithm*: алгоритм, используемый для открытого ключа (например, `RSA`).
    - *alternative_names*: альтернативные имена субъекта (если присутствуют), в противном случае `null`.
  - result: содержит результат проверки.
    - *value*: статус проверки (см. возможные значения ниже).
    - *message*: подробное сообщение о проверке (например, `"certificate verified successfully"`).
  - *sha1_fingerprint*: отпечаток SHA-1 сертификата.
  - *sha256_fingerprint*: отпечаток SHA-256 сертификата.
- Поле `$.result.value` указывает результат проверки сертификата. Возможные значения:
  - *valid* – сертификат действителен и доверенный.
  - *valid-but-self-signed* – сертификат действителен, но является самоподписанным, то есть его субъект совпадает с его издателем.
  - *invalid* – сертификат недействителен из-за такой проблемы, как истечение срока действия, неверное имя узла или неизвестный центр подписи.

Пример:

```
web.certificate.get[example.com,443]
```

Ответ JSON:

```
{
  "x509": {
    "version": 3,
    "serial_number": "0ad893bafa68b0b7fb7a404f06ecaf9a",
    "signature_algorithm": "ECDSA-SHA384",
    "issuer": "CN=DigiCert Global G3 TLS ECC SHA384 2020 CA1,O=DigiCert Inc,C=US",
    "not_before": {
      "value": "Jan 15 00:00:00 2025 GMT",
      "timestamp": 1736899200
    },
    "not_after": {
      "value": "Jan 15 23:59:59 2026 GMT",
      "timestamp": 1768521599
    },
    "subject": "CN=*.example.com,O=Internet Corporation for Assigned Names and Numbers,L=Los Angeles,ST=Ca
    "public_key_algorithm": "ECDSA",
    "alternative_names": [
      "*.example.com",
      "example.com"
    ]
  },
  "result": {
    "value": "valid",
    "message": "certificate verified successfully"
  },
  "sha1_fingerprint": "310db7af4b2bc9040c8344701aca08d0c69381e3",
  "sha256_fingerprint": "455943cf819425761d1f950263ebf54755d8d684c25535943976f488bc79d23b"
}
```


2. Zabbix агент для Windows

Обзор

Элементы данных Windows агента Zabbix представлены в двух списках:

- Общие элементы данных — ключи элементов данных, общие с UNIX агентом Zabbix;
- Элементы данных только для Windows — ключи элементов данных, которые поддерживаются **только** в Windows.

Обратите внимание, что все ключи элементов данных, поддерживаемые агентом Zabbix в Windows, также поддерживаются агентом Zabbix нового поколения 2. См. дополнительные ключи элементов данных, которые можно использовать только с агентом 2.

См. 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 UNIX-агентом Zabbix:

- Ключ элемента данных является ссылкой на полное описание элемента данных UNIX-агента Zabbix
- 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 отслеживаемом файле журнала. Этот элемент данных не поддерживается для журнала событий Windows. Параметр `persistent_dir` не поддерживается в Windows.	
logrt	Мониторинг файла журнала с ротацией. Этот элемент данных не поддерживается для журнала событий Windows. Параметр `persistent_dir` не поддерживается в Windows.	

Item key	Description	Item group
logrt.count	Количество совпавших строк в отслеживаемом файле журнала с ротацией. Этот элемент данных не поддерживается для журнала событий Windows. Параметр `persistent_dir` не поддерживается в Windows.	
modbus.get	Читает данные Modbus.	Modbus
net.dns	Проверяет, доступна ли служба DNS. Параметры `ip`, `timeout` и `count` игнорируются в Windows, если не используется Zabbix агент 2.	Сеть
net.dns.perf	Проверяет производительность службы DNS. Параметры `ip`, `timeout` и `count` игнорируются в Windows, если не используется Zabbix агент 2.	
net.dns.record	Выполняет DNS-запрос. Параметры `ip`, `timeout` и `count` игнорируются в Windows, если не используется Zabbix агент 2.	
net.if.discovery	Список сетевых интерфейсов. Некоторые версии Windows (например, Server 2008) могут требовать установки последних обновлений для поддержки символов не-ASCII в именах интерфейсов.	
net.if.in	Статистика входящего трафика на сетевом интерфейсе. В Windows элемент данных получает значения из 64-битных счётчиков, если они доступны. 64-битные счётчики статистики интерфейсов были представлены в Windows Vista и Windows Server 2008. Если 64-битные счётчики недоступны, агент использует 32-битные счётчики. В Windows поддерживаются многобайтные имена интерфейсов. Получить описания сетевых интерфейсов в Windows можно 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 счётчиков, если они доступны. 64-битные счётчики статистики интерфейсов были представлены в Windows Vista и Windows Server 2008. Если 64-битные счётчики недоступны, агент использует 32-битные счётчики. В Windows поддерживаются многобайтные имена интерфейсов. Получить описания сетевых интерфейсов в Windows можно 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 счётчиков, если они доступны. 64-битные счётчики статистики интерфейсов были представлены в Windows Vista и Windows Server 2008. Если 64-битные счётчики недоступны, агент использует 32-битные счётчики. Получить описания сетевых интерфейсов в Windows можно 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 агентом 2.	
net.tcp.service.perf	Проверяет производительность TCP-службы. Проверка LDAP и HTTPS в Windows поддерживается только Zabbix агентом 2.	
net.tcp.socket.count	Возвращает количество TCP-сокетов, соответствующих параметрам. Этот элемент данных поддерживается в Linux агентом Zabbix, но в Windows поддерживается только Zabbix агентом 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 поддерживается только Zabbix агентом 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

Item key	Description	Item group
system.cpu.load	Загрузка CPU. Когда в агенте Zabbix запускается процесс collector, инициализируются следующие счётчики производительности, которые затем используются для этого элемента данных: `\System\Processor Queue Length`	
system.cpu.num	Количество CPU.	
system.cpu.util	Процент использования CPU. Значение получается с использованием счётчика производительности *Processor Time*. Обратите внимание, что начиная с Windows 8 её диспетчер задач показывает использование CPU на основе счётчика производительности *Processor Utility*, тогда как в предыдущих версиях использовался счётчик *Processor Time* (см. подробнее). *system* — единственный параметр `type`, поддерживаемый в Windows.	
system.hostname	Имя узла системы. Значение этого элемента данных 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	Выполняет указанную команду на узле сети.	
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, имя дистрибутива, младшая и старшая версии и т. д.).	
system.swap.size	Размер пространства подкачки в байтах или в процентах от общего объёма. Параметр типа `pused` поддерживается в Linux агентом Zabbix, но в Windows поддерживается только Zabbix агентом 2. Обратите внимание, что этот ключ может сообщать некорректный размер/процент пространства подкачки на виртуализированных платформах Windows (VMware ESXi, VirtualBox). В этом случае можно использовать ключ `perf_counter[\700(_Total)\702]` для получения корректного процента пространства подкачки.	
system.uname	Идентификация системы. В Windows значение этого элемента данных получается из WMI-классов 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 непрерывной работы системы в секундах.	
vfs.dir.count	Количество записей в каталоге. В Windows символические ссылки на каталоги пропускаются, а жёсткие ссылки учитываются только один раз.	Виртуальные файловые системы
vfs.dir.get	Список записей каталога. В Windows символические ссылки на каталоги пропускаются, а жёсткие ссылки учитываются только один раз.	
vfs.dir.size	Размер каталога. В Windows любые символические ссылки пропускаются, а жёсткие ссылки учитываются только один раз.	
vfs.file.cksum	Контрольная сумма файла, вычисленная по алгоритму UNIX cksum.	
vfs.file.contents	Получение содержимого файла.	
vfs.file.exists	Проверяет, существует ли файл. В Windows двойные кавычки должны быть экранированы обратной косой чертой '\', а весь ключ элемента данных должен быть заключён в двойные кавычки при использовании утилиты командной строки для вызова zabbix_get.exe или agent2. Обратите внимание, что элемент данных может стать неподдерживаемым в Windows, если каталог ищется внутри несуществующего каталога, например: `vfs.file.exists[C:\no\dir,dir]` (где 'no' не существует).	
vfs.file.get	Возвращает информацию о файле. Поддерживаемые типы файлов в Windows: обычный файл, каталог, символическая ссылка	
vfs.file.md5sum	Контрольная сумма MD5 файла.	
vfs.file.owner	Получает владельца файла.	

Item key	Description	Item group
vfs.file.regexp	Извлекает строку из файла.	
vfs.file.regmatch	Находит строку в файле.	
vfs.file.size	Размер файла.	
vfs.file.time	Информация о времени файла. В Windows XP `vfs.file.time[file,change]` может быть равно `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 объёма.	
vm.memory.size	Размер памяти в байтах или в процентах от общего объёма.	Виртуальная память
web.page.get	Получает содержимое веб-страницы.	Веб-мониторинг
web.page.perf	Время загрузки полной веб-страницы.	
web.page.regexp	Находит строку 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 Zabbix агент 2).	
agent.version	Версия агента Zabbix.	
zabbix.stats	Удалённо возвращает набор внутренних метрик сервера или прокси Zabbix.	
zabbix.stats	Удалённо возвращает количество отслеживаемых элементов данных в очереди, которые задерживаются на сервере или прокси Zabbix.	

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
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.	

Ключ элемента данных	Описание	Группа элементов данных
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 ключа элемента данных

Параметры без угловых скобок являются обязательными. Параметры, отмеченные угловыми скобками **< >**, являются необязательными.

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 важность события (без учета регистра). Этот параметр принимает регулярное выражение на основе следующих значений: ”Information”, ”Warning”, ”Error”, ”Critical”, ”Verbose” (при работе в Windows Vista или новее).
- **source** — регулярное выражение, описывающее идентификатор источника (без учета регистра);
- **eventid** — регулярное выражение, описывающее идентификатор(ы) события (с учетом регистра);
- **maxlines** — максимальное количество новых строк в секунду, которое агент будет отправлять на Zabbix сервер или прокси. Этот параметр переопределяет значение ’MaxLinesPerSecond’ в zabbix_agentd.conf.
- **mode** — возможные значения: *all* (по умолчанию) или *skip* — пропустить обработку 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 типа тип информации, отличного от Log, приведет к потере локальной временной метки, а также информации о важности и источнике события;
- 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` (            *Result is TR
```

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 важности (без учета регистра). Этот параметр принимает регулярное выражение на основе следующих значений: ”Information”, ”Warning”, ”Error”, ”Critical”, ”Verbose” (при работе в Windows Vista или новее).
- **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) или *skip* — пропустить обработку 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-журнального типа информации приведет к потере локальной метки времени, а также информации об уровне важности 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 могут 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 счётчика производительности Windows.
 Возвращаемое значение: *Integer*, *float*, *string* или *text* (в зависимости от запроса).

Параметры:

- **counter** — путь к счётчику;
- **interval** — последние N секунд для хранения среднего значения. `interval` должен быть в диапазоне от 1 до 900 секунд включительно, значение по умолчанию — 1.

Комментарии:

- Этот элемент данных использует UNC-пути, что может представлять угрозу безопасности;
- `interval` используется для счётчиков, которым требуется более одной выборки (например, загрузка CPU), поэтому проверка каждый раз возвращает среднее значение за последние "interval" секунд;
- Для получения списка доступных счётчиков можно использовать Performance Monitor.
- См. также: Счётчики производительности Windows.

perf_counter_en[counter,<interval>]

 Значение любого счётчика производительности Windows на английском языке.
 Возвращаемое значение: *Integer*, *float*, *string* или *text* (в зависимости от запроса).

Параметры:

- **counter** — путь к счётчику на английском языке;
- **interval** — последние N секунд для хранения среднего значения. Значение `interval` должно быть от 1 до 900 секунд включительно, значение по умолчанию — 1.

Комментарии:

- Этот элемент данных использует UNC-пути, что может представлять угрозу безопасности;
- `interval` используется для счётчиков, которым требуется более одной выборки (например, загрузка CPU), поэтому при каждой проверке возвращается среднее значение за последние "interval" секунд;
- Этот элемент данных поддерживается только в **Windows Server 2008/Vista** и выше;
- Список строк на английском языке можно найти, просмотрев следующий раздел реестра: `HKEY_LOCAL_MACHINE\SOFTWARE\Mi NT\CurrentVersion\Perflib\009`.

perf_instance.discovery[object]

 Список экземпляров объектов счётчиков производительности Windows. Используется для обнаружения низкого уровня.
 Возвращаемое значение: *объект JSON*.

Параметр:

- **object** — имя объекта (локализованное).

perf_instance_en.discovery[object]

 Список экземпляров объектов счётчиков производительности Windows, обнаруженных с использованием имён объектов на английском языке. Используется для обнаружения низкого уровня.
 Возвращаемое значение: *объект JSON*.

Параметр:

- **object** — 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 смысл, если существует более одного процесса с одинаковым именем)

Комментарии:

- Поддерживаются следующие `attributes`:
vmsize (по умолчанию) — размер виртуальной памяти процесса в Кбайтах
wkset — размер рабочего набора процесса (объём физической памяти, используемой процессом) в Кбайтах
pf — количество страничных ошибок
ktime — время работы процесса в режиме ядра в миллисекундах
utime — время работы процесса в пользовательском режиме в миллисекундах
io_read_b — количество байт, прочитанных процессом во время операций ввода-вывода
io_read_op — количество операций чтения, выполненных процессом
io_write_b — количество байт, записанных процессом во время операций ввода-вывода
io_write_op — количество операций записи, выполненных процессом
io_other_b — количество байт, переданных процессом во время операций, отличных от 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>;
- В 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] #                                       Internet Explorer
```

registry.data[key,<имя значения>]

 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);
- **имя значения** — имя значения реестра в ключе (по умолчанию пустая строка ””). Значение по умолчанию возвращается, если имя значения не указано.

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"] #
registry.data["HKLM\SOFTWARE\Microsoft\Windows\Windows Error Reporting","EnableZip"] #
```

registry.get[key,<mode>,<name regexp>]

 Список значений или ключей реестра Windows, расположенных по указанному ключу.
 Возвращаемое значение: *объект JSON*.

Параметры:

- **key** — ключ реестра, включая корневой ключ; допускаются сокращения корневых ключей (например, HKLM) (полный список сокращений см. в комментариях для registry.data[]);
- **mode** — возможные значения:
values (по умолчанию) или *keys*;
- **name regexp** — обнаруживать только значения, имена которых соответствуют регулярному выражению (по умолчанию обнаруживаются все значения). Допустимо только при mode, равном *values*.

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 обнаружения низкого уровня.
 Возвращаемое значение: *объект JSON*.

service.info[service,<param>]

 Информация о службе.
 Возвращаемое значение: *Integer* — если param имеет значение *state*, *startup*; *String* — если param имеет значение *displayname*, *path*, *user*; *Text* — если param имеет значение *description*
В частности, для *state*: 0 — работает, 1 — приостановлена, 2 — ожидает запуска, 3 — ожидает приостановки, 4 — ожидает продолжения, 5 — ожидает остановки, 6 — остановлена, 7 — неизвестно, 255 — такая служба не существует
В частности, для *startup*: 0 — автоматически, 1 — автоматически с задержкой, 2 — вручную, 3 — отключена, 4 — неизвестно, 5 — автоматический запуск по триггеру, 6 — автоматический запуск с задержкой по триггеру, 7 — запуск вручную по триггеру

Параметры:

- **service** — фактическое имя службы или ее отображаемое имя, как оно показано в оснастке MMC «Службы»;
- **param** — *state* (по умолчанию), *displayname*, *path*, *user*, *startup* или *description*.

Комментарии:

- Элементы данных вида service.info[service,state] и service.info[service] возвращают одинаковую информацию;
- Только если param имеет значение *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[<type>,<state>,<exclude>]

 Список служб.
 Возвращаемое значение: *0* — если пусто; *Text* — список служб, разделённых символом новой строки.

Параметры:

- **type** — *all* (по умолчанию), *automatic*, *manual* или *disabled*;
- **state** — *all* (по умолчанию), *stopped*, *started*, *start_pending*, *stop_pending*, *running*, *continue_pending*, *pause_pending* или *paused*;
- **exclude** — службы, которые следует исключить из результата. Исключаемые службы должны быть перечислены в двойных кавычках, разделены запятыми, без пробелов.

Примеры:

```
services[,started] #                   ;
services[automatic, stopped] #                     ,                ;
services[automatic, stopped, "service1,service2,service3"] #                    ,                ,
```

vm.vmemory.size[<type>]

 Размер виртуальной памяти в байтах или в процентах от общего объема.
 Возвращаемое значение: *Integer* — для байтов; *float* — для процентов.

Параметр:

- **type** — возможные значения: *available* (доступная виртуальная память), *pavailable* (доступная виртуальная память, в процентах), *pused* (используемая виртуальная память, в процентах), *total* (общий объем виртуальной памяти, по умолчанию) или *used* (используемая виртуальная память)

Комментарии:

- Мониторинг статистики виртуальной памяти основан на:
  - общем объеме виртуальной памяти в Windows (общая физическая память + размер файла подкачки);
  - максимальном объеме памяти, который может зарезервировать агент Zabbix;
  - текущем пределе выделенной памяти для системы или агента Zabbix, в зависимости от того, что меньше.

Пример:

```
vm.vmemory.size[pavailable] #
```

wmi.get[<namespace>,<query>]

 Выполнить WMI-запрос и вернуть первый выбранный объект.
 Возвращаемое значение: *Integer*, *float*, *string* или *text* (в зависимости от запроса).

Параметры:

- **namespace** — пространство имен WMI;
- **query** — WMI-запрос, возвращающий один объект.

WMI-запросы выполняются с помощью WQL.

Пример:

```
wmi.get[root\cimv2,select status from Win32_DiskDrive where Name like '%PHYSICALDRIVE0%'] #
```

wmi.getall[<namespace>,<query>]

 Выполняет WMI-запрос и возвращает полный ответ. Может использоваться для обнаружения низкого уровня.
 Возвращаемое значение: *объект JSON*

Параметры:

- **namespace** — пространство имен WMI;
- **query** — WMI-запрос.

Комментарии:

- WMI-запросы выполняются с помощью WQL.
- Для указания более конкретных значений в возвращаемом JSON можно использовать предобработку JSONPath.

Пример:

```
wmi.getall[root\cimv2,select * from Win32_DiskDrive where Name like '%PHYSICALDRIVE%'] #returns status inf
```

Мониторинг служб Windows

В этом руководстве приведены пошаговые инструкции по настройке мониторинга служб Windows. Предполагается, что сервер Zabbix и агент настроены и работают.

Шаг 1

Получите имя службы.

Имя службы можно узнать, открыв оснастку MMC «Службы» и вызвав свойства нужной службы. На вкладке *Общие* вы должны увидеть поле с названием «Имя службы». 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 мониторинга службы.

Элемент данных `service.info[service,<param>]` получает информацию об определённой службе. В зависимости от того, какая информация вам нужна, укажите параметр `param`, который принимает следующие значения: *displayname*, *state*, *path*, *user*, *startup* или *description*. Значение по умолчанию — *state*, если `param` не указан (`service.info[service]`).

Тип возвращаемого значения зависит от выбранного `param`: integer для *state* и *startup*; character string для *displayname*, *path* и *user*; text для *description*.

Пример:

- *Ключ:* `service.info[lanmanworkstation]`
- *Тип информации:* Числовой (без знака)

Элемент данных `service.info[lanmanworkstation]` будет получать информацию о состоянии службы в виде числового значения. Чтобы сопоставить числовое значение с текстовым представлением в веб-интерфейсе (”0” как ”Running”, ”1” как ”Paused” и т. д.), вы можете настроить преобразование значений на узле сети, на котором настроен элемент данных. Для этого либо привяжите шаблон *Windows services by Zabbix agent* или *Windows services by Zabbix agent active* к узлу сети, либо настройте на узле сети новую карту значений на основе карты значений *Windows service state*, настроенной в указанных шаблонах.

Обратите внимание, что в обоих упомянутых шаблонах настроено правило обнаружения, которое будет автоматически обнаруживать службы. Если это не требуется, вы можете отключить правило обнаружения на уровне узла сети после того, как шаблон будет привязан к узлу сети.

Обнаружение служб Windows

Низкоуровневое обнаружение предоставляет способ автоматически создавать элементы данных, триггеры и графики для различных сущностей на компьютере. Zabbix может автоматически начать мониторинг служб Windows на вашей машине без необходимости знать точное имя службы или вручную создавать элементы данных для каждой службы. Фильтр можно использовать для создания реальных элементов данных, триггеров и графиков только для интересующих служб.

2. SNMP агент

Обзор

Вы можете использовать мониторинг SNMP на таких устройствах, как принтеры, сетевые коммутаторы, маршрутизаторы или UPS, 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 предварительно настроен с поддержкой SNMP с указанием 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.

Демоны сервера Zabbix и прокси записывают в журнал строки, подобные следующей, если получают некорректный ответ SNMP:

```
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 комбинированные запросы.

Сервер Zabbix/прокси будет повторять попытки до 5 раз для SNMP-элементов данных `walk` и `get`. Механизм повторных попыток не применяется к ошибкам разрешения DNS.

Для устаревших проверок SNMP (одно число OID или строка) сервер Zabbix/прокси повторит попытку как минимум один раз после неудачного запроса: либо с помощью механизма повторных попыток библиотеки SNMP, либо с помощью внутреннего механизма комбинированной обработки.

Warning:
Если выполняется мониторинг устройств SNMPv3, убедитесь, что msgAuthoritativeEngineID (также известный как snmpEngineID или ”Engine ID”) никогда не используется совместно двумя устройствами. 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, в результате чего их сообщения SNMP после перезапуска отбрасываются как устаревшие. В такой ситуации необходимо вручную очистить кэш SNMP на сервере/прокси (с помощью -R snmp_cache_reload) или 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 не может быть использован повторно, будет выполнена повторная попытка с probe для обнаружения нового EngineID.

Настройка мониторинга SNMP

Чтобы начать мониторинг устройства через SNMP, необходимо выполнить следующие шаги:

Шаг 1

Определите SNMP-строку (или OID) элемента данных, который вы хотите отслеживать.

Чтобы получить список SNMP-строк, используйте команду **snmpwalk** (часть программного обеспечения net-snmp, которое должно быть установлено в составе установки Zabbix) или аналогичный инструмент:

```
snmpwalk -v 2c -c public <host IP> .
```

Поскольку '2c' здесь обозначает версию SNMP, вы также можете заменить ее на '1', чтобы указать SNMP версии 1 на устройстве.

В результате вы должны получить список SNMP-строк и их последних значений. Если этого не происходит, возможно, что SNMP 'community' отличается от стандартного 'public'; в таком случае вам нужно выяснить, какое 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, вам следует использовать строку `IF-MIB::ifHCInOctets.3` из этой строки:

```
IF-MIB::ifHCInOctets.3 = Counter64: 3409739121
```

Теперь вы можете использовать команду **snmpget**, чтобы узнать 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: Динамические индексы.

Это должно дать вам что-то вроде следующего:

```
.1.3.6.1.2.1.31.1.1.1.6.3 = Counter64: 3472126941
```

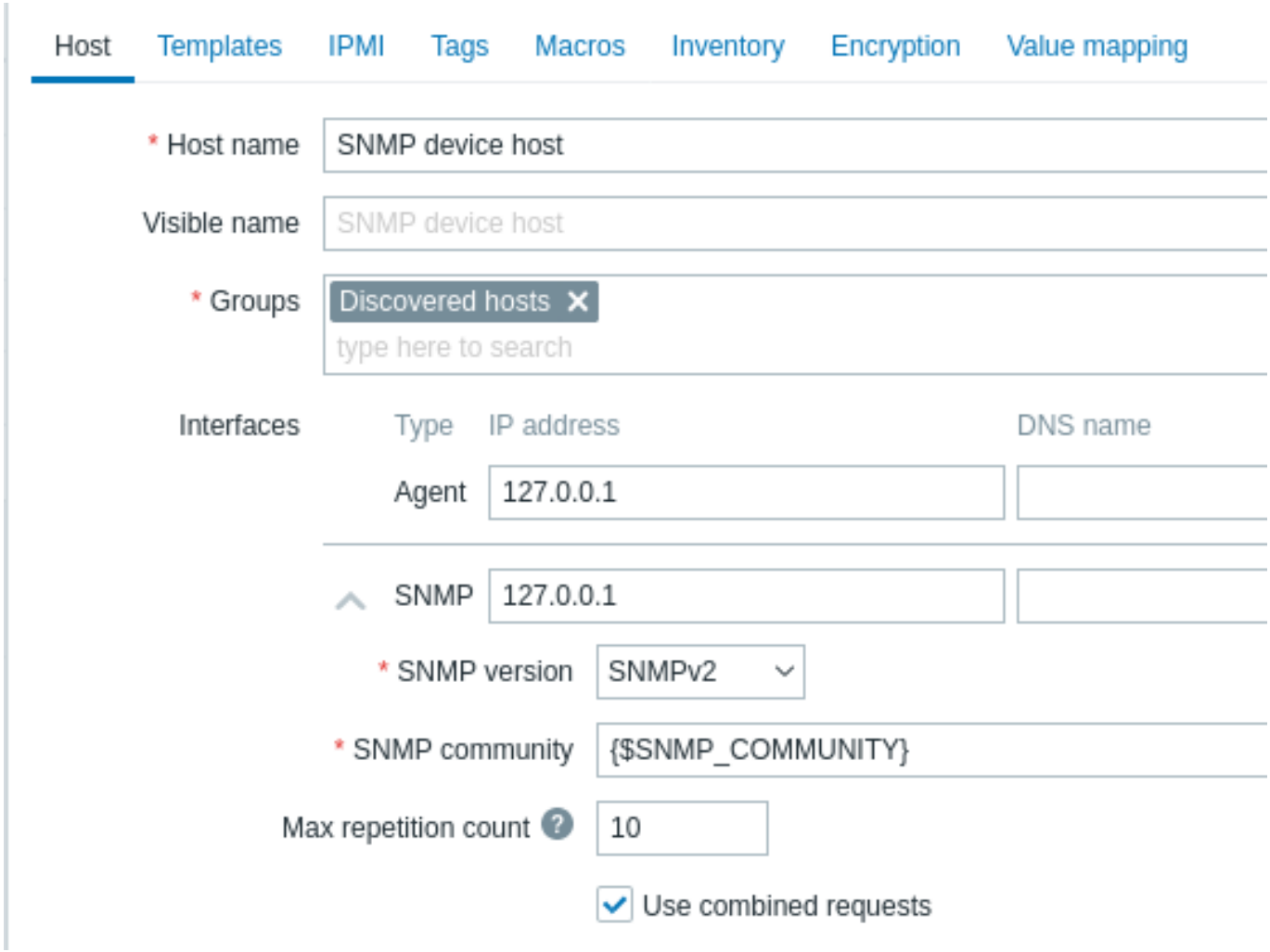
И снова, последнее число в OID — это номер порта.

> **Note:**
> Некоторые из наиболее часто используемых SNMP OID автоматически преобразуются в числовое представление в Zabbix.

В последнем примере выше тип значения — "Counter64", что 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 приблизительно 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 display hint.

Шаг 2

Создайте узел сети, соответствующий устройству.



Добавьте SNMP-интерфейс для узла сети:

- Введите IP-адрес/DNS-имя и номер порта.
- Выберите *версию SNMP* из выпадающего списка.
- Добавьте учетные данные интерфейса в зависимости от выбранной версии SNMP:
  - Для SNMPv1, v2 требуется только community (обычно 'public').
  - Для SNMPv3 требуются более специфические параметры (см. ниже).
- Укажите значение максимального повторения (по умолчанию: 10) для собственных SNMP bulk-запросов (GetBulkRequest-PDUs); только для элементов данных `discovery[]` и `walk[]` в SNMPv2 и v3. Обратите внимание, что установка слишком высокого значения может привести к тайм-ауту проверки SNMP-агента.
- Установите флажок *Использовать комбинированные запросы*, чтобы разрешить комбинированную обработку SNMP-запросов (не относится к собственным SNMP bulk-запросам "walk" и "get").

Параметр SNMPv3	Описание
Имя контекста	Введите имя контекста для идентификации элемента данных в подсети SNMP. Пользовательские макросы разрешаются в этом поле.
Имя безопасности	Введите имя безопасности. Пользовательские макросы разрешаются в этом поле.
Уровень безопасности	Выберите уровень безопасности: **noAuthNoPriv** - не используются ни протоколы аутентификации, ни протоколы приватности **AuthNoPriv** - используется протокол аутентификации, протокол приватности не используется **AuthPriv** - используются и протоколы аутентификации, и протоколы приватности
Протокол аутентификации	Выберите протокол аутентификации - *MD5*, *SHA1*; с net-snmp 5.8 и новее также *SHA224*, *SHA256*, *SHA384* или *SHA512*.
Парольная фраза аутентификации	Введите парольную фразу аутентификации. Пользовательские макросы разрешаются в этом поле.

Параметр SNMPv3	Описание
Протокол приватности	Выберите протокол приватности - *DES*, *AES128*, *AES192*, *AES256*, *AES192C* (Cisco) или *AES256C* (Cisco). См. примечания о поддержке протокола приватности
Парольная фраза приватности	Введите парольную фразу приватности. Пользовательские макросы разрешаются в этом поле.

В случае неверных учетных данных SNMPv3 (имя безопасности, протокол/парольная фраза аутентификации, протокол приватности):

- Zabbix получает ERROR от net-snmp, за исключением неверной *парольной фразы приватности*, в этом случае Zabbix получает ошибку TIMEOUT от net-snmp.
- Доступность SNMP-интерфейса переключится на красный цвет (недоступен).

Note:
Изменения в *протоколе аутентификации*, *парольной фразе аутентификации*, *протоколе приватности* или *парольной фразе приватности*, выполненные без изменения *имени безопасности*, обычно применяются автоматически при обновлении соответствующего интерфейса SNMPv3 в Zabbix. В случаях, когда также изменяется *имя безопасности*, все параметры будут обновлены немедленно.

Вы можете использовать один из предоставляемых шаблонов SNMP, который автоматически добавит набор элементов данных. Перед использованием шаблона убедитесь, что он совместим с узлом сети.

Нажмите *Добавить*, чтобы сохранить узел сети.

Поддержка протоколов конфиденциальности

В зависимости от вашей операционной системы и конфигурации net-snmp некоторые протоколы конфиденциальности могут быть недоступны:

- В некоторых новых операционных системах (например, RHEL9) поддержка DES исключена из пакета net-snmp.
- Протоколы шифрования AES192 и более сильные не поддерживаются «из коробки» в операционных системах старше 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`, AES192+ поддерживается.

Обратите внимание, что net-snmp-config является частью пакета разработки для SNMP (libsnmp-dev для Debian/Ubuntu, net-snmp-devel для CentOS/RHEL/OL/SUSE) и может быть не 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.

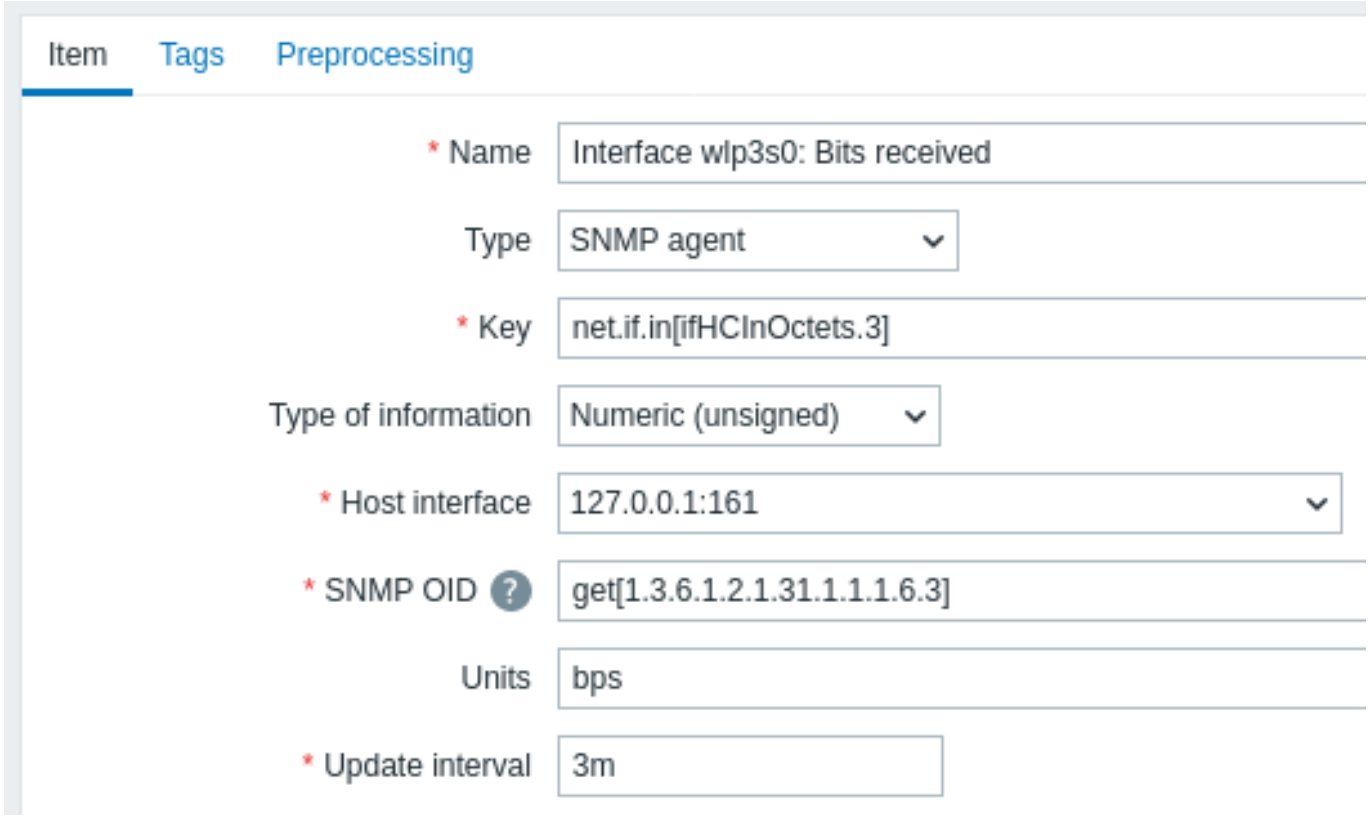
Если ваша библиотека net-snmp не поддерживает протоколы AES192 и выше, перекомпилируйте net-snmp с параметром `--enable-blumenthal-aes`, затем перекомпилируйте сервер Zabbix, указав параметр `--with-net-snmp=/home/user/yourcusto`

Шаг 3

Создайте элемент данных для мониторинга.

Теперь вернитесь в Zabbix и нажмите *Items* для SNMP-узла сети, который вы создали ранее. В зависимости от того, использовали ли вы шаблон при создании узла сети, у вас будет либо список SNMP-элементов данных, связанных с вашим узлом сети, либо просто пустой список. Мы будем исходить из того, что вы собираетесь создать элемент данных самостоятельно, используя информацию, которую вы только что получили с помощью snmpwalk и snmpget, поэтому нажмите *Create item*.

Заполните обязательные параметры в форме нового элемента данных:



Параметр	Описание
Name	Введите имя элемента данных.
Type	Выберите здесь **SNMP agent**.
Key	Введите ключ с понятным смыслом.
Host interface	Убедитесь, что выбрали SNMP-интерфейс, например интерфейс вашего коммутатора/маршрутизатора.

Параметр	Описание
SNMP OID	Используйте один из поддерживаемых форматов для ввода значения(й) OID: **walk[OID1,OID2,...]** — получить поддерево значений. Например: `walk[1.3.6.1.2.1.2.2.1.2,1.3.6.1.2.1.2.2.1.3]`. Этот вариант использует собственные SNMP bulk-запросы (GetBulkRequest-PDUs) асинхронно. Настройки тайм-аута для этого элемента данных можно задать в форме конфигурации элемента данных. Рекомендуется установить небольшое значение тайм-аута, чтобы избежать длительных задержек, если устройство недоступно, так как будет выполнено до 5 повторных попыток, если предыдущие завершились по тайм-ауту или с ошибкой (например, тайм-аут в 3 секунды может привести к ожиданию в 15 секунд). Вы можете использовать это как мастер-элемент данных, а зависимые элементы данных будут извлекать данные из мастер-элемента с помощью предобработки. Можно указать несколько OID в одном snmp walk, например `walk[OID1,OID2,...]`, чтобы асинхронно обрабатывать по одному OID за раз. Если bulk-запрос не возвращает результатов, будет предпринята попытка получить одну запись без bulk-запроса. Имена MIB поддерживаются в качестве параметров; таким образом, `walk[1.3.6.1.2.1.2.2.1.2]` и `walk[ifDescr]` вернут одинаковый вывод. Если указано несколько OID/MIB, то есть `walk[ifDescr,ifType,ifPhysAddress]`, то вывод будет представлять собой объединенный список. Запросы GetBulk используются с интерфейсами SNMPv2 и v3, а GetNext — для интерфейсов SNMPv1; максимальное число повторений для bulk-запросов настраивается на уровне интерфейса. Параметр максимального числа повторений влияет на bulk-запросы, определяя максимальное количество OID, возвращаемых в одном bulk-ответе. Более высокое значение приводит к более крупным bulk-ответам, уменьшая количество необходимых передач. Однако не все устройства могут поддерживать очень высокие значения, что может вызвать проблемы. Этот элемент данных возвращает вывод утилиты snmpwalk с параметрами -Oe -Ot -On. Вы можете использовать этот элемент данных как мастер-элемент данных в SNMP-обнаружении. **get[OID]** — получить *одно* значение асинхронно. Например: `get[1.3.6.1.2.1.31.1.1.1.6.3]` Настройки тайм-аута для этого элемента данных можно задать в форме конфигурации элемента данных. Рекомендуется установить небольшое значение тайм-аута, чтобы избежать длительных задержек, если устройство недоступно, так как будет выполнено до 5 повторных попыток, если предыдущие завершились по тайм-ауту или с ошибкой (например, тайм-аут в 3 секунды может привести к ожиданию в 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 тайм-аут проверки элемента данных будет равен значению, заданному в файле конфигурации сервера. **Рекомендуется** использовать элементы данных `walk[OID]` и `get[OID]` для лучшей производительности. Все элементы данных `walk[OID]` и `get[OID]` выполняются асинхронно — не требуется получать ответ на один запрос до запуска других проверок. Разрешение DNS также выполняется асинхронно. Максимальная параллельность асинхронных проверок составляет 1000 (определяется параметром MaxConcurrentChecksPerPoller). Количество асинхронных SNMP poller задается параметром StartSNMPPollers. Обратите внимание, что для статистики сетевого трафика, возвращаемой любым из методов, на вкладке *Preprocessing* необходимо добавить шаг *Change per second*; в противном случае вы получите накопительное значение с SNMP-устройства вместо последнего изменения.

Все обязательные поля ввода отмечены красной звездочкой.

Теперь сохраните элемент данных и перейдите в *Monitoring* > *Latest data* для ваших SNMP-данных.

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, используемая как ссылка на триггеры> Например, ”my_param”.

Обратите внимание, что OID может быть указан как в числовой, так и в строковой форме. Однако в некоторых случаях строковый OID необходимо преобразовать в числовое представление. Для этой цели 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

Встроенные SNMP bulk-запросы

Элемент данных **walk[OID1,OID2,...]** позволяет использовать встроенную функциональность SNMP для bulk-запросов (GetBulkRequest-PDUs), доступную в SNMP версий 2/3.

Запрос GetBulk в SNMP выполняет несколько запросов GetNext и возвращает результат в одном ответе. Это может использоваться как для обычных SNMP-элементов данных, так и для SNMP-обнаружения, чтобы минимизировать количество сетевых обменов.

SNMP-элемент данных **walk[OID1,OID2,...]** может использоваться как основной элемент данных, который собирает данные одним запросом, с зависимыми элементами данных, разбирающими ответ по мере необходимости с помощью предобработки.

Обратите внимание, что использование встроенных SNMP bulk-запросов не связано с опцией объединения SNMP-запросов, которая является собственным способом Zabbix объединять несколько SNMP-запросов (см. следующий раздел).

Для SNMP bulk-элементов данных будет выполнено до пяти повторных попыток, чтобы избежать сбоя, если один из пакетов будет потерян. Тайм-аут для SNMP-элементов данных с `get` и `walk` (задается в форме настройки элемента данных) устанавливается для всей сессии. Тайм-аут применяется независимо от того, получены ли данные полностью; если данные получены частично (например, данные успешно собраны только для одного из нескольких OID), то элемент данных становится неподдерживаемым с сообщением ”Only partial data received”. Если достигнут тайм-аут, будет выполнена повторная попытка, тайм-аут будет сброшен, а последний запрос будет отправлен повторно, что позволит продолжить сессию с последнего запроса, если один пакет был потерян или пришел слишком поздно. Рекомендуется устанавливать небольшое значение тайм-аута, чтобы избежать длительных задержек, если устройство недоступно, так как будет выполнено до 5 повторных попыток, если предыдущие завершились по тайм-ауту или неудачей (например, тайм-аут 3 секунды может привести к ожиданию в 15 секунд).

Внутренний механизм комбинированной обработки

Сервер Zabbix и прокси могут опрашивать устройства SNMP для получения нескольких значений в одном запросе. Это влияет на несколько типов SNMP-элементов данных:

- обычные SNMP-элементы данных
- SNMP-элементы данных с динамическими индексами
- SNMP-правила низкоуровневого обнаружения

Все SNMP-элементы данных на одном интерфейсе с одинаковыми параметрами планируются для опроса одновременно. Первые два типа элементов данных обрабатываются опрашивателями пакетами максимум по 128 элементов данных, тогда как правила низкоуровневого обнаружения, как и раньше, обрабатываются по отдельности.

На более низком уровне для запроса значений выполняются два вида операций: получение нескольких указанных объектов и обход дерева OID.

Для «получения» используется GetRequest-PDU максимум с 128 привязками переменных. Для «обхода» используется GetNextRequest-PDU для SNMPv1, а для SNMPv2 и SNMPv3 используется GetBulkRequest с полем ”max-repetitions” не более 128.

Таким образом, преимущества комбинированной обработки для каждого типа SNMP-элементов данных приведены ниже:

- обычные SNMP-элементы данных получают преимущества от улучшений «получения»;
- SNMP-элементы данных с динамическими индексами получают преимущества как от улучшений «получения», так и от улучшений «обхода»: «получение» используется для проверки индекса, а «обход» — для построения кэша;
- SNMP-правила низкоуровневого обнаружения получают преимущества от улучшений «обхода».

Однако существует техническая проблема: не все устройства способны возвращать 128 значений на один запрос. Некоторые всегда возвращают корректный ответ, но другие либо отвечают ошибкой ”tooBig(1)”, либо вообще не отвечают, если потенциальный ответ превышает определённый предел.

Чтобы найти оптимальное количество объектов для запроса у конкретного устройства, Zabbix использует следующую стратегию. Сначала он действует осторожно и запрашивает 1 значение в одном запросе. Если это успешно, он запрашивает 2 значения в одном запросе. Если это снова успешно, он запрашивает 3 значения в одном запросе и продолжает аналогичным образом, умножая количество запрашиваемых объектов на 1.5, что приводит к следующей последовательности размеров запросов: 1, 2, 3, 4, 6, 9, 13, 19, 28, 42, 63, 94, 128.

Однако как только устройство отказывается возвращать корректный ответ (например, для 42 переменных), Zabbix делает две вещи.

Во-первых, для текущего пакета элементов данных он уменьшает количество объектов в одном запросе вдвое и запрашивает 21 переменную. Если устройство доступно, то в подавляющем большинстве случаев такой запрос должен сработать, потому что известно, что 28 переменных работали, а 21 значительно меньше этого значения. Однако если и это не сработает, тогда Zabbix переключается на запрос значений по одному. Если и на этом этапе запрос не удаётся, значит устройство определённо не отвечает, и размер запроса тут ни при чём.

Во-вторых, для последующих пакетов элементов данных Zabbix начинает с последнего успешного количества переменных (28 в нашем примере) и продолжает увеличивать размер запросов на 1, пока не будет достигнут предел. Например, если предположить, что максимальный размер ответа составляет 32 переменные, то последующие запросы будут иметь размеры 29, 30, 31, 32 и 33. Последний запрос завершится неудачей, 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 завершаются неудачей при таком количестве переменных, это может означать одно из двух. Точные критерии, которые устройство использует для ограничения размера ответа, неизвестны, но мы пытаемся приблизительно оценить это по количеству переменных. Поэтому первая возможность состоит в том, что это количество переменных находится около фактического предела размера ответа устройства в общем случае: иногда ответ меньше предела, а иногда больше него. Вторая возможность состоит в том, что UDP-пакет просто был потерян в одном из направлений. По этим причинам, если Zabbix получает неудачный запрос, он уменьшает максимальное количество переменных, чтобы попытаться уйти глубже в «комфортный» диапазон устройства, но не более двух раз.

В приведённом выше примере, если запрос с 32 переменными случайно завершится неудачей, Zabbix уменьшит количество до 31. Если и этот запрос завершится неудачей, Zabbix уменьшит количество до 30. Однако Zabbix не будет уменьшать количество ниже 30, потому что будет считать, что дальнейшие сбои 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 помогает, для каждого интерфейса предусмотрен параметр ”Use combined requests”, который позволяет отключить комбинированные запросы для этого устройства.

Если комбинированные запросы вызывают частичные или некорректно сформированные ответы, что приводит к неправильным вычислениям значений в секунду (delta) (например, к видимым всплескам счётчиков интерфейса), отключите *Use combined requests* для затронутого интерфейса, чтобы принудительно выполнять отдельные запросы для каждого элемента данных; это часто предотвращает ложные всплески. В качестве альтернативы рассмотрите использование асинхронных элементов данных `get[]` или `walk[]`, которые выполняются асинхронно и не подпадают под пакетную обработку *Use combined requests* на уровне интерфейса — их можно использовать вместо устаревших синхронных проверок OID, чтобы избежать проблем, связанных с комбинированными запросами. Чтобы выявить затронутые устройства, ищите записи в журнале сервера/прокси, похожие на показанную в разделе Обзор.

Кроме того, если интерфейс часто становится недоступным, может потребоваться увеличить параметр `UnavailableDelay` в конфигурационных файлах сервера Zabbix или прокси Zabbix, чтобы уменьшить частоту запросов. Элементы данных могут стать неподдерживаемыми, если во время обнаружения или обхода OID получены частичные данные.

1. Динамические индексы

Обзор

Хотя вы может быть и найдёте требуемый номер индекса (например, сетевого интерфейса) из всех 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 ссылается 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.

2. Специальные OID'ы

Некоторые OID’ы, наиболее часто используемые в SNMP, автоматически преобразуются в числовые представления самим Zabbix. Например, **ifIndex** преобразуется в **1.3.6.1.2.1.2.2.1.1**, **ifIndex.0** преобразуется в **1.3.6.1.2.1.2.2.1.1.0**.

В таблице содержится перечень специальных OID’ов.

Специальный OID	Идентификатор	Описание
ifIndex	1.3.6.1.2.1.2.2.1.1	Уникальное значение каждого интерфейса.
ifDescr	1.3.6.1.2.1.2.2.1.2	Текстовая строка, которая содержит информацию об интерфейсе. Эта строка может включать в себя название компании-производителя, имя продукта и версию аппаратного интерфейса.
ifType	1.3.6.1.2.1.2.2.1.3	Тип интерфейса, выделяется в соответствии с физическим/канальным протоколом(ами) непосредственно «под» сетевым уровнем стека сетевых протоколов.
ifMtu	1.3.6.1.2.1.2.2.1.4	Размер наибольшей дейтаграммы, которую может отправить/получить интерфейс, указывается в октетах
ifSpeed	1.3.6.1.2.1.2.2.1.5	Оценочная скорость текущей пропускной способности интерфейса в битах за секунду.
ifPhysAddress	1.3.6.1.2.1.2.2.1.6	Физический адрес интерфейса на канальном уровне непосредственно «под» сетевым уровнем в стеке сетевых 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 число октетов, полученных на интерфейсе, включая символы кадрирования.
ifInUcastPkts	1.3.6.1.2.1.2.2.1.11	Количество пакетов одноадресной рассылки, доставленных протоколу более высокого уровня.
ifInNUcastPkts	1.3.6.1.2.1.2.2.1.12	Количество пакетов НЕ одноадресной рассылки (т.е. широковещательных или многоадресных), доставленных протоколу более высокого уровня.
ifInDiscards	1.3.6.1.2.1.2.2.1.13	Количество входящих, но отвергнутых пакетов, даже если не было обнаружено ошибок, мешающих доставке пакетов протоколу более высокого уровня. Одной из возможных причин для отвержения пакета могло быть освобождение места в буфере.
ifInErrors	1.3.6.1.2.1.2.2.1.14	Количество входящих пакетов, полученных с ошибкой, из-за которой пакеты не были доставлены протоколу более высокого уровня.
ifInUnknownProtos	1.3.6.1.2.1.2.2.1.15	Количество пакетов, полученных через интерфейс, но отвергнутых из-за неизвестного или неподдерживаемого протокола.
ifOutOctets	1.3.6.1.2.1.2.2.1.16	Общее количество отправленных октетов с интерфейса, включая символы кадрирования.
ifOutUcastPkts	1.3.6.1.2.1.2.2.1.17	Общее количество пакетов, которые пытался отправить верхний уровень стека сетевых протоколов, и которые адресованы НЕ на многоадресный или широковещательный адрес на этом подуровне, включая те, которые были отвергнуты или не отправлены.
ifOutNUcastPkts	1.3.6.1.2.1.2.2.1.18	Общее количество пакетов, которые пытался отправить верхний уровень стека сетевых протоколов, и которые адресованы на многоадресный или широковещательный адрес на этом подуровне, включая те, которые были отвергнуты или не отправлены.
ifOutDiscards	1.3.6.1.2.1.2.2.1.19	Количество исходящих пакетов, которые были отвергнуты, даже если не было обнаружено ошибок, мешающих отправке. Одной из возможных причин отвержения пакета могло быть освобождение места в буфере.
ifOutErrors	1.3.6.1.2.1.2.2.1.20	Количество исходящих пакетов, которые не смогли быть отправлены из-за ошибок.
ifOutQLen	1.3.6.1.2.1.2.2.1.21	Длина очереди исходящих пакетов (в пакетах).

3. MIB файлы

Введение

MIB обозначает Базу Управляющей Информации (Management Information Base). MIB файлы позволяют вам использовать текстовое представление OID'а (Идентификатора Объекта). При мониторинге Zabbix'ом устройств SNMP можно использовать сырые OID'ы, но если для Вас более комфортно использовать текстовое представление, то потребуется установить MIB файлы.

Например,

`ifHCOutOctets`

является текстовым представлением OID'а

`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 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 помогут 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.

3. SNMP трапы

Обзор

Получение SNMP-трапов является противоположностью опроса устройств с поддержкой SNMP.

В этом случае информация отправляется с устройства с поддержкой SNMP в snmptrapd и собирается, или «перехватывается», сервером Zabbix или прокси Zabbix из файла.

Обычно трапы отправляются при изменении некоторого условия, и агент подключается к серверу через порт 162 (в отличие от порта 161 на стороне агента, который используется для опросов). Использование трапов позволяет обнаруживать кратковременные проблемы, возникающие в промежутках между опросами и которые могут быть пропущены данными опроса.

Получение SNMP-трапов в Zabbix рассчитано на работу с **snmptrapd** и одним из механизмов передачи трапов в Zabbix — либо Bash- или Perl-скриптом, либо SNMPTT.

Note:
Самый простой способ настроить мониторинг трапов после настройки 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 решение на встроенном Perl (либо скрипт с опцией `do perl`, либо SNMPTT).

Процесс получения трапа:

1. `snmptrapd` получает трап
2. `snmptrapd` передает трап скрипту-приемнику (Bash, Perl) или SNMPTT
3. Приемник разбирает, форматирует и записывает трап в файл
4. SNMP trapper Zabbix читает и разбирает файл трапов
5. Для каждого трапа Zabbix находит все элементы данных "SNMP trapper" с интерфейсами узла сети, соответствующими адресу полученного трапа. Обратите внимание, что при сопоставлении используется только выбранное значение "IP" или "DNS" в интерфейсе узла сети.
6. Для каждого найденного элемента данных трап сравнивается с регулярным выражением в `snmptrap[regexp]`. Трап устанавливается как значение **всех** совпавших элементов данных. Если совпадающий элемент данных не найден и существует элемент данных `snmptrap.fallback`, трап устанавливается как его значение.
7. Если трап не был установлен как значение какого-либо элемента данных, Zabbix по умолчанию записывает несовпавший трап в журнал. (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-ловушек

Для этого типа элемента данных требуется следующая настройка веб-интерфейса.

1. Создайте SNMP-интерфейс для вашего узла сети

- В разделе *Data collection* > *Hosts* создайте/отредактируйте узел сети и в поле *Interfaces* добавьте интерфейс типа "SNMP", указав IP-адрес или DNS-имя.

Адрес из каждой полученной ловушки будет сравниваться с IP- и DNS-адресами всех SNMP-интерфейсов для поиска соответствующих узлов сети.

2. Настройте элемент данных

- В разделе *Data collection* > *Hosts* создайте/отредактируйте необходимый элемент данных.
- В поле *Key* используйте один из ключей SNMP-ловушек:

Key		
Описание	Возвращаемое значение	Комментарии
snmptrap[regexp] Перехватывает все SNMP-ловушки, соответствующие регулярному выражению, указанному в **regexp**. Если regexp не указан, перехватывает любую ловушку.	SNMP-ловушка	Этот элемент данных можно задать только для SNMP-интерфейсов. В параметре этого ключа элемента данных поддерживаются пользовательские макросы и глобальные регулярные выражения.
snmptrap.fallback Перехватывает все SNMP-ловушки, которые не были перехвачены ни одним из элементов данных snmptrap[] для этого интерфейса.	SNMP-ловушка	Этот элемент данных можно задать только для SNMP-интерфейсов.

Note:
Сопоставление с многострочными регулярными выражениями в настоящее время не поддерживается.

- Установите *Type of information* в значение "Log", чтобы временные метки разбирались автоматически. Другие форматы, такие как "Numeric", также допустимы, но могут потребовать пользовательского обработчика ловушек.

Настройка мониторинга SNMP трапов

Настройка сервера/прокси Zabbix

Чтобы читать ловушки, сервер Zabbix или прокси должны быть настроены на запуск процесса SNMP trapper и указание файла ловушек, в который выполняется запись SNMPTT или Bash/Perl-приёмником ловушек. 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 Bash-приёмника ловушек

Требования: только snmptrapd.

Скрипт Bash-приёмника ловушек может использоваться для передачи ловушек на сервер Zabbix из snmptrapd с использованием файла trapper. Чтобы настроить его, добавьте параметр `traphandle` в файл конфигурации snmptrapd (`snmptrapd.conf`), см. пример.

Note:
Для применения изменений в конфигурации может потребоваться перезапуск snmptrapd.

Настройка получателя Perl трапов

Требования: Perl, скомпилированный с --enable-embedded-perl Net-SNMP (компилируется по умолчанию, начиная с Net-SNMP 5.4)

Получатель трапов Perl (ищите в misc/snmptrap/zabbix_trap_receiver.pl) можно использовать для передачи трапов Zabbix серверу напрямую с snmptrapd. Для его настройки:

- добавьте Perl скрипт в файл конфигурации snmptrapd (snmptrapd.conf), например:

```
perl do "[           PERL     -     ]";
```

- настройте сам получатель, например:

```
$SNMPTrapperFile = '[TRAP   ]';
$DateTimeFormat = '[      /    ]';
```

Note:
snmptrapd может потребоваться перезапустить, чтобы изменения в его конфигурации вступили в силу.

Note:
Если имя скрипта не заключено в кавычки, snmptrapd откажется запускаться с сообщениями наподобие этих:

```
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

Для начала, snmptrapd необходимо настроить на использование SNMPTT.

Note:
Для лучшей производительности, SNMPTT нужно сконфигурировать как демон, используя **snmptthandler-embedded** для передачи ему трапов. Смотрите инструкции по настройке SNMPTT.

Когда SNMPTT настроен на получение трапов, настройте `snmptt.ini`:

1. включите использование модуля Perl из пакета NET-SNMP:

```
net_snmp_perl_enable = 1
```

2. включите журналирование трапов в файл с трапами, который будет читаться Zabbix'ом:

```
log_enable = 1
log_file = [TRAP   ]
```

3. задайте формат даты/времени:

```
date_time_format = %Y-%m-%dT%H:%M:%S%z
```

Warning:
Пакет «net-snmp-perl» удалён в RHEL 8.0-8.2; добавлен заново в RHEL 8.3. Для получения более подробной информации смотрите известные проблемы.

Теперь задайте формат трапов, чтобы они распознавались Zabbix'ом (отредактируйте snmptt.conf):

1. Каждая инструкция FORMAT должна начинаться с "ZBXTRAP [адрес]", где [адрес] будет сравниваться с IP и DNS адресами у 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 трапов смотрите ниже.

Attention:
Не используйте неизвестные трапы — Zabbix может их не распознать. Неизвестные трапы могут быть обработаны, если задать 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] — штамп времени в формате «%Y-%m-%dT%H:%M:%S%z»
- ZBXTRAP — заголовок, который указывает, что с этой строки начался новый трап
- [адрес] — IP адрес, используемый для поиска узла сети для этого трапа

Обратите внимание, что «ZBXTRAP» и «[адрес]» 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.
```

Системные требования

Note:
Рекомендуется установить MIB-файлы, чтобы значения элементов данных отображались в правильном формате. Без MIB-файлов могут возникать проблемы с форматированием, например отображение значений в HEX вместо UTF-8 или наоборот.

Поддержка больших файлов

У Zabbix поддерживает большие файлы при работе с файлами SNMP трапов. Максимальный размер файла, который Zabbix может прочитать, — это 2^63 (8 эксабайт). Обратите внимание, что файловая система может иметь меньшее ограничение на максимальный размер файлов.

Ротация журналов

Zabbix не предоставляет никакой системы ротации журналов — это должен обрабатывать пользователь. При ротации журналов сначала следует переименовать старый файл и только потом удалить его, чтобы не потерялись никакие трапы:

1. Zabbix открывает файл трапов в последнем известном расположении и переходит к шагу 3
2. Zabbix проверяет, был ли ротирован текущий открытый файл, сравнивая номер inode с номером inode заданного файла трапов. Если открытого файла нет, Zabbix сбрасывает последнее расположение и переходит к шагу 1.
3. Zabbix читает данные из текущего открытого файла и устанавливает новое расположение.
4. Новые данные разбираются. Если это был ротированный файл, файл закрывается, и выполняется возврат к шагу 2.
5. Если новых данных нет, Zabbix ожидает 1 секунду и возвращается к шагу 2.

Файловая система

Из-за реализации файла с трапами, для Zabbix требуется файловая система с поддержкой inode для того чтобы различать файлы (эта информация берётся из вызова stat()).

Примеры настройки с использованием разных версий протокола SNMP

В этом примере используются snmptrapd и Bash-скрипт-приемник для передачи трапов на сервер Zabbix.

Настройка:

1. Настройте Zabbix на запуск SNMP trapper и укажите файл трапов. Добавьте в `zabbix_server.conf`:

```
StartSNMPTrapper=1
SNMPTrapperFile=/var/lib/zabbix/snmptraps/snmptraps.log
```

2. Загрузите Bash-скрипт в /usr/sbin/zabbix_trap_handler.sh:

```
curl -o /usr/sbin/zabbix_trap_handler.sh https://raw.githubusercontent.com/zabbix/zabbix-docker/trunk/Dock
```

При необходимости скорректируйте переменную ZABBIX_TRAPS_FILE в скрипте. Чтобы использовать значение по умолчанию, сначала создайте родительский каталог:

```
mkdir -p /var/lib/zabbix/snmptraps
```

3. Добавьте следующее в snmtrapd.conf (см. рабочий пример)

```
traphandle default /bin/bash /usr/sbin/zabbix_trap_handler.sh
```

> **Note:**
> Для применения изменений в конфигурации snmptrapd может потребоваться перезапуск.

4. Создайте SNMP элемент данных TEST (имейте в виду начальные 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 мы настроим snmptrapd для выбранной версии протокола SNMP и отправим тестовые трапы с помощью утилиты snmptrap.

SNMPv1, SNMPv2

Протоколы SNMPv1 и SNMPv2 используют аутентификацию по "community string". В примере ниже мы будем использовать "secret" в качестве community string. Оно должно быть установлено в то же значение на отправителях SNMP trap.

Обратите внимание, что хотя SNMPv2 по-прежнему широко используется в производственных средах, он не предоставляет ни шифрования, ни реальной аутентификации отправителя. Данные передаются в открытом виде, поэтому эти версии протокола следует использовать только в защищенных средах, таких как частная сеть, и никогда не использовать в публичных сетях или сетях третьих сторон.

Версия SNMP 1 в наши дни практически не используется, поскольку не поддерживает 64-битные счетчики и считается устаревшим протоколом.

Чтобы включить прием трапов SNMPv1 или SNMPv2, добавьте следующую строку в snmptrapd.conf. Замените "secret" на SNMP community string, настроенную на отправителях SNMP trap:

```
authCommunity log,execute,net secret
```

Далее мы можем отправить тестовый трап с помощью snmptrap. В этом примере мы будем использовать общий OID "link up":

```
snmptrap -v 2c -c secret localhost '' linkUp.0
```

SNMPv3

SNMPv3 решает проблемы безопасности SNMPv1/v2 и предоставляет аутентификацию и шифрование. Вы можете использовать MD5 или несколько методов аутентификации SHA, а также DES/несколько вариантов AES в качестве шифра.

Чтобы включить прием SNMPv3, добавьте следующие строки в snmptrapd.conf:

```
createUser -e 0x8000000001020304 traptest SHA mypassword AES
authuser log,execute traptest
```

> **Attention:**
> Обратите внимание на ключевое слово "execute", которое позволяет выполнять скрипты для этой модели безопасности пользователя.

```
snmptrap -v 3 -n "" -a SHA -A mypassword -x AES -X mypassword -l authPriv -u traptest -e 0x80000000010203
```

Warning:
Если вы хотите использовать стойкие методы шифрования, такие как AES192 или AES256, используйте net-snmp версии 5.8 или новее. Возможно, вам 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 вы увидите похожие строки в `/var/lib/zabbix/snmptraps/snmptraps.log`:

```
2024-01-30T10:04:23+0200 ZBXTRAP 127.0.0.1
UDP: [127.0.0.1]:56585->[127.0.0.1]:162
DISMAN-EVENT-MIB::sysUpTimeInstance = 2538834
SNMPv2-MIB::snmpTrapOID.0 = IF-MIB::linkUp.0
```

Значение элемента данных в Zabbix будет следующим:

```
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]

4. Проверки IPMI

Обзор

В Zabbix можно отслеживать состояние и доступность устройств Intelligent Platform Management Interface (IPMI). Для выполнения IPMI-проверок сервер Zabbix должен быть предварительно настроен с поддержкой IPMI.

IPMI — это стандартизованный интерфейс для удаленного управления компьютерными системами в режиме "lights-out" или "out-of-band". Он позволяет отслеживать состояние оборудования напрямую с так называемых карт управления "out-of-band", независимо от операционной системы и даже от того, включена ли вообще машина.

IPMI-мониторинг в Zabbix работает только для устройств с поддержкой IPMI (HP iLO, DELL DRAC, IBM RSA, Sun SSP и т. д.).

Процесс менеджера IPMI планирует IPMI-проверки, выполняемые IPMI-опросчиками. Узел сети всегда опрашивается только одним IPMI-опросчиком одновременно, что уменьшает количество открытых соединений с контроллерами BMC. Таким образом, можно безопасно увеличивать количество IPMI-опросчиков, не беспокоясь о перегрузке контроллеров BMC. Процесс менеджера IPMI автоматически запускается, когда запущен хотя бы один IPMI-опросчик.

См. также известные проблемы для IPMI-проверок.

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.

Настройка сервера

По умолчанию, Zabbix сервер не запускает IPMI поллеры, поэтому любые добавленные элементы данных IPMI не будут работать. Чтобы изменить это, откройте файл конфигурации (zabbix_server.conf) Zabbix сервера из-под root и найдите следующую строку:

```
# StartIPMIPollers=0
```

Раскомментируйте эту строку и задайте количество поллеров, скажем, равное 3 так, чтобы строка была такой:

```
StartIPMIPollers=3
```

Сохраните файл, после чего 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*

Тайм-аут и завершение сеанса

Тайм-ауты сообщений IPMI и количество повторных попыток определяются в библиотеке OpenIPMI. Из-за текущей архитектуры OpenIPMI невозможно сделать эти значения настраиваемыми в Zabbix ни на уровне интерфейса, ни на уровне элемента данных.

Тайм-аут неактивности сеанса IPMI для LAN составляет 60 +/-3 секунды. В настоящее время невозможно реализовать периодическую отправку команды Activate Session с помощью OpenIPMI. Если от Zabbix не выполняются проверки элементов данных IPMI к определённому BMC дольше, чем тайм-аут сеанса, настроенный в BMC, то следующая проверка IPMI после истечения тайм-аута завершится по тайм-ауту из-за индивидуальных тайм-аутов сообщений, повторных попыток или ошибки приёма. После этого будет открыта новая сессия и инициировано полное повторное сканирование BMC. Для управления новым сеансом может быть открыт новый UDP-порт. Неактивность определяется отсутствием как исходящих запросов, так и входящих ответов. Если вы хотите избежать ненужных повторных сканирований BMC, рекомендуется установить интервал опроса элементов данных IPMI меньше тайм-аута неактивности сеанса IPMI, настроенного в BMC.

Примечания по дискретным датчикам IPMI

Чтобы найти датчики на узле сети, запустите сервер Zabbix с включённым **DebugLevel=4**. Подождите несколько минут и найдите записи об обнаружении датчиков в журнале 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
```

Для расшифровки типов и состояний датчиков IPMI доступна копия спецификации IPMI 2.0 (обратите внимание, что дальнейшие обновления спецификации IPMI не планируются).

Первый параметр, с которого следует начать, — это "reading_type". Используйте "Table 42-1, Event/Reading Type Code Ranges" из спецификации для расшифровки кода "reading_type". Большинство датчиков в нашем примере имеют "reading_type:0x1", что означает датчик "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 "аналоговыми" датчиками, так как они измеряют непрерывные параметры, такие как температура, напряжение, число оборотов 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 значащих битов). Например, для датчика 'CATERR' с "type:0x7" "Table 42-3, Sensor Type Codes" показывает, что этот тип означает "Processor", а значение отдельных битов следующее: 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". Смещение 00h означает "PowerOff/Power Down". Другими словами, если младший значащий бит равен 1, то сервер выключен. Чтобы проверить этот бит, можно использовать функцию `bitand` с маской '1'. Выражение триггер может выглядеть так:

```
bitand(last(/www.example.com/Power Unit Stat,#1),1)=1
```

чтобы предупреждать 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.

5. Простые проверки

Обзор

Простые проверки обычно используются для удалённых безагентных проверок сервисов.

Обратите внимание, что Zabbix агент не требуется для простых проверок. За обработку простых проверок (выполнение внешних подключений и т. д.) отвечает Zabbix сервер/прокси.

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 *Имя пользователя* и *Пароль* (ограничены 255 символами) в настройке 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. Щёлкните по ключу элемента данных, чтобы увидеть полные сведения.

См. также ключи элементов данных 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, запущен ли сервис и принимает ли он TCP-соединения.
net.tcp.service.perf	Проверяет производительность TCP-сервиса.
net.udp.service	Проверяет, запущен ли сервис и отвечает ли он на UDP-запросы.
net.udp.service.perf	Проверяет производительность UDP-сервиса.

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.

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[service,<ip>,<port>]

 Проверяет, запущен ли сервис и принимает ли он TCP-соединения.
 Возвращаемое значение: *0* — сервис недоступен; *1* — сервис запущен.

Параметры:

- **service** — возможные значения: *ssh*, *ldap*, *smtp*, *ftp*, *http*, *pop*, *nntp*, *imap*, *tcp*, *https*, *telnet* (см. подробности);
- **ip** — IP-адрес или DNS-имя (по умолчанию используется IP/DNS узла сети);
- **port** — номер порта (по умолчанию используется стандартный номер порта сервиса).

Комментарии:

- Обратите внимание, что для сервиса *tcp* указание порта обязательно;
- Эти проверки могут приводить к появлению дополнительных сообщений в системных лог-файлах демонов (обычно регистрируются SMTP- и SSH-сеансы);
- 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[tcp,<ip>,port]` для подобных проверок.

Пример:

```
net.tcp.service[ftp,,45] #                                  FTP-      TCP-    45.
```

Attention:
Если SELinux работает в режиме enforced, пользовательские простые TCP/UDP-проверки могут блокироваться политикой. Чтобы проверить и разрешить новое исходящее соединение, просмотрите отказы аудита: с помощью `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 в `net.tcp.service[ldap]`) требуется OpenLDAP версии 2.6.1 или новее.

Обработка тайм-аута

Гибкие тайм-ауты элементов данных, хотя и поддерживаются для простых проверок, не применяются к `icmpping*` и элементам данных VMware. См. Поддержка гибких тайм-аутов.

ICMP-пинги

Zabbix использует внешнюю утилиту **fping** для обработки ICMP-пингов (**icmpping**, **icmppingloss**, **icmppingretry**, **icmppingsec**).

Установка

fping не входит в состав Zabbix и должен быть установлен отдельно:

- На различных Unix-подобных платформах пакет fping доступен в стандартных репозиториях, но не предустановлен. В этом случае для установки fping можно использовать менеджер пакетов.
- Zabbix предоставляет пакеты fping для RHEL и производных от него дистрибутивов. Обратите внимание, что эти пакеты предоставляются без официальной поддержки.
- fping также можно скомпилировать из исходного кода.

Настройка

Укажите расположение fping в параметре *FpingLocation* файла конфигурации сервера/прокси Zabbix (или параметр *Fping6Location* для использования адресов IPv6).

fping должен быть доступен для выполнения пользователем, от имени которого запущен сервер/прокси Zabbix, и у этого пользователя должны быть достаточные права.

Смотрите также: Известные проблемы для обработки простых проверок с версиями fping ниже 3.10.

Значения по умолчанию

Значения по умолчанию, ограничения и описание значений для параметров ICMP-проверки:

Параметр	Единица	Описание	Флаг Fping	Значения по умолчанию задаются		Допустимые ограничения в Zabbix	
				fping	**Zabbix**	**мин**	**макс**
packets	number	Количество пакетов запроса, отправляемых целевому узлу	-C		3	1	10000
interval	milliseconds	Время ожидания между последовательными пакетами для отдельного целевого узла	-p	1000		20	не ограничено
size	bytes	Размер пакета в байтах 56 байт на x86, 68 байт на x86_64	-b	56 или 68		24	65507
timeout	milliseconds	**fping v3.x** — тайм-аут ожидания после отправки последнего пакета, зависит от флага *-C* **fping v4.x** — индивидуальный тайм-аут для каждого пакета	-t	**fping v3.x** — 500 **fping v4.x** и новее — наследуется от флага *-p*, но не более 2000		50	не ограничено
retries	number	Количество попыток пинга целевого узла, не включая первую попытку	-r	3	1	0	не ограничено
backoff factor	number	Число, на которое умножается время ожидания при каждом последующем запросе	-B	1.5	1.0	1.0	5.0

Значения по умолчанию могут немного отличаться в зависимости от платформы и версии.

Кроме того, Zabbix использует параметры fping *-i interval ms* (не путайте с параметром элемента данных *interval*, упомянутым в таблице выше, который соответствует параметру fping *-p*) и *-S source IP address* (или *-I* в более старых версиях fping). Эти параметры автоматически определяются путём запуска проверок с различными комбинациями параметров. Zabbix пытается определить минимальное значение в миллисекундах, которое fping позволяет использовать с *-i*, проверяя 3 значения: 0, 1 и 10. Затем первое успешно сработавшее значение используется для последующих ICMP-проверок. Этот процесс выполняется каждым процессом ICMP pinger отдельно.

Автоматически определённые параметры fping сбрасываются каждый час и определяются заново при следующей попытке выполнить ICMP-проверку. Установите DebugLevel>=4, чтобы просматривать подробности этого процесса в файле журнала сервера или прокси.

Zabbix записывает IP-адреса, которые должны быть проверены любым из ключей *icmpping**, во временный файл, который затем передаётся fping. Если элементы данных имеют разные параметры ключа, в один файл записываются только те, у которых параметры ключа совпадают. Все IP-адреса, записанные в один файл, будут проверяться fping параллельно, поэтому процесс ICMP pinger в Zabbix будет затрачивать фиксированное количество времени независимо от числа IP-адресов в файле.

1. Ключи элементов данных для мониторинга VMware

Список ключей элементов данных для мониторинга VMware перенесён в раздел Мониторинг VMware.

6. Мониторинг файлов журналов

Обзор

Zabbix можно использовать для централизованного мониторинга и анализа файлов журналов с поддержкой и без поддержки ротации логов.

Уведомления можно использовать, чтобы предупреждать пользователей, когда файл журнала содержит определённые строки или шаблоны строк.

Для мониторинга файла журнала необходимо иметь:

- запущенный агент Zabbix на узле сети
- настроенный элемент данных мониторинга журнала

Attention:
Ограничение на размер отслеживаемого файла журнала зависит от поддержки больших файлов.

Настройка

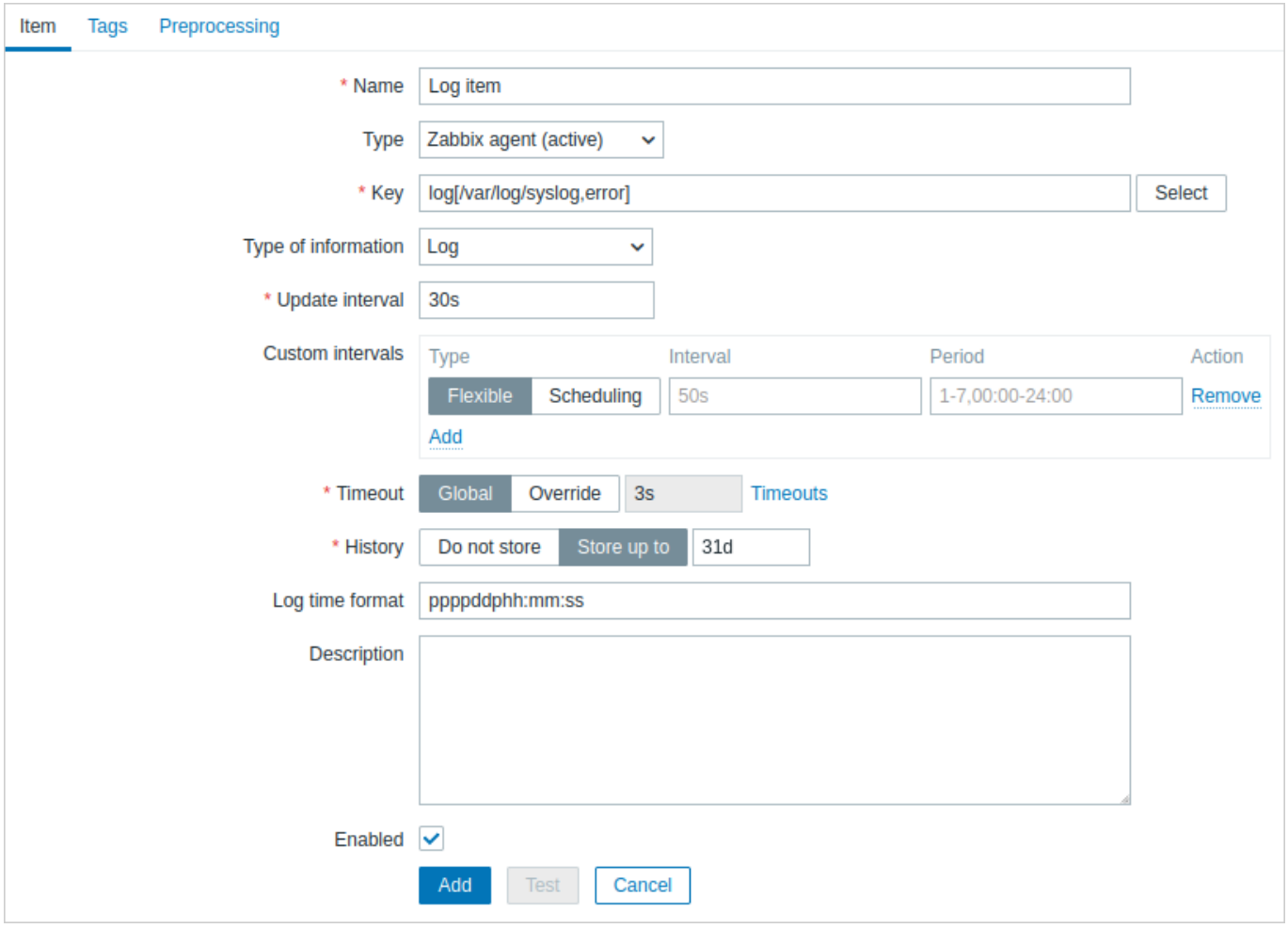
Проверка параметров агента

Убедитесь, что в файле конфигурации агента:

- параметр `Hostname` соответствует имени узла сети в веб-интерфейсе.
- серверы в параметре `ServerActive` указаны для обработки активных проверок.

Настройка элемента данных

Настройте элемент данных для мониторинга журнала.



Все обязательные поля ввода отмечены красной звёздочкой.

Специально для элементов данных наблюдения за журналами вам необходимо указать:

Тип	Здесь выберите **Zabbix агент (активный)**.
Ключ	Используйте один из следующих ключей: **log[]** или **logrt[]:** Эти два ключа элементов данных позволяют выполнять мониторинг файлов журналов и фильтровать записи в журналах по их содержимому при помощи регулярного выражения, если оно присутствует. Например: `log[/var/log/syslog,error]`. Убедитесь, что у файла имеются права на чтение для пользователя «zabbix», в противном случае состояние элемента данных будет «не поддерживается». **log.count[]** или **logrt.count[]**: Эти два ключа элементов данных позволяют получить только количество соответствующих строк. Для получения более подробных сведений касательно использования этих элементов данных и их параметров смотрите раздел поддерживаемых ключей элементов данных Zabbix агентом.
Тип информации	Заполняется автоматически: Для элементов данных log[] или logrt[] — ( ); для элементов данных log.count[] или logrt.count[] — ( ). Если используется опциональный параметр , вы можете вручную выбрать подходящий тип информации, отличный от «Журнал (лог)». Обратите внимание, что выбор не журнального типа информации приведёт к потере локального штампа времени.
Интервал обновления (в сек)	Этот параметр определяет, как часто Zabbix агент будет проверять наличие любых изменений в файле журнала. Указав этот параметр равным 1 секунде, вы можете быть уверенными, что получите новые записи как можно скорее.

Формат времени журнала	В этом поле вы можете опционально задать шаблон для анализа штампа времени строки журнала. Поддерживаемые заменители: * **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 пустым, штамп времени будет выставлен в 0, что в формате Unix time представляет 1 января 1970 года. Например, рассмотрим следующую строку из файла журнала Zabbix агента: « 23480:20100328:154718.045 Zabbix agent started. Zabbix 1.8.2 (revision 11211).» Она начинается с шести символьных позиций для PID, далее следуют дата, время и остальная часть строки. Форматом времени журнала для такой строки могло бы являться «ppppp:yyyyMMdd:hhmmss». Обратите внимание, что символы «p» и «:» являются лишь заменителями и могут быть любыми символами, за исключением «yMdhms».

Важные примечания

- сервер и агент хранят информацию о размере отслеживаемого журнала и времени последнего изменения (для logrt) в двух счетчиках. Дополнительно:
  - 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 принятия 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, позволяющий разумно действовать в неопределенных условиях, когда ротация файлов журналов приводит к появлению нескольких файлов журналов с одинаковым временем последнего изменения.
  - Номера inode, индексы файлов и MD5-суммы собираются агентом Zabbix внутренне. 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 копированием файла обратно под его исходным именем (это создает новый inode). В обоих случаях Zabbix может считать файл другим файлом и перечитать его с начала, что может привести к дублирующимся оповещениям.
  - Если для элемента данных `logrt[]` имеется несколько подходящих файлов журналов и агент Zabbix отслеживает самый новый из них, а этот самый новый файл журнала удаляется, в журнал записывается предупреждение `"there are no files matching "<regexp mask>" in "<directory>"`. агент Zabbix игнорирует файлы журналов со временем изменения меньше самого нового времени изменения, которое агент видел для проверяемого элемента данных `logrt[]`.
- агент начинает чтение файла журнала с того места, где он остановился в предыдущий раз.
- Количество уже проанализированных байт (счетчик размера) и время последнего изменения (счетчик времени) хранятся в базе данных Zabbix и отправляются агенту, чтобы гарантировать, что агент начнет чтение файла журнала с этой точки 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 подходящие файлы, счетчик размера сбрасывается в 0, чтобы при последующем появлении файлов анализ начался с начала.
- Всякий раз, когда файл журнала становится меньше, чем известный агенту счетчик размера журнала, счетчик сбрасывается в ноль, и агент начинает читать файл журнала с начала с учетом счетчика времени.
- Если в каталоге есть несколько подходящих файлов с одинаковым временем последнего изменения, агент пытается корректно проанализировать все файлы журналов с одинаковым временем изменения и избежать пропуска данных или повторного анализа одних и тех же данных, хотя это невозможно гарантировать во всех ситуациях. агент не предполагает какую-либо конкретную схему ротации файлов журналов и не определяет ее. При наличии нескольких файлов журналов с одинаковым временем последнего изменения агент будет

обрабатывать их в лексикографически убывающем порядке. Таким образом, для некоторых схем ротации файлы журналов будут анализироваться и передаваться в их исходном порядке. Для других схем ротации исходный порядок файлов журналов соблюдаться не будет, что может привести к передаче совпавших записей файлов журналов в измененном порядке (эта проблема не возникает, если файлы журналов 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.
- Для поиска требуемой строки Zabbix обрабатывает в 10 раз больше новых строк, чем задано в MaxLinesPerSecond. Таким образом, например, если элемент данных `log[]` или `logrt[]` имеет *Update interval* 1 секунду, по умолчанию агент проанализирует не более 200 записей файла журнала и отправит не более 20 совпавших 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 анализируемых записей файла журнала и 1000 совпавших записей, отправляемых на сервер Zabbix за одну проверку. Если *Update interval* установлен в 2 секунды, пределы для одной проверки будут в 2 раза выше, чем при *Update interval* 1 секунда.
- Кроме того, значения log и log.count всегда ограничены 50% размера буфера отправки агента, даже если в нем нет значений, не относящихся к журналам. Поэтому, чтобы значения **maxlines** были отправлены в одном соединении (а не в нескольких), параметр агента BufferSize должен быть не меньше maxlines x 2. агент Zabbix может выгружать данные во время сбора журналов и тем самым освобождать буфер, тогда как агент Zabbix 2 остановит сбор журналов до тех пор, пока данные не будут выгружены и буфер не освободится; это выполняется асинхронно.
- При отсутствии элементов данных журналов весь размер буфера агента используется для значений, не относящихся к журналам. Когда поступают значения журналов, они при необходимости заменяют более старые значения, не относящиеся к журналам, вплоть до выделенных 50%.
- Для записей файла журнала длиннее 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 после повторного запуска агента длинная запись может быть проанализирована снова и иначе.
- Особое примечание для разделителей пути "\": если file_format имеет значение "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 каталогов по регулярному выражению не поддерживается.
- На платформах UNIX элемент данных `logrt[]` становится NOTSUPPORTED, если каталог, в котором ожидается наличие файлов журналов, не существует.
- В Microsoft Windows, если каталог не существует, элемент данных не станет NOTSUPPORTED (например, если в ключе элемента данных допущена ошибка в имени каталога).
- Отсутствие файлов журналов для элемента данных `logrt[]` не делает его NOTSUPPORTED. Ошибки чтения файлов журналов 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 DebugLevel=4 или DebugLevel=5.
- Поиск вопросительного знака с помощью регулярного выражения, например `\?`, может приводить к ложноположительным результатам, если текстовый файл содержит символы NUL, поскольку они заменяются на "?" в Zabbix для продолжения обработки строки до символа новой строки.

Извлечение совпадающей части регулярного выражения

Иногда может потребоваться извлечь из целевого файла только интересующее значение вместо возврата всей строки при обнаружении совпадения по регулярному выражению.

Элементы данных журнала позволяют извлекать нужные значения из совпавших строк. Это достигается с помощью дополнительного параметра **output** в элементах данных `log` и `logrt`.

Использование параметра 'output' позволяет указать «захватывающую группу» совпадения, которая нас интересует.

Так, например,

```
log[/path/to/the/file,"large result buffer allocation.*Entries: ([0-9]+)",,,,\1]
```

должно позволить вернуть 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, потому что **\1** ссылается на первую и единственную захватывающую группу: **([0-9]+)**.

А благодаря возможности извлекать и возвращать число, это значение можно использовать для определения триггеров.

Использование параметра maxdelay

Параметр `maxdelay` в элементах данных журналов позволяет игнорировать некоторые старые строки из файлов журналов, чтобы анализировать самые новые строки в пределах `maxdelay` секунд.

> **Warning:**
> Указание `maxdelay > 0` может привести к **игнорированию важных записей файлов журналов и пропущенным оповещениям**. Используйте этот параметр осторожно, на свой страх и риск, только при необходимости.

По умолчанию элементы данных для мониторинга журналов отслеживают все новые строки, появляющиеся в файлах журналов. Однако существуют приложения, которые в некоторых ситуациях начинают записывать в свои файлы журналов огромное количество сообщений. 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 будут добросовестно проанализированы, а совпадающие строки будут отправлены на сервер в соответствии с настройками элементов 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 передаётся большое количество потенциально не слишком информативных сообщений, которые занимают место в базе данных. Во-вторых, из-за ограниченного количества строк, анализируемых в секунду, агент может отставать от самых новых записей журнала на часы. Вполне вероятно, что вы предпочтёте быстрее получать информацию о текущей ситуации в файлах журналов, вместо того чтобы часами разбирать старые записи.

Решением обеих проблем является использование параметра `maxdelay`. Если указано `maxdelay > 0`, то во время каждой проверки измеряются количество обработанных байтов, количество оставшихся байтов и время обработки. На основе этих значений агент вычисляет предполагаемую задержку — сколько секунд потребуется, чтобы проанализировать все оставшиеся записи в файле журнала.

Если задержка не превышает `maxdelay`, то агент продолжает анализировать файл журнала как обычно.

Если задержка больше `maxdelay`, то агент **игнорирует часть файла журнала, "перепрыгивая" через неё** к новой предполагаемой позиции, чтобы оставшиеся строки можно было проанализировать в пределах `maxdelay` секунд.

Обратите внимание, что агент даже не считывает игнорируемые строки в буфер, а вычисляет приблизительную позицию, к которой нужно перейти в файле.

Факт пропуска строк файла журнала записывается в файл журнала агента следующим образом:

```
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
```

Число после "to byte" является приблизительным, потому что после "прыжка" агент корректирует позицию в файле до начала строки журнала, которая может находиться дальше в файле или раньше.

В зависимости от того, как скорость роста файла соотносится со скоростью анализа файла журнала, вы можете не увидеть "прыжков" вообще, увидеть редкие или частые "прыжки", большие или маленькие "прыжки", или даже небольшой "прыжок" при каждой проверке. Колебания системной нагрузки и задержка сети также влияют на вычисление задержки и, следовательно, на "перепрыгивание" вперёд для соблюдения параметра `maxdelay`.

Не рекомендуется задавать `maxdelay < update interval` (это может привести к частым небольшим "прыжкам").

Примечания по обработке ротации лог-файлов с `copytruncate`

`logrt` с параметром `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 оригиналом, а другой — копией.

`logrt` с параметром `copytruncate` старается 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[], частые перезапуски агента, не рекомендуются. Агент старается обрабатывать все эти ситуации достаточно корректно, но хорошие результаты не могут быть гарантированы при любых обстоятельствах.

Примечания к постоянным файлам у элементов данных log*[]

Назначение постоянных файлов

При запуске Zabbix агент получает список активных проверок от Zabbix сервера или прокси. Для метрик log*[] он получает размер обработанного 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 размера обработанного журнала, либо повторно проанализировать файл журнала с начала.

Работающий агент поддерживает более широкий набор атрибутов для отслеживания всех контролируемых файлов журналов между проверками. Это состояние в памяти теряется при остановке агента.

Новый необязательный параметр **persistent_dir** задаёт 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 Zabbix агента.

Основной сценарий использования — мониторинг файла журнала, расположенного в зеркальной файловой системе. До определённого момента времени запись в файл журнала выполняется на оба зеркала. Затем зеркала разделяются. На активной копии файл журнала продолжает расти, в него добавляются новые записи. Zabbix агент анализирует его и отправляет на сервер размер обработанных журналов и время изменения. На пассивной копии файл журнала остаётся без изменений и значительно отстаёт от активной копии. Позже операционная система и Zabbix агент перезагружаются с пассивной копии. Размер обработанного журнала и время изменения, которые Zabbix агент получает от сервера, могут быть недействительны для ситуации на пассивной копии. Чтобы продолжить мониторинг файла журнала с того места, на котором агент остановился в момент разделения зеркала файловой системы, агент восстанавливает своё 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 для некоторых элементов данных журнала должны использоваться постоянные файлы 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 разделение зеркала файловой системы происходят одновременно, очень мала, специальная обработка для этого не предусмотрена. После записи в постоянный файл принудительная синхронизация с носителем данных НЕ выполняется (fsync() не вызывается).

Перезапись последними данными выполняется после успешной отправки соответствующей записи файла журнала или метаданных (обработанный размер файла журнала и время изменения) на сервер Zabbix. Это может происходить так же часто, как и 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 persistent-файлы для удаления. Persistent-файл становится устаревшим, если:

1. Соответствующий элемент данных журнала больше не отслеживается.
2. Элемент данных журнала перенастроен с другим расположением **persistent_dir**, отличным от прежнего.

Удаление выполняется с задержкой 24 часа, поскольку файлы журналов в состоянии NOTSUPPORTED не включаются в список активных проверок, но позже они могут перейти в состояние SUPPORTED, и их persistent-файлы будут полезны.

Если агент остановлен до истечения 24 часов, устаревшие файлы не будут удалены, так как Zabbix агент больше не получает от Zabbix сервера информацию об их расположении.

Warning:
Повторная настройка **persistent_dir** элемента данных журнала обратно на старое расположение **persistent_dir** в то время, когда агент остановлен, без удаления пользователем старого persistent-файла, приведет к восстановлению состояния агента из старого persistent-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/test.log,] — это разные элементы 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„,10] из списка активных проверок агента и созданию элемента данных logrt[/home/zabbix/test.log„,20] (некоторые атрибуты сохраняются при изменении в веб-интерфейсе/сервере, но не в агенте).

Имя файла состоит из MD5-суммы ключа элемента данных с добавленной длиной ключа элемента данных для уменьшения вероятности коллизий. Например, состояние элемента данных logrt[/home/zabbix50/test.log„„„„/home/zabbix50/agent_private] будет храниться в постоянном файле c963ade4008054813bbc0a650bb8e09266.

Несколько элементов данных журнала могут использовать одно и то же значение **persistent_dir**.

persistent_dir указывается с учетом особенностей структуры файловой системы, точек монтирования, параметров монтирования и конфигурации зеркалирования хранилища — постоянный файл должен находиться в той же зеркалируемой файловой системе, что и контролируемый файл журнала.

Если каталог **persistent_dir** не может быть создан, не существует или права доступа для агента Zabbix не позволяют создавать/записывать/читать/удалять файлы, элемент данных журнала становится NOTSUPPORTED.

Если права доступа к файлам постоянного хранилища удалены во время работы агента или возникают другие ошибки (например, диск переполнен), то ошибки записываются в файл журнала агента, но элемент данных журнала не становится NOTSUPPORTED.

Нагрузка на I/O

Постоянный файл элемента данных обновляется после успешной отправки каждого пакета данных (содержащего данные элемента данных) на сервер. Например, значение `BufferSize` по умолчанию равно 100. Если элемент данных журнала обнаружил 70 совпадающих записей, то первые 50 записей будут отправлены одним пакетом, постоянный файл будет обновлён, затем оставшиеся 20 записей будут отправлены (возможно, с некоторой задержкой, когда накопится больше данных) во 2-м пакете, и постоянный файл будет обновлён снова.

Действия, если связь между агентом и сервером прерывается

Каждая совпавшая 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%-ной области буфера отправки агента. Элементы буфера регулярно отправляются на сервер (или прокси), после чего слоты буфера снова освобождаются.

Пока в выделенной области журнала буфера отправки агента есть свободные слоты и связь между агентом и сервером (или прокси) отсутствует, результаты мониторинга журналов накапливаются в буфере отправки. Это помогает смягчить последствия кратковременных сбоев связи.

При более длительных сбоях связи все слоты для журналов оказываются заняты, и выполняются следующие действия:

- Проверки элементов данных `log[]` и `logrt[]` останавливаются. Когда связь восстанавливается и в буфере появляются свободные слоты, проверки возобновляются с предыдущей позиции. Ни одна совпавшая строка не теряется, они просто отправляются 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 совпавших строк в файле журнала, но поскольку в буфере нет свободных слотов, проверка останавливается. Когда связь восстанавливается, агент подсчитывает те же 100 совпавших строк, а также 70 новых совпавших строк. Теперь агент отправляет count = 170, как если бы они были найдены в ходе одной проверки.
- Проверки `log.count[]` и `logrt.count[]` с `maxdelay > 0`: если во время проверки не было ”скачка”, то поведение аналогично описанному выше. Если произошел ”скачок” через строки файла журнала, то позиция после ”скачка” сохраняется, а подсчитанный результат отбрасывается. Таким образом, агент старается не отставать от растущего файла журнала даже в случае сбоя 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, то элемент данных переходит в состояние NOTSUPPORTED 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 (на некоторых или на всех записях журнала), то элемент данных журнала остается поддерживаемым, и мониторинг продолжается. Ошибка времени выполнения записывается в файл журнала агента Zabbix (без записи из файла журнала).

Частота журналирования ограничена одной ошибкой времени выполнения на одну проверку, чтобы агент Zabbix мог отслеживать свой собственный файл журнала. Например, если анализируется 10 записей и для 3 записей возникает ошибка времени выполнения regexp, в журнал агента будет добавлена одна запись.

Исключение: если `MaxLinesPerSecond=1` и интервал обновления=1 (за одну проверку допускается анализ только 1 записи), то ошибки времени выполнения regexp не журналируются.

zabbix_agentd записывает ключ элемента данных в случае ошибки времени выполнения, а zabbix_agent2 записывает ID элемента данных, чтобы помочь определить, у какого элемента данных журнала возникают ошибки времени выполнения. В случае ошибок времени выполнения рекомендуется переработать регулярное выражение.

7. Вычисляемые элементы данных

Обзор

Вычисляемый элемент данных позволяет создавать вычисление на основе значений некоторых существующих элементов данных. Например, может потребоваться вычислить почасовое среднее значение какого-либо элемента данных или рассчитать суммарное значение для группы элементов данных. Именно для этого и предназначен вычисляемый элемент данных.

В вычислениях можно использовать как:

- отдельные значения индивидуальных элементов данных
- сложный фильтр для выбора нескольких элементов данных для агрегации (подробности см. в агрегирующих вычислениях)

Вычисляемые элементы данных — это способ создания виртуальных источников данных. Все вычисления выполняются только сервером Zabbix. Значения периодически вычисляются на основе используемого арифметического выражения.

Полученные данные сохраняются в базе данных Zabbix так же, как и для любого другого элемента данных; сохраняются как значения истории, так и трендов, и могут быть построены графики.

Note:
Если результат вычисления является значением с плавающей точкой, он будет усечён до целого числа, если для вычисляемого элемента данных выбран тип информации *Numeric (unsigned)*.
Также, если в кэше нет недавних данных и в функции не определён период запроса, Zabbix по умолчанию будет обращаться к базе данных за историческими значениями вплоть до одной недели в прошлое.

Вычисляемые элементы данных используют тот же синтаксис, что и выражения триггеров. Если вы знакомы с выражениями триггеров, можно рассматривать вычисляемые элементы данных как функции, анализирующие некоторые значения, но без сравнения с порогом.

В вычисляемых элементах данных допускается сравнение со строками. На вычисляемые элементы данных могут ссылаться макросы или другие сущности так же, как и на любой другой тип элемента данных.

Чтобы использовать вычисляемые элементы данных, выберите тип элемента данных **Calculated**.

Настраиваемые поля

Ключ — это уникальный идентификатор элемента данных (для каждого узла сети). Вы можете создать любое имя ключа, используя поддерживаемые символы.

Определение вычисления следует вводить в поле **Формула**. Между формулой и ключом нет связи. Параметры ключа никак не используются в формуле.

Синтаксис простой формулы:

`function(/host/key,<parameter1>,<parameter2>,...)`

где:

Element	Description
`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 опустить (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.
`parameter(s)`	Параметры функции, если требуются. Поддерживаются суффиксы времени и суффиксы размера памяти.

Attention:
Пользовательские макросы в формуле будут развёрнуты, если используются для ссылки на параметр функции, параметр фильтра элемента данных или константу. Пользовательские макросы НЕ будут развёрнуты, если они ссылаются на функцию, имя узла сети, ключ элемента данных, параметр ключа элемента данных или оператор.

Более сложная формула может использовать комбинацию функций, операторов и скобок. Вы можете использовать все функции и операторы, поддерживаемые в выражениях триггеров. Логика и приоритет операторов полностью совпадают.

В отличие от выражений триггеров, Zabbix обрабатывает вычисляемые элементы данных в соответствии с интервалом обновления элемента данных, а не при получении нового значения.

Все элементы данных, на которые ссылаются функции истории в формуле вычисляемого элемента данных, должны существовать и собирать данные. Также, если вы измените ключ элемента данных, на который есть ссылка, вам придётся вручную обновить все формулы, использующие этот ключ.

Вычисляемый элемент данных может стать неподдерживаемым в нескольких случаях:

- элемент(ы) данных, на который есть ссылка,
  - не найден
  - отключён
  - принадлежит отключённому узлу сети
  - не поддерживается (кроме функции nodata() и операторов с неизвестными значениями)
- нет данных для вычисления функции
- деление на ноль
- используется некорректный синтаксис

Примеры использования

Пример 1

Вычисление процентного отношения свободного места на диске для '/'.

Использование функции **last**:

```
100*last(//vfs.fs.size[/,free])/last(//vfs.fs.size[/,total])
```

Zabbix будет брать последние значения полного и свободного объёма диска и вычислять процентное отношение в соответствии с заданной формулой.

Пример 2

Вычисление 10-минутного усреднения количества значений, обрабатываемых Zabbix.

Использование функции **avg**:

```
avg(/Zabbix Server/zabbix[wcache,values],10m)
```

Обратите внимание, что интенсивное использование вычисляемых элементов данных с большими периодами времени может повлиять на производительность Zabbix сервера.

Пример 3

Вычисление общей пропускной способности на eth0.

Сумма двух функций:

```
last(//net.if.in[eth0,bytes])+last(//net.if.out[eth0,bytes])
```

Пример 4

Вычисление процентного отношения входящего трафика.

Более сложное выражение:

```
100*last(//net.if.in[eth0,bytes])/(last(//net.if.in[eth0,bytes])+last(//net.if.out[eth0,bytes]))
```

Смотрите также: Примеры агрегированных вычислений.

1. Агрегированные вычисления

Обзор

Агрегированные вычисления — это тип вычисляемых элементов данных, который позволяет собирать информацию с нескольких элементов данных Zabbix сервером, и затем вычислять совокупность, в зависимости от использованной функции агрегирования.

Агрегированные вычисления не требуют наличия какого-либо агента, запущенного на наблюдаемом узле сети.

Синтаксис

Для получения агрегированных значений 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 требуемых 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 для 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 агрегирования:

```
aggregate_function(function(/host/key,parameter),function(/host2/key2,parameter),...)
```

Обратите внимание, что `function` здесь должна быть функцией history/trend.

Note:
Если результат агрегирования является значением с плавающей точкой, он будет приведён к целому числу, если агрегированный тип информации элемента данных — *Numeric (unsigned)*.

Пользовательские макросы и макросы низкоуровневого обнаружения поддерживаются в:

- параметрах ключа элемента данных
- параметрах функции
- условиях фильтрации (имя группы узлов сети и имя тега)
- константах выражения

Агрегатное вычисление может стать неподдерживаемым, если:

- не найден ни один из указанных элементов данных (это может произойти, если ключ элемента данных неверен, ни один из элементов данных не существует или все включённые группы указаны неверно)
- нет данных для вычисления функции

Примеры использования

Примеры ключей для агрегированных вычислений.

Пример 1

Общий объём дискового пространства группы узлов сети 'MySQL Servers'.

```
sum(last_foreach(/*/vfs.fs.size[/,total]?[group="MySQL Servers"]))
```

Пример 2

Сумма последних значений всех элементов данных, соответствующих net.if.in[*], на узле сети.

```
sum(last_foreach(/host/net.if.in[*]))
```

Пример 3

Средняя загрузка процессора группы узлов сети 'MySQL Servers'.

```
avg(last_foreach(/*/system.cpu.load[,avg1]?[group="MySQL Servers"]))
```

Пример 4

5-минутное среднее количество запросов в секунду для группы узлов сети 'MySQL Servers'.

```
avg(avg_foreach(/*/mysql.qps?[group="MySQL Servers"],5m))
```

Пример 5

Средняя загрузка CPU на всех узлах сети в нескольких группах узлов сети, у которых есть указанные теги.

```
avg(last_foreach(/*/system.cpu.load?[(group="Servers A" or group="Servers B" or group="Servers C") and (ta
```

Пример 6

Вычисление, использующее суммы последних значений элементов данных для всей группы узлов сети.

```
sum(last_foreach(/*/net.if.out[eth0,bytes]?[group="video"])) / sum(last_foreach(/*/nginx_stat.sh[active]?[
```

Пример 7

Общее количество неподдерживаемых элементов данных в группе узлов сети 'Zabbix servers'.

```
sum(last_foreach(/*/zabbix[host,,items_unsupported]?[group="Zabbix servers"]))
```

Пример 8

Сумма последних числовых результатов DNS-проверок по всем узлам сети. Показанная форма элемента данных — net.dns[192.0.2.0,example.com,A], как пример одного из возможных ключей.

```
sum(last_foreach(/*/net.dns[*,*,*]))
```

Обратите внимание, что подстановочные знаки должны соответствовать количеству параметров в ключе (здесь net.dns имеет три параметра: ip, name, type).

Примеры правильного/неправильного синтаксиса

Выражения (включая вызовы функций) не могут использоваться в качестве параметров функций history, trend или foreach. Однако сами эти функции могут использоваться в других (неисторических) параметрах 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 выражении вида:

```
sum(sum_foreach(//resptime[*],5m))/sum(count_foreach(//resptime[*],5m))
```

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 уже другой набор значений.

8. Внутренние проверки

Обзор

Внутренние проверки позволяют отслеживать внутренние процессы Zabbix. Другими словами, вы можете контролировать, что происходит с сервером Zabbix или прокси Zabbix.

Внутренние проверки вычисляются:

- на сервере Zabbix — если узел сети контролируется сервером
- на прокси Zabbix — если узел сети контролируется прокси

Внутренние проверки обрабатываются сервером или прокси независимо от статуса обслуживания узла сети.

Чтобы использовать этот элемент данных, выберите тип элемента данных **Zabbix internal**.

> **Note:**
> Внутренние проверки обрабатываются Zabbix poller.

Производительность

Использование некоторых внутренних элементов данных может отрицательно повлиять на производительность. К таким элементам данных относятся:

- zabbix[host,,items]
- zabbix[host,,items_unsupported]
- zabbix[hosts]
- zabbix[items]
- zabbix[items_unsupported]
- zabbix[queue,,]
- zabbix[requiredperformance]
- zabbix[stats,,,queue,,]
- zabbix[triggers]

Разделы веб-интерфейса Информация о системе и Очередь также подвержены этому влиянию.

Поддерживаемые проверки

Ключи элементов данных перечислены без настраиваемых параметров и дополнительной информации. Нажмите на ключ элемента данных, чтобы увидеть полные сведения.

Ключ элемента данных	Описание
zabbix[boottime]	Время запуска процесса Zabbix сервер или Zabbix прокси в секундах.
zabbix[cluster,discovery,nodes]	Обнаруживает узлы кластера высокой доступности.
zabbix[connector_queue]	Количество значений, поставленных в очередь коннектора.
zabbix[discovery_queue]	Количество сетевых проверок, поставленных в очередь обнаружения.
zabbix[host„items]	Количество включённых элементов данных (поддерживаемых и неподдерживаемых) на узле сети.
zabbix[host„items_unsupported]	Количество включённых неподдерживаемых элементов данных на узле сети.
zabbix[host„maintenance]	Текущий статус обслуживания узла сети.
zabbix[host,active_agent,available]	Доступность активных проверок агента на узле сети.
zabbix[host,discovery,interfaces]	Сведения обо всех настроенных интерфейсах узла сети в веб-интерфейсе Zabbix.
zabbix[host„available]	Доступность основного интерфейса определённого типа проверок на узле сети.
zabbix[hosts]	Количество отслеживаемых узлов сети.
zabbix[items]	Количество включённых элементов данных (поддерживаемых и неподдерживаемых).
zabbix[items_unsupported]	Количество неподдерживаемых элементов данных.
zabbix[java„]	Информация о Zabbix Java gateway.
zabbix[lld_queue]	Количество значений, поставленных в очередь обработки low-level discovery.
zabbix[preprocessing]	Статистика значений, полученных менеджером предобработки.
zabbix[preprocessing_queue]	Количество значений, поставленных в очередь предобработки.
zabbix[process„,]	Процент времени, в течение которого определённый процесс Zabbix или группа процессов (определяемая по <type> и <mode>) находились в состоянии <state>.
zabbix[proxy„]	Информация о Zabbix прокси.
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 реального времени.
zabbix[proxy_buffer,buffer,]	Возвращает статистику использования буфера памяти прокси.
zabbix[proxy_buffer,state,changes]	Возвращает количество изменений состояния между режимами дискового/памятного буфера 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 Zabbix сервер или Zabbix прокси в ожидаемом количестве новых значений в секунду.
zabbix[stats„]	Возвращает внутренние метрики Zabbix сервер или прокси. Если указаны <ip> и <port>, метрики извлекаются из удалённого экземпляра; в противном случае — из локального экземпляра.
zabbix[stats„,queue„]	Возвращает внутренние метрики очереди Zabbix сервер или прокси. Если указаны <ip> и <port>, метрики извлекаются из удалённого экземпляра; в противном случае — из локального экземпляра.
zabbix[tcache„]	Статистика эффективности кэша функций трендов Zabbix.
zabbix[triggers]	Количество включённых триггеров в базе данных Zabbix, у которых все элементы данных включены на включённых узлах сети.
zabbix[uptime]	Время непрерывной работы процесса Zabbix сервер или прокси в секундах.
zabbix[vcache,buffer,]	Статистика доступности кэша значений Zabbix.
zabbix[vcache,cache,]	Статистика эффективности кэша значений Zabbix.
zabbix[version]	Версия Zabbix сервер или прокси.

Ключ элемента данных	Описание
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,<type>,available]`).
- Параметры в угловых скобках **< >** должны быть заменены допустимым значением. Если у параметра есть значение по умолчанию, его можно опустить.
- Значения для элементов данных и параметров элементов данных, помеченные как ”not supported on proxy”, могут быть получены только в том случае, если узел сети мониторится сервером. И наоборот, значения, помеченные как ”not supported on server”, могут быть получены только в том случае, если узел сети мониторится прокси.

zabbix[boottime]

 Время запуска процесса Zabbix сервера или Zabbix прокси в секундах.
 Возвращаемое значение: *Целое число*.

zabbix[cluster,discovery,nodes]

 Обнаруживает узлы кластера высокой доступности.
 Возвращаемое значение: *объект JSON*.

Комментарии:

- Этот элемент данных можно использовать в низкоуровневом обнаружении.
- Возвращаемые поля в основном соответствуют свойствам объекта узла высокой доступности.
- Дополнительные поля:
  - `db_timestamp` — текущее время сервера (временная метка Unix), включённое в ответ
  - `lastaccess_age` — количество секунд с момента последнего heartbeat узла (db_timestamp - lastaccess)
  - `status` — состояние узла:
    - 0 — резервный
    - 1 — остановлен вручную
    - 2 — недоступен
    - 3 — активный

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]

 Текущий статус обслуживания узла сети.
 Возвращаемые значения: *0* — нормальное состояние; *1* — обслуживание со сбором данных; *2* — обслуживание без сбора данных.

Комментарии:

- Этот элемент данных всегда обрабатывается сервером Zabbix независимо от расположения узла сети (на сервере или прокси). Прокси не получит этот элемент данных вместе с данными конфигурации.
- Второй параметр должен быть пустым и зарезервирован для будущего использования.

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,<type>,available]

 Доступность основного интерфейса определенного типа проверок на узле сети.
 Возвращаемые значения: *0* — неизвестно; *1* — доступен; *2* — недоступен.

Параметры:

- **type** — *агент*, *snmp*, *ipmi* или *jmx*.

Комментарии:

- Значение элемента данных вычисляется в соответствии с параметрами конфигурации, относящимися к недостижимости/недос узла 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 использования в будущем.

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 предобработки:

- *queued* — количество и размер значений в очереди, требующих предобработки (счётчик)
- *direct* — количество и размер значений в очереди, не требующих предобработки (счётчик)
- *queue* — количество значений, поставленных в очередь предобработки (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, в течение которого определённый процесс Zabbix или группа процессов (определяемая по <type> и <mode>) находились в состоянии <state>. Вычисляется 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*, *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 fork-процессов 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 запущено 4 процесса trapper, значение может быть от 1 до 4);
- **state** — *busy* — процесс находится в занятом состоянии, например, обрабатывает запрос (по умолчанию);
idle — процесс находится в состоянии простоя и ничего не делает.

Комментарии:

- Если <mode> — это номер процесса Zabbix, который не запущен (например, при 5 запущенных poller в <mode> указано значение 6), такой элемент данных станет неподдерживаемым.
- Минимум и максимум относятся к проценту использования одного процесса. То есть, если в группе из 3 poller процент использования по процессам составляет 2, 18 и 66, то min вернёт 2, а max — 66.
- Процессы сообщают о том, что они делают, в разделяемую память, а процесс self-monitoring суммирует эти данные каждую секунду. Изменения состояния (busy/idle) регистрируются в момент изменения — таким образом, процесс, который становится busy, регистрируется как busy и не изменяет и не обновляет состояние, пока не станет idle. Это гарантирует, что даже полностью зависшие процессы будут корректно зарегистрированы как занятые на 100%.
- В настоящее время "busy" означает "не спит", однако в будущем могут быть добавлены дополнительные состояния — ожидание блокировок, выполнение запросов к базе данных и т. д. Обратите внимание, что асинхронные poller считаются занятыми, если они достигли предела, заданного параметром конфигурации MaxConcurrentChecksPer-Poller server/proxy.
- 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 heartbeat-сообщения, полученного от прокси;
delay — как долго собранные значения остаются неотправленными; вычисляется как "задержка прокси" + ("текущее время сервера" - "proxy lastaccess"), где "задержка прокси" — это разница между текущим временем прокси и временной меткой самого старого неотправленного значения на прокси.

Комментарии:

- Этот элемент данных всегда обрабатывается сервером Zabbix независимо от расположения узла сети (на сервере или прокси).
- Функцию `fuzzytime()` можно использовать для проверки доступности прокси.

Пример:

```
zabbix[proxy,"Germany",lastaccess] #the timestamp of the last heartbeat message received from "Germany" pr
```

zabbix[proxy,discovery]

 Список прокси-серверов Zabbix с именем, режимом, шифрованием, сжатием, версией, последним контактом, количеством узлов сети, количеством элементов данных, требуемыми значениями в секунду (vps), статусом версии (текущая/устаревшая/неподдерживаемая), тайм-аутами по типам элементов данных, именем группы прокси (если прокси принадлежит группе), состоянием (неизвестно/офлайн/онлайн).
 Возвращаемое значение: *Объект JSON*.

zabbix[proxy group,<имя>,available]

 Количество онлайн-прокси в группе прокси.
 Возвращаемое значение: *Целое число*

Параметры:

- **имя** — имя группы прокси.

zabbix[proxy group,<имя>,pavailable]

 Процент онлайн-прокси в группе прокси.
 Возвращаемое значение: *Число с плавающей точкой*.

Параметры:

- **имя** — имя группы прокси.

zabbix[proxy group,<имя>,proxies]

 Список прокси-серверов Zabbix в группе прокси с именем, режимом, шифрованием, сжатием, версией, последним контактом, количеством узлов сети, количеством элементов данных, требуемыми значениями в секунду (vps), статусом версии (текущая/устаревшая/неподдерживаемая), тайм-аутами, именем группы прокси, состоянием (неизвестно/офлайн/онлайн).<b
Возвращаемое значение: *Объект JSON*.

Параметры:

- **имя** — имя группы прокси.

zabbix[proxy group,<имя>,state]

 Состояние группы прокси.
 Возвращаемое значение: *0* — неизвестно; *1* — офлайн; *2* — восстановление; *3* — онлайн; *4* — деградация.

Параметры:

- **имя** — имя группы 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, которое требуется. 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 возвращает прокси, не входящие ни в одну группу.
- Если для "failover_delay" или "min_online" указано недопустимое значение, сообщается специальное значение *-1* для его обозначения. Недопустимые значения могут возникать, если для конфигурации используются макросы и эти макросы не могут быть развернуты в допустимое значение.
- Состояние группы прокси сообщается как целое число: *0* - неизвестно; *1* - не в сети; *2* - восстановление; *3* - в сети; *4* - ухудшение.

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,<режим>]

 Статистика использования буфера памяти прокси.
 Возвращаемое значение: *Целое число* (для размера); *Число с плавающей точкой* (для процентов).

Параметры:

- **режим**: *total* — общий размер буфера (можно использовать для проверки, включён 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 отключён» (Proxy memory buffer is disabled), когда буфер памяти отключён;
- 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.

Комментарии:

- ”0” также возвращается, когда буфер памяти отключен.

zabbix[proxy_history]

 Количество значений в таблице истории прокси, ожидающих отправку на сервер.
 Возвращаемое значение: *Целое число*.

Комментарии:

- Этот элемент данных не поддерживается на сервере Zabbix.

zabbix[queue,<from>,<to>]

 Количество отслеживаемых элементов данных в очереди, задержанных как минимум на <from> секунд, но менее чем на <to> секунд.
 Возвращаемое значение: *Целое число*.

Параметры:

- **from** — задержка как минимум на указанное значение (по умолчанию 6 секунд);
- **to** — задержка не более чем на указанное значение (по умолчанию бесконечность).

Комментарии:

- В параметрах поддерживаются суффиксы времени (s,m,h,d,w).

Attention:
Прокси Zabbix не учитывает периоды обслуживания; подробности см. в разделе Вычисление очередей во время обслуживания.

zabbix[rcache,<кэш>,<режим>]

 Статистика доступности кэша конфигурации Zabbix.
 Возвращаемое значение: *Целое число* для размера); *Число с плавающей точкой* (для процентов).

Параметры:

- **cache** — *buffer*;
- **mode** — *total* — полный размер буфера;
free — размер свободного места в буфере;
pfree — процент свободного места в буфере;
used — размер использованного места в буфере;
pused — процент использования буфера.

zabbix[requiredperformance]

 Требуемая производительность сервера Zabbix или прокси Zabbix, выраженная в ожидаемом количестве новых значений в секунду.
 Возвращаемое значение: *Float*.

Комментарии:

- Приблизительно соответствует значению «Требуемая производительность сервера, новых значений в секунду» в разделе *Отчёты* > *Информация о системе*.

zabbix[stats,<ip>,<port>]

 Возвращает внутренние метрики сервера Zabbix или прокси. Если указаны <ip> и <port>, метрики извлекаются с удалённого экземпляра; в противном случае — с локального экземпляра.
 Возвращаемые значения: *объект JSON*.

Параметры:

- **ip** — список IP/DNS/сетевых масок серверов/прокси, которые будут опрашиваться удалённо (по умолчанию 127.0.0.1);
- **port** — порт сервера/прокси, который будет опрашиваться удалённо (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 Удалённый 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 удалённого экземпляра; в противном случае — с локального экземпляра.
 Возвращаемые значения: *JSON object*.

Параметры:

- **ip** — список IP/DNS/сетевых масок серверов/прокси, которые будут удалённо опрашиваться (по умолчанию 127.0.0.1);
- **port** — порт сервера/прокси, который будет удалённо опрашиваться (по умолчанию 10051);
- **from** — задержка не менее чем на (по умолчанию 6 секунд);
- **to** — задержка не более чем на (по умолчанию бесконечность).

Комментарии:

- Запрос статистики будет приниматься только с адресов, перечисленных в параметре 'StatsAllowedIP' server/proxy на целевом экземпляре.
- Этот элемент данных возвращает выбранный набор внутренних метрик. Подробности см. в разделе Удалённый мониторинг статистики Zabbix.

Attention:
Прокси Zabbix не учитывает периоды обслуживания; подробности см. в разделе Расчёт очередей во время обслуживания.

zabbix[tcache,cache,<parameter>]

 Статистика эффективности кэша функций трендов Zabbix.
 Возвращаемые значения: *Integer* (для размера); *Float* (для процентов).

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 данных;
requests - количество кэшированных запросов;
pitems - процент кэшированных элементов данных от суммы кэшированных элементов данных и запросов. Низкий процент, скорее всего, означает, что размер кэша можно уменьшить.

Комментарии:

- Этот элемент данных не поддерживается на Zabbix прокси.

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,<режим>]

 Статистика доступности кэша значений Zabbix.
 Возвращаемое значение: *Целое число* (для размера); *Число с плавающей точкой* (для процентов).

Параметры:

- **режим** — *total* — полный размер буфера;
free — размер свободного места в буфере;
pfree — процент свободного места в буфере;
used — размер использованного места в буфере;
pused — процент использованного места в буфере.

Комментарии:

- Этот элемент данных не поддерживается на Zabbix прокси.

zabbix[vcache,cache,<parameter>]

 Статистика эффективности кэша значений Zabbix.
 Возвращаемое значение: *Integer*. Если для <parameter> установлено значение *mode*, возвращаемое значение будет следующим: *0* — нормальный режим; *1* — режим нехватки 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 нехватки 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 предобработки *Change per second*, чтобы получить 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,<режим>]

 Статистика доступности Zabbix кэша VMware.
 Возвращаемое значение: *Целое число* (для размера); *Число с плавающей точкой* (для процентов).

Параметры:

- **режим** — *total* — полный размер буфера;
free — размер свободного места в буфере;
pfree — процент свободного места в буфере;
used — размер использованного места в буфере;
pused — процент использованного места в буфере.

zabbix[vps,written]

 Общее количество значений истории, записанных в базу данных.
 Возвращаемое значение: *Целое число*.

zabbix[wcache,<cache>,<mode>]

 Статистика и доступность кэша записи Zabbix.
 Возвращаемые значения: *Integer* (для количества/размера); *Float* (для процентов).

Параметры:

- **cache** — *values*, *history*, *index* или *trend*;
- **mode** — (с *values*) *all* (по умолчанию) — общее количество значений, обработанных сервером/прокси Zabbix, кроме неподдерживаемых элементов данных (счётчик);
float — количество обработанных значений с плавающей точкой (счётчик);
uint — количество обработанных значений беззнакового целого числа (счётчик);
str — количество обработанных символьных/строковых значений (счётчик);
log — количество обработанных значений журнала (счётчик);
text — количество обработанных текстовых значений (счётчик);
bin — количество обработанных двоичных значений (счётчик);
json — количество обработанных значений JSON (счётчик);
not supported — количество случаев, когда обработка элемента данных приводила к тому, что элемент данных становился неподдерживаемым или оставался в этом состоянии (счётчик);
(с кэшем *history*, *index*, *trend*) *pfree* (по умолчанию) — процент свободного буфера;
total — общий размер буфера;
free — размер свободного буфера;
used — размер используемого буфера;
pused — процент используемого буфера.

Комментарии:

- Параметр кэша `trend` не поддерживается в прокси Zabbix.
- Кэш history используется для хранения значений элементов данных. Низкое значение указывает на проблемы с производительностью на стороне базы данных.
- Кэш индекса history используется для индексации значений, хранящихся в кэше history.

- После заполнения и последующей очистки кэша history кэш индекса history всё равно будет хранить некоторые данные. Такое поведение является ожидаемым и помогает системе работать эффективнее, избегая дополнительной обработки, необходимой для постоянного изменения размера памяти.
- Кэш trend хранит агрегированные данные за текущий час для всех элементов данных, получающих данные.
- Вы можете использовать ключ zabbix[wcache,values] с шагом предобработки *Change per second*, чтобы получить статистику значений в секунду.

9 Проверки через SSH

Обзор

SSH-проверки выполняются как мониторинг без использования агента. Для SSH-проверок Zabbix агент не требуется.

Для выполнения SSH-проверок Zabbix сервер должен быть предварительно 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 с парольной фразой

Проверки SSH предоставляют два метода аутентификации — пару имя пользователя/пароль и аутентификацию на основе файла ключа.

Если вы не собираетесь использовать ключи, никакая дополнительная настройка не требуется, кроме подключения libssh или libssh2 к Zabbix, если вы собираете его из исходного кода.

Аутентификация с помощью файлов ключей

Чтобы использовать аутентификацию на основе ключей для SSH-элементов данных, необходимо внести определённые изменения в конфигурацию сервера.

Откройте файл конфигурации сервера Zabbix (*zabbix_server.conf*) от имени `root` и найдите следующую строку:

```
##### SSHKeyLocation=
```

Раскомментируйте её и укажите полный путь к каталогу, где будут находиться открытый и закрытый ключи:

```
SSHKeyLocation=/home/zabbix/.ssh
```

Сохраните файл и после этого перезапустите сервер Zabbix.

Путь */home/zabbix* здесь является домашним каталогом учётной записи *zabbix*, а *.ssh* — это каталог, в котором по умолчанию будут сгенерированы открытый и закрытый ключи командой ssh-keygen внутри домашнего каталога.

Обычно установочные пакеты сервера Zabbix из различных дистрибутивов ОС создают учётную запись *zabbix* с домашним каталогом в другом месте, например, */var/lib/zabbix* (как для системных учётных записей).

Перед генерацией ключей вы можете перенести домашний каталог в */home/zabbix*, чтобы он соответствовал параметру конфигурации сервера Zabbix `SSHKeyLocation`, упомянутому выше.

> **Note:**
> Следующие шаги можно пропустить, если учётная запись *zabbix* была добавлена вручную в соответствии с разделом установки. В таком случае домашний каталог для учётной записи *zabbix*, скорее всего, уже имеет путь */home/zabbix*.

Чтобы изменить домашний каталог учётной 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 его переноса (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. Безопасный способ сделать это:

```
test -d /home/zabbix || mkdir /home/zabbix
```

Чтобы убедиться, что всё защищено, можно дополнительно выполнить команды для установки прав доступа к домашнему каталогу:

```
chown zabbix:zabbix /home/zabbix
chmod 700 /home/zabbix
```

Ранее остановленные процессы теперь 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 лучшей читаемости приглашения командной строки закомментированы):

```
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):
<        >
##### Enter same passphrase again:
<        >
##### 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 сервера Zabbix `SSHKeyLocation`.

Attention:
Типы ключей, отличные от "rsa", могут поддерживаться утилитой ssh-keygen и SSH-серверами, но могут не поддерживаться библиотекой libssh2, используемой Zabbix.

Форма настройки shell

Этот шаг следует выполнить только один раз для каждого узла сети, который будет контролироваться с помощью SSH-проверок.

С помощью следующих команд файл **публичного** ключа можно установить на удалённый узел сети *10.10.10.10*, чтобы SSH-проверки могли выполняться с учётной записью *root* (для лучшей читаемости приглашения командной строки закомментированы):

```
sudo -u zabbix ssh-copy-id root@10.10.10.10
#####               '10.10.10.10 (10.10.10.10)'                .
#####           RSA: 38:ba:f2:a4:b5:d9:8f:52:00:09:f7:1f:75:cc:0b:46.
#####       ,                      (yes/no)?
yes
#####         : '10.10.10.10' (RSA)                                .
```

```
#####     root@10.10.10.10:
<         root>
#####                          "ssh 'root@10.10.10.10'",
#####       ,                    ,           .
```

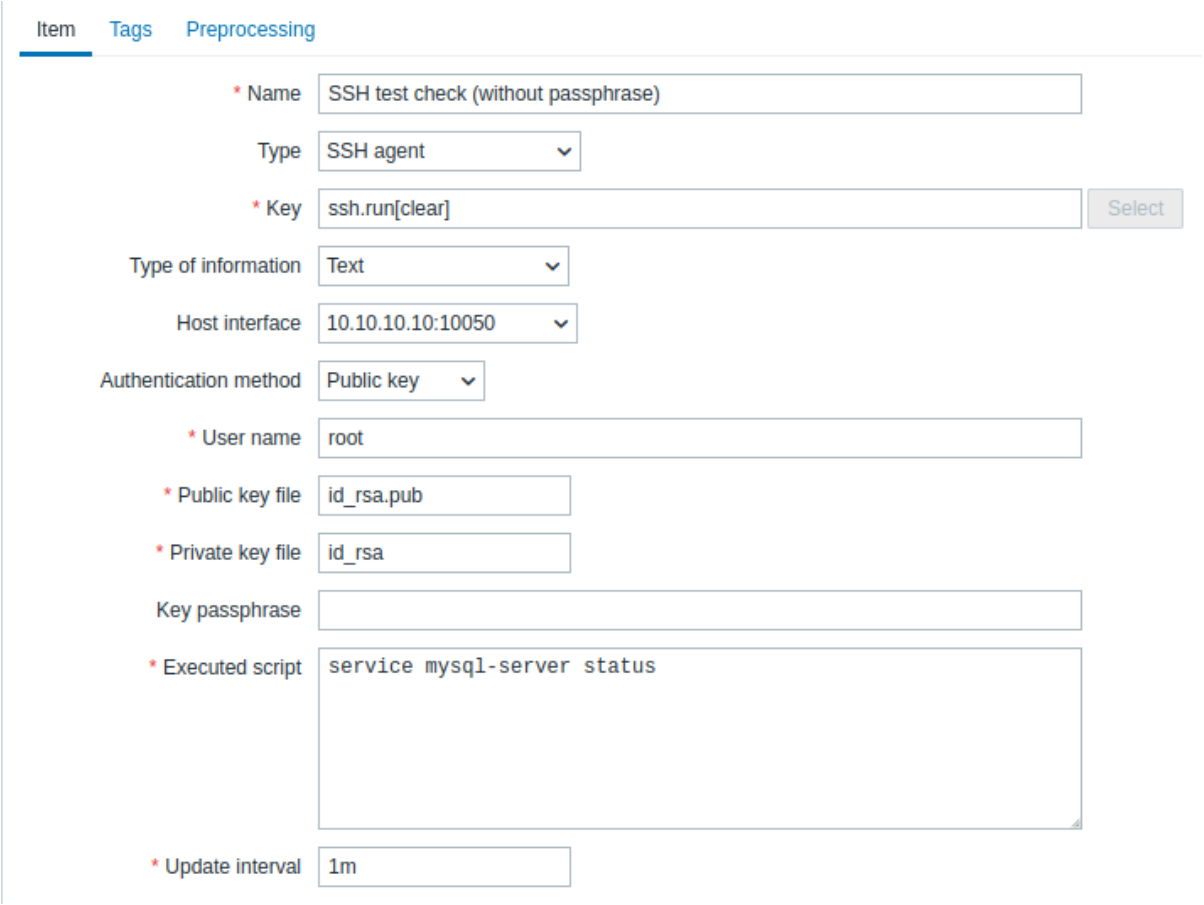
Теперь можно проверить вход по SSH, используя приватный ключ по умолчанию (*/home/zabbix/.ssh/id_rsa*) для учётной записи пользователя *zabbix*:

```
sudo -u zabbix ssh root@10.10.10.10
```

Если вход выполнен успешно, значит часть настройки в shell завершена, и удалённый SSH-сеанс можно закрыть.

Настройка элемента данных

Фактические команды для выполнения необходимо указать в поле *Выполняемый скрипт* в настройках элемента данных. Несколько команд можно выполнять последовательно, размещая их каждую с новой строки. В этом случае возвращаемые значения также будут отформатированы как многострочные.



Все обязательные поля ввода отмечены красной звёздочкой.

Поля, требующие специальной информации для SSH-элементов данных:

Параметр	Описание	Комментарии
Type	Выберите здесь **SSH agent**.	

Параметр	Описание	Комментарии

Key	Уникальный (в пределах узла сети) ключ элемента данных в формате **ssh.run[unique short description,<ip>,<port>,<encoding>,<ssh options>,<subsystem>]**	**unique short description** является обязательным и должен быть уникальным для каждого SSH-элемента данных на узле сети.
		
		Порт по умолчанию — 22, а не порт, указанный в интерфейсе, которому назначен этот элемент данных.
		
		ssh options позволяют передавать дополнительные параметры SSH в формате *key1=value1;key2=value2,value3*. Несколько значений для одного ключа можно передавать через запятую (в этом случае параметр должен быть заключён в кавычки); несколько ключей параметров можно передавать через точку с запятой.
		
		Поддерживаются следующие ключи параметров: `KexAlgorithms`, `HostkeyAlgorithms`, `Ciphers`, `MACs`, `PubkeyAcceptedKeyTypes`. Поддержка ключей и значений параметров зависит от SSH-библиотеки (например, `PubkeyAcceptedKeyTypes` поддерживается только с libssh); если параметр не поддерживается, будет возвращена ошибка, и элемент данных станет неподдерживаемым.
		
		Обратите внимание, что знак "+" для добавления настроек шифров и "!" для отключения определённых настроек шифров (как в GnuTLS и OpenSSL) не поддерживаются.
		
		Примеры:
		=>
		`ssh.run[KexAlgorithms,127.0.0.1,,,Ciphers=aes128`
		=>
		`ssh.run[KexAlgorithms,,,,"KexAlgorithms=diffie-he`
		=>
		`ssh.run[PubkeyAcceptedKeyTypes,127.0.0.1,,,Pubkey`
		
		subsystem позволяет передавать подсистему SSH, ограничивая SSH-соединение определёнными операциями, разрешёнными этой подсистемой (например, передачей файлов через SFTP или управлением сетевыми устройствами через NETCONF). Обратите внимание, что использование подсистемы также может требовать применения специального синтаксиса скрипта в параметре *Executed script*.
		
		Примеры:
		=>
		`ssh.run[SFTPBackup,192.0.2.18,,,,sftp]`
		=>
		`ssh.run[Cisco1234,192.0.2.18,,,,netconf]`

Authentication method	Один из вариантов: "Password" или "Public key".	
User name	Имя пользователя (до 255 символов) для аутентификации на удалённом узле сети. Обязательно.	

Параметр	Описание	Комментарии
Public key file	Имя файла открытого ключа, если *Authentication method* — ”Public key”. Обязательно.	Пример: *id_rsa.pub* — имя файла открытого ключа по умолчанию, создаваемого командой ssh-keygen.
Private key file	Имя файла закрытого ключа, если *Authentication method* — ”Public key”. Обязательно.	Пример: *id_rsa* — имя файла закрытого ключа по умолчанию.
Password or *Key passphrase*	Пароль (до 255 символов) для аутентификации или Парольная фраза, **если** она использовалась для закрытого ключа.	Оставьте поле *Key passphrase* пустым, если парольная фраза не использовалась. См. также известные проблемы, связанные с использованием парольной фразы.

Executed script	Выполняемые команды оболочки с использованием удалённой SSH-сессии.	Возвращаемое значение выполняемых команд оболочки ограничено 16 МБ (включая конечные пробельные символы, которые усекаются); также применяются ограничения базы данных.
		
		Обратите внимание, что библиотека libssh2 может усекать 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>`
		`]]>]]>`

10 Проверки через Telnet

Обзор

Telnet проверки выполняются без наличия какого-либо агента. Zabbix агент не требуется для проверок через Telnet.

Настраиваемые поля

Фактическую команду(ы), которые должны быть выполнены, необходимо указать в поле **Executed script** в конфигурации элемента данных.
Несколько команд можно выполнять последовательно, размещая каждую на новой строке. В этом случае возвращаемое значение также будет отформатировано как многострочное.

Поддерживаемые символы, которыми может заканчиваться приглашение оболочки:

- $
- #
- >
- %

Note:
Строка приглашения telnet, оканчивающаяся одним из этих символов, будет удалена из возвращаемого значения, но только для первой команды в списке команд, то есть только в начале telnet-сеанса.

Ключ	Описание
telnet.run[<unique short description>,<ip>,<port>,<encoding>]	Выполнить команду на удалённом устройстве, используя telnet-соединение

Attention:
Если telnet-проверка возвращает значение с не-ASCII символами и в кодировке, отличной от UTF8, тогда параметр *<encoding>* ключа должен быть указан корректно. Подробнее см. на странице encoding of returned values.

11 Внешние проверки

Обзор

Внешняя проверка — это проверка, выполняемая сервером Zabbix путем запуска shell-скрипта или бинарного файла. Однако, если узлы сети мониторятся прокси Zabbix, внешние проверки выполняются прокси.

Для внешних проверок не требуется запущенный на мониторируемом узле сети 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 будет искать указанный каталог для внешних скриптов и выполнять команду (см. параметр `ExternalScripts` в файле конфигурации Zabbix server/proxy). Команда будет выполняться от имени того же пользователя, что и сервер Zabbix/прокси, поэтому при необходимости любые права доступа или переменные окружения следует обрабатывать в оберточном скрипте. Права доступа к команде также должны позволять этому пользователю выполнять ее. Для выполнения доступны только команды из указанного каталога.

Warning:
Не злоупотребляйте внешними проверками, так как каждый скрипт требует запуска fork-процесса сервером Zabbix/прокси, 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 на IP-адрес или DNS-имя 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 — это стандартный вывод вместе со стандартным выводом ошибок, сформированными проверкой.

Attention:
Элемент данных, который возвращает текст (символьный, журнал или тип информации text), не станет неподдерживаемым в случае вывода в стандартный поток ошибок.

Возвращаемое значение ограничено 16 МБ (включая завершающие пробельные символы, которые усекаются); также применяются ограничения базы данных.

Если запрошенный скрипт не найден или сервер/прокси Zabbix не имеет 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 процесс, созданный для скрипта, будет завершён.

12 Траппер элементы данных

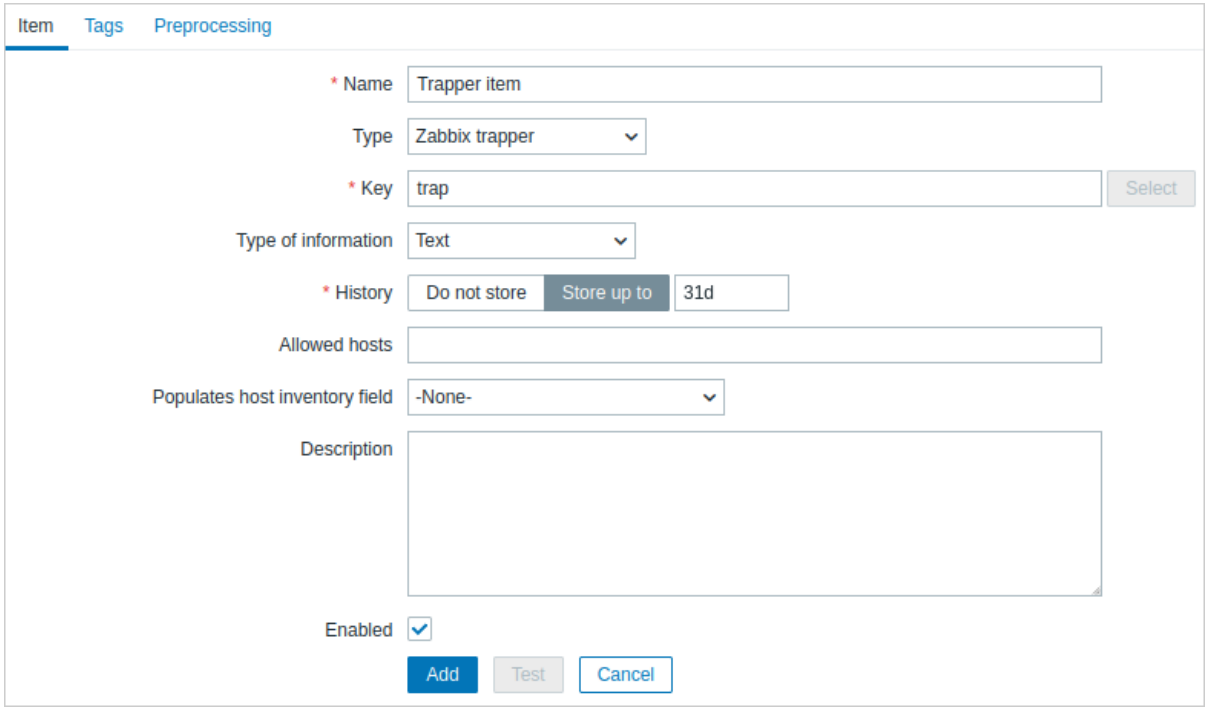
Обзор

Элементы данных типа trapper принимают входящие данные вместо того, чтобы запрашивать их. Это полезно для любых данных, которые вы хотите отправить в Zabbix.

Настройка

Чтобы настроить траппер элемент данных:

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 форму конфигурации.



Все обязательные поля ввода отмечены красной звездочкой.

Поля, требующие конкретной информации для элементов траппера:

Тип	Выберите «Zabbix траппер».
Ключ	Введите ключ, который будет использоваться для распознавания элемента данных при отправке данных на сервер Zabbix.
Тип информации	Выберите тип информации, который будет соответствовать формату отправляемых данных.

Разрешенные хосты	Список IP-адресов, разделенных запятыми (необязательно в нотации CIDR) или DNS-имен. Если указано, входящие соединения будут приниматься только с хостов, перечисленных здесь. Если поддерживается IPv6, то ”127.0.0.1”, ”::127.0.0.1”, ”::ffff:127.0.0.1” обрабатываются одинаково, а ”::/0” разрешает любой адрес IPv4 или IPv6. ”0.0.0.0/0” можно использовать для разрешения любого адреса IPv4. Обратите внимание, что ”IPv4-совместимые адреса IPv6” (префикс 0000::/96) поддерживаются, но не рекомендуются в соответствии с [RFC4291] (https://tools.ietf.org/html/rfc4291#section-2.5.5). 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} Пробелы, пользовательские макросы и поддерживаются макросы хоста {HOST.HOST}, {HOST.NAME}, {HOST.IP}, {HOST.DNS}, {HOST.CONN}.

Note:
Перед отправкой значений, возможно, придется подождать до 60 секунд после сохранения элемента данных, пока сервер Zabbix не применит изменения из обновления кэша конфигурации.

Отправка данных

Отправка данных на Zabbix сервер или прокси возможна с помощью утилиты Zabbix sender или протокола Zabbix sender. Отправка данных на сервер Zabbix также возможна с использованием 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 ”test value”:

```
zabbix_sender -z <server IP address> -p 10051 -s "New host" -k trap -o "test value"
```

Для отправки ”test value” используются следующие параметры команды:

- `-z` — чтобы указать IP-адрес сервера Zabbix
- `-p` — чтобы указать номер порта сервера Zabbix (10051 по умолчанию)
- `-s` — чтобы указать узел сети (убедитесь, что используется техническое, а не видимое имя узла сети)
- `-k` — чтобы указать ключ элемента данных, настроенного в элементе данных типа trapper
- `-o` — чтобы указать отправляемое значение

Attention:
Процесс Zabbix trapper не раскрывает макросы, используемые в ключе элемента данных, при проверке существования соответствующего ключа элемента данных для целевого узла сети.

Для получения дополнительной информации о взаимодействии между Zabbix sender и сервером Zabbix или прокси см. *протокол Zabbix sender*.

history.push

Для отправки данных на сервер Zabbix с помощью метода API `history.push` можно выполнить следующий HTTP POST-запрос, содержащий несколько тестовых значений:

```
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 корректен, 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 на сервере Zabbix. Это может произойти по следующим причинам:

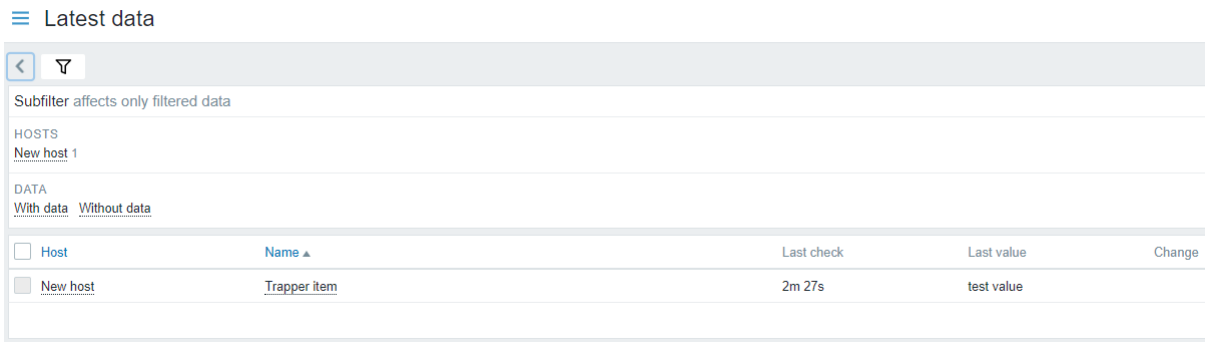
- у пользователя, отправляющего данные, нет прав *read* на узел сети элемента данных;
- узел сети отключён или находится на обслуживании без сбора данных;
- элемент данных не существует или ещё не включён в кэш конфигурации сервера;
- элемент данных отключён или его тип отличается от Zabbix trapper или HTTP agent (с включённым trapping);
- IP-адрес или DNS пользователя не указан в списке *Allowed hosts* элемента данных;
- другой элемент данных уже имеет значение с дублирующейся временной меткой на уровне наносекунд.

Отсутствие ошибок указывает на то, что отправленные значения были приняты в обработку, которая включает предобработку (если она есть), обработку триггеров 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.

Дополнительную информацию о работе с Zabbix API см. в разделе API.

Отображение данных

После отправки данных вы можете перейти к *Мониторинг* → *Последние данные*, чтобы увидеть результат:



Note:
Если отправляется одно числовое значение, в данных graph будут отображаться горизонтальные линии слева и справа от метки времени значения.

13 JMX мониторинг

Обзор

JMX-мониторинг можно использовать для мониторинга JMX-счётчиков Java-приложения.

JMX-мониторинг имеет встроенную поддержку в Zabbix в виде демона Zabbix под названием "Zabbix Java gateway".

Чтобы получить значение определённого JMX-счётчика на узле сети, сервер Zabbix обращается к Zabbix **Java gateway**, который, в свою очередь, использует API управления JMX для удалённого опроса интересующего приложения.

Подробнее и о настройке см. в разделе Zabbix Java gateway.

Warning:
Связь между Java gateway и отслеживаемым JMX-приложением не должна блокироваться межсетевым экраном.

Включение удаленного JMX-мониторинга для Java-приложения

Для Java-приложения не требуется установка какого-либо дополнительного программного обеспечения, однако его необходимо запускать с указанными ниже параметрами командной строки, чтобы обеспечить поддержку удаленного JMX-мониторинга.

В минимальном варианте, если вы хотите просто начать с мониторинга простого Java-приложения на локальном узле без применения механизмов безопасности, запустите его с такими 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 /path/to/your/application.jar
```

Это заставит Java прослушивать входящие JMX-подключения на порту 12345 только с локального узла, а также укажет не требовать аутентификацию или SSL.

Если вы хотите разрешить подключения на другом интерфейсе, задайте параметр -Djava.rmi.server.hostname равным IP-адресу этого интерфейса.

Если вы хотите более строго подойти к вопросам безопасности, Java предоставляет множество других параметров. Например, следующий пример запускает приложение с более гибким набором параметров и открывает доступ из более широкой сети, а не только с локального узла.

```
java \
-Djava.rmi.server.hostname=192.168.3.14 \
-Dcom.sun.management.jmxremote \
-Dcom.sun.management.jmxremote.port=12345 \
-Dcom.sun.management.jmxremote.authenticate=true \
-Dcom.sun.management.jmxremote.password.file=/etc/java-6-openjdk/management/jmxremote.password \
-Dcom.sun.management.jmxremote.access.file=/etc/java-6-openjdk/management/jmxremote.access \
-Dcom.sun.management.jmxremote.ssl=true \
-Dcom.sun.management.jmxremote.registry.ssl=true \
-Djavax.net.ssl.keyStore=$YOUR_KEY_STORE \
-Djavax.net.ssl.keyStorePassword=$YOUR_KEY_STORE_PASSWORD \
-Djavax.net.ssl.trustStore=$YOUR_TRUST_STORE \
-Djavax.net.ssl.trustStorePassword=$YOUR_TRUST_STORE_PASSWORD \
-Dcom.sun.management.jmxremote.ssl.need.client.auth=true \
-jar /path/to/your/application.jar
```

Большинство (если не все) этих настроек можно указать в файле `$JRE/lib/management/management.properties` (или там, где этот файл находится в вашей системе).

Обратите внимание, что если вы хотите использовать SSL, необходимо изменить скрипт startup.sh, добавив параметры `-Djavax.net.ssl.*` в Java gateway, чтобы он знал, где искать хранилища ключей и доверенных сертификатов.

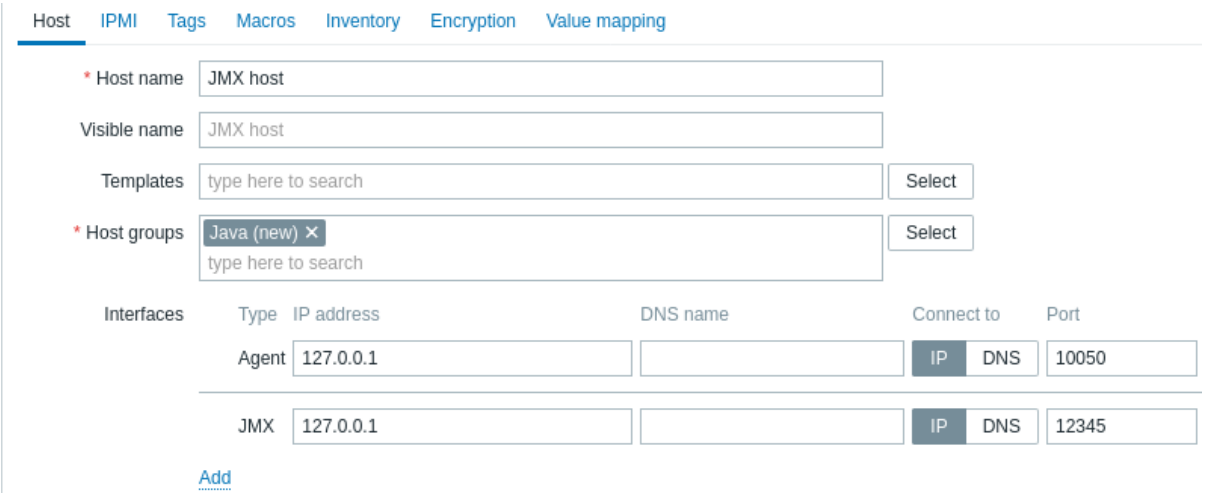
Подробное описание см. в Monitoring and Management Using JMX.

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 удаленного JMX мониторинга, самое время настроить интерфейсы и элементы данных в веб-интерфейсе Zabbix.

Настройка JMX интерфейса

Начнем с создания интерфейса JMX-типа у интересующего узла сети.

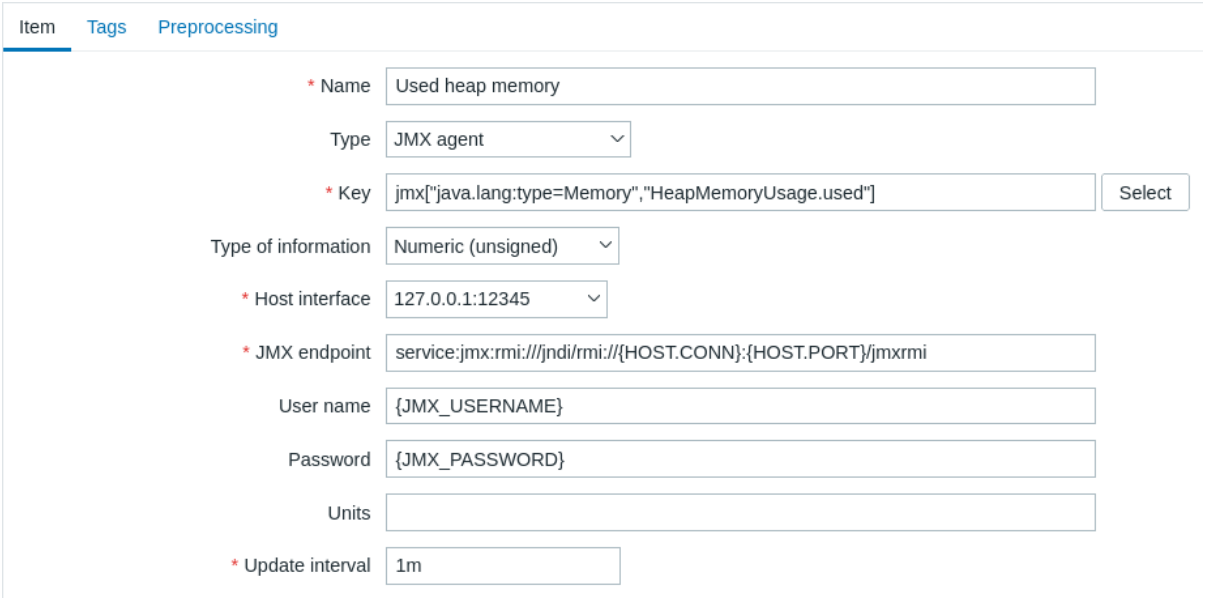


Все обязательные поля ввода отмечены красной звёздочкой.

Добавление элемента данных агента JMX

Для каждого интересующего вас счетчика JMX необходимо добавить элемент данных **агента JMX**, привязанный к этому интерфейсу.

Ключ на снимке экрана ниже: `jmx["java.lang:type=Memory","HeapMemoryUsage.used"]`.



Все обязательные поля ввода отмечены красной звездочкой.

Поля, требующие специальной информации для элементов данных JMX:

Type	Укажите здесь **агент JMX**.
Key	Ключ элемента данных `jmx[]` содержит три параметра: **имя объекта** — имя объекта MBean **имя атрибута** — имя атрибута MBean с необязательными именами полей составных данных, разделенными точками **уникальное краткое описание** — уникальное описание, позволяющее иметь несколько элементов данных JMX с одинаковыми именем объекта и именем атрибута на узле сети (необязательно) Ниже приведены более подробные сведения о ключах элементов данных JMX. Вы можете обнаруживать MBeans 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 JMX-интерфейсу. Этого можно добиться, используя макросы {HOST.*}, как это сделано в JMX endpoint по умолчанию. Поддерживаются {HOST.*} макросы и пользовательские макросы.

User name	Укажите имя пользователя (до 255 символов), если в вашем Java-приложении настроена аутентификация. Поддерживаются пользовательские макросы.
Password	Укажите пароль (до 255 символов), если в вашем Java-приложении настроена аутентификация. Поддерживаются пользовательские макросы.

Если вы хотите отслеживать логический счетчик, который может принимать значения ”true” или ”false”, укажите тип информации ”Numeric (unsigned)” и выберите шаг предобработки ”Boolean to decimal” на вкладке Preprocessing. Сервер будет сохранять логические значения как 1 или 0 соответственно.

Детальная информация о ключах JMX элементов данных

Простые атрибуты

Имя объекта MBean — это не что иное, как строка, которую вы определяете в своем Java-приложении. Имя атрибута, с другой стороны, может быть более сложным. Если атрибут возвращает примитивный тип данных (целое число, строку и т. д.), беспокоиться не о чем — ключ будет выглядеть следующим образом:

```
jmx[com.example:Type=Hello,weight]
```

В этом примере имя объекта — ”com.example:Type=Hello”, имя атрибута — ”weight”, а тип возвращаемого значения, вероятно, должен быть ”Числовой (float)”.

Атрибуты, возвращающие составные данные

Ключ становится все 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 данных, 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, тогда значение этого элемента данных вернет всю структуру атрибута в JSON формате. Значения отдельных элементов из табличных данных можно извлечь с использованием предварительной обработки.

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 всё хорошо. Но что, если имя атрибута или ключ хэша содержит символ точки? Вот пример:

```
jmx[com.example:Type=Hello,all.fruits.apple.weight]
```

Это проблема. Как сообщить Zabbix, что имя атрибута — ”all.fruits”, а не просто ”all”? Как отличить точку, которая является частью имени, от точки, которая разделяет имя атрибута и ключи хэша?

Это возможно, всё, что нужно сделать, — экранировать точки, которые являются частью имени, с помощью обратной косой черты:

```
jmx[com.example:Type=Hello,all\.fruits.apple.weight]
```

Точно так же, если ваш ключ хэша содержит точку, её нужно экранировать:

```
jmx[com.example:Type=Hello,all\.fruits.apple.total\.weight]
```

Другие проблемы

Символ обратной косой черты в имени атрибута должен быть экранирован:

```
jmx[com.example:type=Hello,c:\\documents]
```

О том, как обрабатывать любые другие специальные символы в ключе элемента данных JMX, см. раздел о формате ключа элемента данных section.

На этом всё. Удачного JMX-мониторинга!

Непримитивные типы данных

Можно работать с пользовательскими MBean, которые возвращают непримитивные типы данных и переопределяют метод **toString()**.

Использование пользовательской конечной точки с JBoss EAP 6.4

Пользовательские конечные точки позволяют работать с различными транспортными протоколами, отличными от RMI по умолчанию.

Чтобы проиллюстрировать эту возможность, попробуем в качестве примера настроить мониторинг JBoss EAP 6.4. Сначала сделаем несколько предположений:

- У вас уже установлен Zabbix Java gateway. Если нет, вы можете сделать это в соответствии с документацией.
- Zabbix сервер и Java gateway установлены с префиксом `/usr/local/`
- JBoss уже установлен в `/opt/jboss-eap-6.4/` и работает в автономном режиме
- Предполагается, что все эти компоненты работают на одном и том же узле сети
- Межсетевой экран и SELinux отключены (или настроены соответствующим образом)

Сделаем несколько простых настроек в zabbix_server.conf:

```
JavaGateway=127.0.0.1
StartJavaPollers=5
```

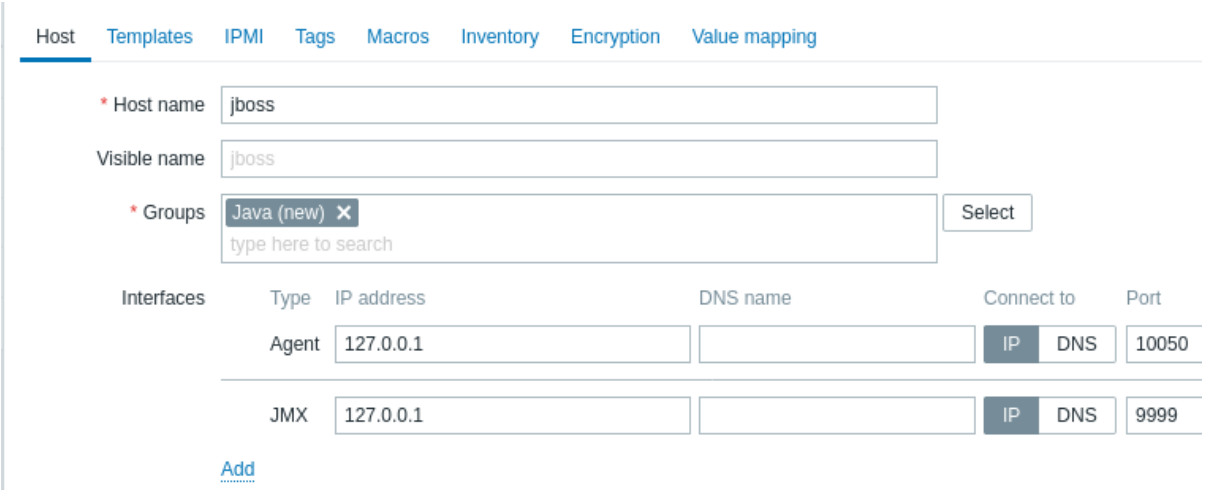
И в конфигурационном файле `zabbix_java/settings.sh` (или `zabbix_java_gateway.conf`):

```
START_POLLERS=5
```

Проверьте, что JBoss прослушивает свой стандартный порт управления:

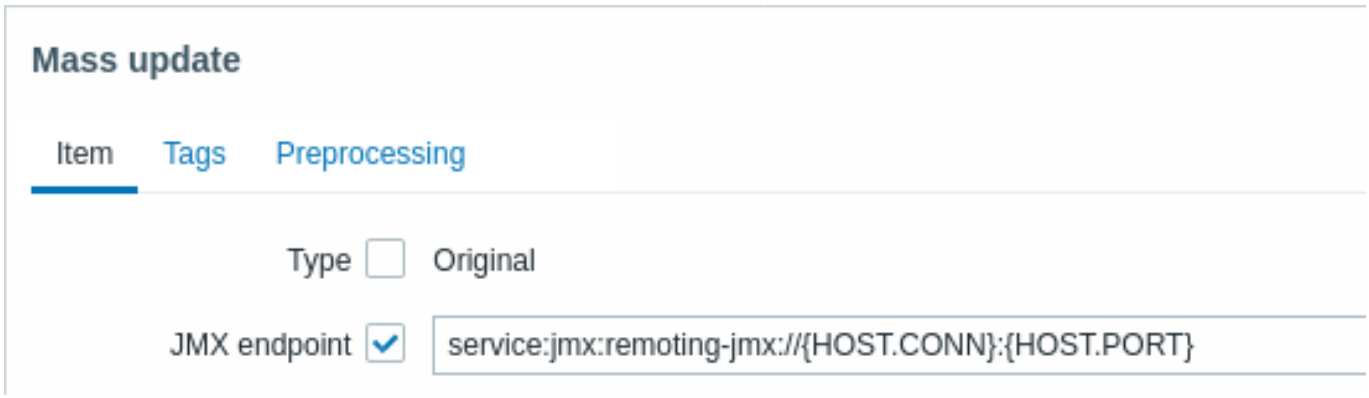
```
$ netstat -natp | grep 9999
tcp        0      0 127.0.0.1:9999          0.0.0.0:*               LISTEN      10148/java
```

Теперь создадим в Zabbix узел сети с интерфейсом JMX 127.0.0.1:9999.



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:

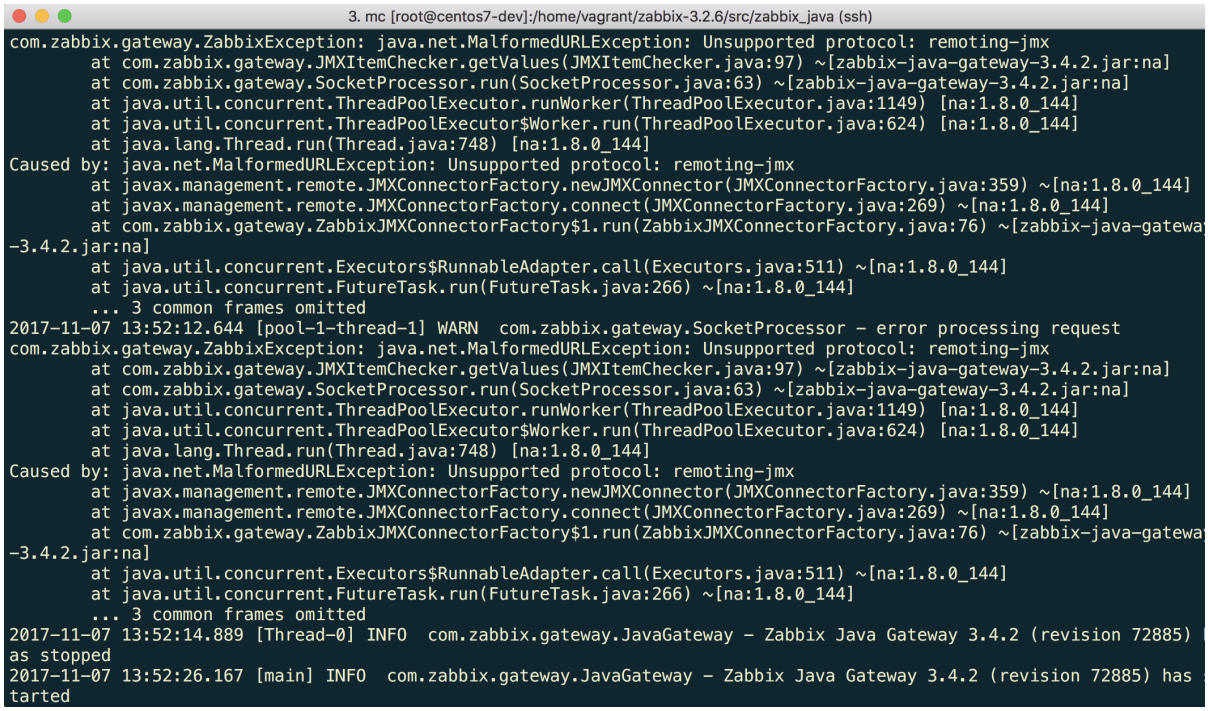
`service:jmx:remoting-jmx://{HOST.CONN}:{HOST.PORT}`



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. Остаётся только перезапустить Java gateway, немного подождать и, если вы всё сделали правильно, убедиться, что данные мониторинга

JMX начинают поступать в Zabbix (см. также: Последние данные).

14 ODBC мониторинг

Обзор

Мониторинг ODBC соответствует типу элемента данных **Database monitor** в веб-интерфейсе Zabbix.

ODBC — это middleware API языка программирования C для доступа к системам управления базами данных (СУБД). Концепция ODBC была разработана Microsoft и позднее перенесена на другие платформы.

Zabbix может выполнять запросы к любой базе данных, которая поддерживается ODBC. Для этого Zabbix не подключается к базам данных напрямую, а использует интерфейс ODBC и драйверы, настроенные в ODBC. Это позволяет более эффективно выполнять мониторинг различных баз данных для множества целей (например, проверять определенные очереди базы данных, статистику использования и т. д.).

Zabbix поддерживает unixODBC, одну из наиболее широко используемых реализаций ODBC API с открытым исходным кодом.

Attention:
См. также: известные проблемы для проверок ODBC.

Установка unixODBC

Рекомендуемый способ установки unixODBC — использовать репозитории пакетов, используемые по умолчанию в операционной системе Linux. В наиболее популярных дистрибутивах Linux unixODBC по умолчанию включён 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 скомпилировать со следующей опцией конфигурации:

 `--with-unixodbc[=ARG]` # ODBC unixODBC.

Установка драйверов unixODBC

Драйвер базы данных unixODBC должен быть установлен для базы данных, которая будет находиться под мониторингом. Список поддерживаемых баз данных и драйверов см. на домашней странице unixODBC: http://www.unixodbc.org/drivers.html.

Note:
В некоторых дистрибутивах Linux драйверы баз данных включены в репозитории пакетов.

MySQL

Чтобы установить драйвер базы данных MySQL для unixODBC, используйте менеджер пакетов 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 для `mysql-connector-odbc` или к документации MariaDB 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] (https://www.postgresql.org/

Oracle

Чтобы установить драйвер базы данных unixODBC, обратитесь к документации Oracle.

MSSQL

Чтобы установить драйвер базы данных MSSQL unixODBC, используйте менеджер пакетов 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.

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 выполнения команды должен содержать информацию, аналогичную 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	Расположение 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).

Параметр	Описание
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
```

Parameter	Description
TEST_MYSQL	Имя источника данных (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.

О других возможных вариантах параметров настройки см. в документации MySQL.

Файл `odbc.ini` для PostgreSQL может содержать дополнительные параметры:

```
[TEST_PSQL]
Description=             PostgreSQL
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=
```

Параметр	Описание
ReadOnly	Указывает, разрешает ли подключение к базе данных только операции чтения (запросы `SELECT`) и запрещает изменения (операторы `INSERT`, `UPDATE` и `DELETE`); полезно в сценариях, где данные должны оставаться неизменными.
Protocol	Версия протокола backend PostgreSQL (игнорируется при использовании SSL-подключений).

Параметр	Описание
ShowOidColumn	Указывает, следует ли включать Object ID (OID) в SQLColumns.
FakeOidIndex	Указывает, следует ли создавать фиктивный уникальный индекс для OID.
RowVersioning	Указывает, следует ли включать возможность для приложений определять, были ли данные изменены другими пользователями в то время, когда вы пытаетесь обновить строку. Обратите внимание, что этот параметр может ускорить процесс обновления, поскольку для обновления строки не требуется указывать каждый отдельный столбец в предложении `WHERE`.
ShowSystemTables	Указывает, должен ли драйвер базы данных обрабатывать системные таблицы как обычные таблицы в SQLTables; полезно для обеспечения доступности, позволяя видеть системные таблицы.
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`.
SSLmode	Указывает режим SSL для подключения.
ConnSettings	Дополнительные настройки, отправляемые backend при подключении.

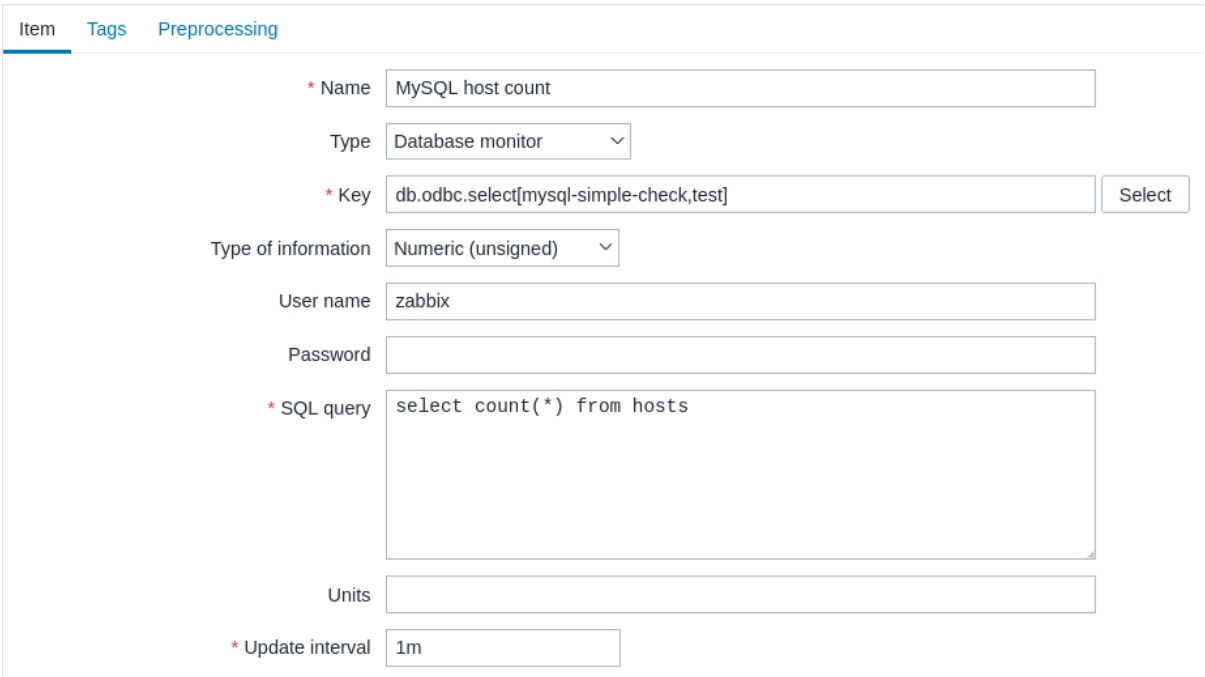
Проверка ODBC-соединения

Чтобы проверить, успешно ли работает ODBC-соединение, вы можете использовать утилиту `isql` (входит в пакет `unixODBC`):

```
isql test
+---------------------------------------+
| Connected!                            |
|                                       |
| sql-statement                         |
| help [tablename]                      |
| quit                                  |
|                                       |
+---------------------------------------+
```

Настройка элемента данных в веб-интерфейсе Zabbix

Настройте **Database monitoring** элемент данных.



Все обязательные поля ввода отмечены красной звездочкой.

Для элементов данных мониторинга баз 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 low-level discovery.
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, то 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, то оно добавляется к строке подключения как `PWD=<password>`. В этом поле поддерживаются специальные символы. Пароль будет добавлен к строке подключения после имени пользователя, например: `UID=<username>;PWD=P?;)*word`. Чтобы проверить получившуюся 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 при использовании `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 мониторинга базы данных станут неподдерживаемыми, если в конфигурации сервера или прокси не запущены процессы *odbc poller*. Чтобы активировать ODBC poller, задайте параметр `StartODBCPollers` в конфигурационном файле Zabbix server или, для проверок, выполняемых прокси, в конфигурационном файле Zabbix proxy.
- Значение параметра *Timeout* в форме настройки элемента данных используется как тайм-аут входа ODBC и тайм-аут выполнения запроса. Обратите внимание, что эти настройки тайм-аута могут игнорироваться, если установленный драйвер ODBC их не поддерживает.
- Команда SQL должна возвращать набор результатов, как и любой запрос, использующий оператор `select`. Синтаксис запроса зависит от СУБД, которая будет его обрабатывать. Синтаксис запроса к хранимой процедуре должен начинаться с ключевого слова `call`.

Подробная информация о ключе элемента данных

Параметры без угловых скобок являются обязательными. Параметры, отмеченные угловыми скобками **< >**, являются необязательными.

db.odbc.select[<уникальное краткое описание>,<dsn>,<строка подключения>]

 Возвращает одно значение, а именно первый столбец первой строки результата SQL-запроса.
 Возвращаемое значение: зависит от SQL-запроса.

Параметры:

- **уникальное краткое описание** — уникальное краткое описание для идентификации элемента 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 драйвера аргументы).

Комментарии:

- Хотя `dsn` и являются необязательными параметрами, требуется как минимум один из них; если определены оба, `dsn` будет проигнорирован.
- Если запрос возвращает более одного столбца, считывается только первый столбец. Если запрос возвращает более одной строки, считывается только первая строка.

db.odbc.get[<unique short description>,<dsn>,<connection string>]

 Преобразует результат SQL-запроса в массив JSON.
 Возвращаемое значение: *объект JSON*.

Параметры:

- **unique short description** — уникальное краткое описание для идентификации элемента данных (для использования в триггерах и т. д.);
- **dsn** — имя источника данных (как указано в `odbc.ini`);
- **connection string** — строка подключения (может содержать специфичные для драйвера аргументы).

Комментарии:

- Хотя `dsn` и `connection string` являются необязательными параметрами, как минимум один из них обязателен; если заданы оба, `dsn` будет проигнорирован.
- Может быть возвращено несколько строк/столбцов в формате JSON. Этот элемент данных может использоваться как основной элемент данных, который собирает все данные одним системным вызовом, в то время как в зависимых элементах данных может использоваться предобработка JSONPath для извлечения отдельных значений. Дополнительную информацию см. в примере возвращаемого формата, используемого в обнаружении низкого уровня.

Пример:

```
#                  MySQL ODBC 5:
db.odbc.get[MySQL example,,"Driver=/usr/local/lib/libmyodbc5a.so;Database=master;Server=127.0.0.1;Port=330
```

db.odbc.discovery[<краткое уникальное описание>,<dsn>,<строка подключения>]

 Преобразует результат SQL-запроса в массив JSON, используемый для обнаружения низкого уровня. Имена столбцов из результата запроса преобразуются в имена макросов обнаружения низкого уровня, сопоставленные с обнаруженными значениями полей. Эти макросы можно использовать при создании прототипов элементов данных, триггеров и т. д.
 Возвращаемое значение: *объект JSON*.

Параметры:

- **краткое уникальное описание** — уникальное краткое описание для идентификации элемента 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 драйвера аргументы).

Комментарии:

- Хотя `dsn` и являются необязательными параметрами, требуется указать как минимум один из них; если определены оба, `dsn` будет проигнорирован.

Сообщения об ошибках

Сообщения об ошибках ODBC структурированы по полям, чтобы предоставить подробную информацию. Например, сообщение об ошибке может выглядеть так:

```
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 возврата ODBC
- "`[42601]`" — SQLState
- "`[7]`" — собственный код ошибки
- "`[ERROR: syntax error at or near ";"; Error while executing the query]`" — собственное 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 объединить их (разделяя символом |) настолько, насколько это позволяет ограничение длины.

15 Зависимые элементы данных

Обзор

Бывают ситуации, когда один элемент данных собирает сразу несколько метрик или даже имеет больше смысла собирать связанные метрики одновременно, например:

- Использование CPU отдельными ядрами
- Входящий/исходящий/общий сетевой трафик

Чтобы обеспечить массовый сбор метрик и их одновременное использование в нескольких связанных элементах данных, Zabbix поддерживает зависимые элементы данных. Зависимые элементы данных зависят от мастер-элемента данных, который собирает их данные одновременно, одним запросом. Новое значение мастер-элемента данных автоматически

заполняет значения зависимых элементов данных. Зависимые элементы данных не могут иметь интервал обновления, отличный от интервала мастер-элемента данных.

Параметры предобработки Zabbix можно использовать для извлечения из данных мастер-элемента данных той части, которая нужна для зависимого элемента данных.

Предобработка управляется процессом `preprocessing manager` вместе с рабочими потоками, которые выполняют шаги предобработки. Все значения с предобработкой, полученные от различных сборщиков данных, проходят через preprocessing manager перед добавлением в кэш истории. Для связи между сборщиками данных (poller, trapper и т. д.) и процессом предобработки используется межпроцессное взаимодействие (IPC) на основе сокетов.

Сервер Zabbix или прокси Zabbix (если узел сети мониторится прокси) выполняют шаги предобработки и обрабатывают зависимые элементы данных.

Элементы данных любого типа, даже зависимые элементы данных, могут быть назначены мастер-элементами данных. Дополнительные уровни зависимых элементов данных могут использоваться для извлечения меньших частей из значения существующего зависимого элемента данных.

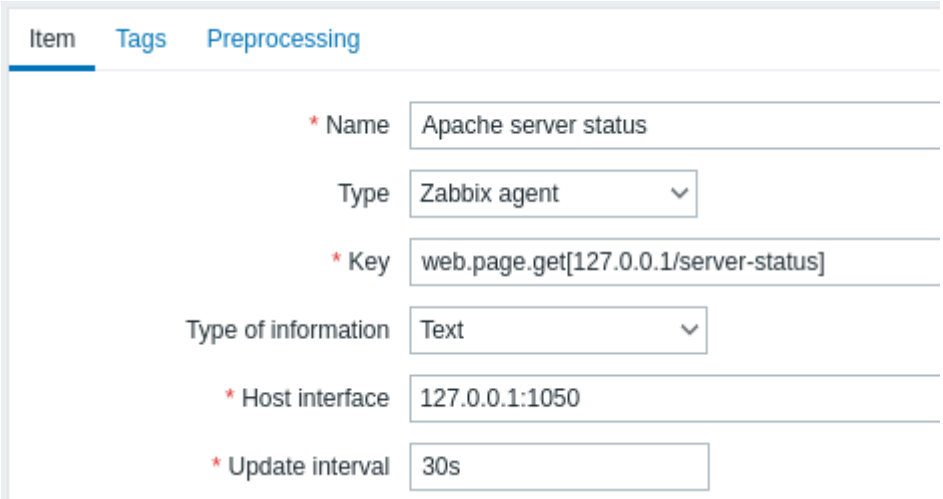
Ограничения

- Разрешены зависимости только в пределах одного и того же узла сети (шаблона)
- Прототип элемента данных может зависеть от другого прототипа элемента данных или обычного элемента данных с того же узла сети
- Зависимый элемент данных на узле сети, у которого основной элемент данных получен из шаблона, не будет экспортирован в XML

Настройка элемента данных

Зависимый элемент данных зависит от своего мастер-элемента данных в части получения данных. Поэтому **мастер-элемент данных** должен быть настроен (или уже существовать) первым:

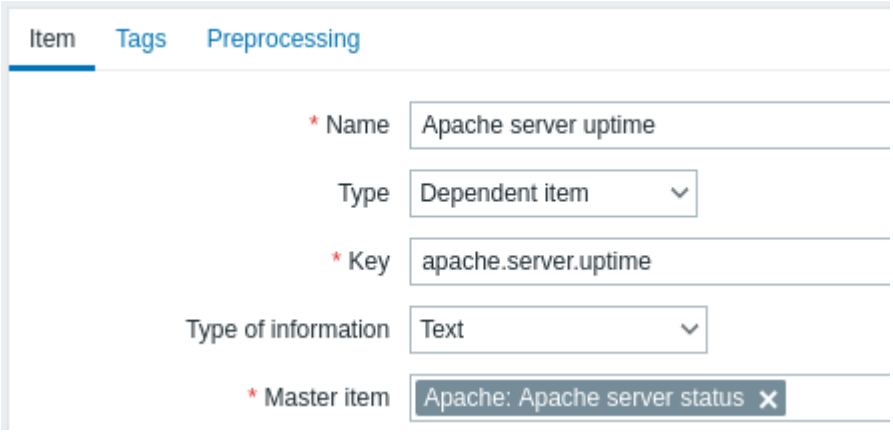
- Перейдите в: *Сбор данных* → *Узлы сети*
- Нажмите *Элементы данных* в строке узла сети
- Нажмите *Создать элемент данных*
- Введите параметры элемента данных в форме



Все обязательные поля ввода отмечены красной звездочкой.

Нажмите *Добавить*, чтобы сохранить мастер-элемент данных.

Затем вы можете настроить **зависимый элемент данных**.

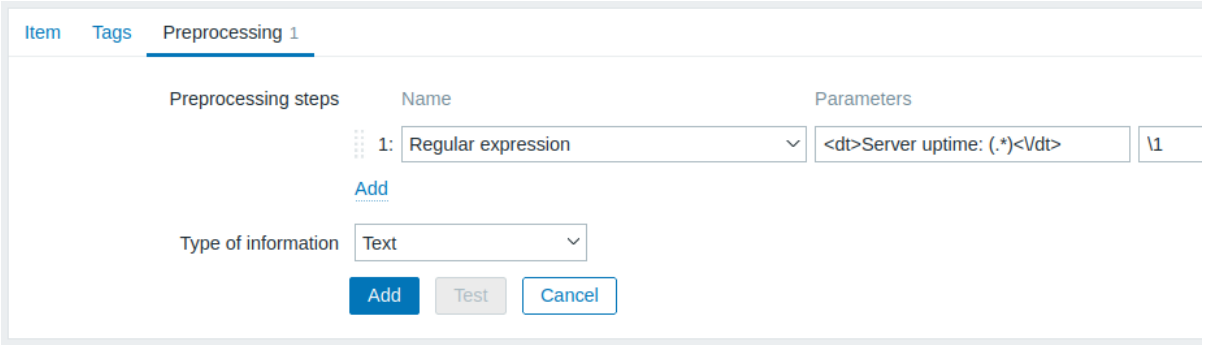


Все обязательные поля ввода отмечены красной звездочкой.

Поля, требующие специальной информации для зависимых элементов данных:

Тип	Выберите здесь **Зависимый элемент данных**.
Ключ	Введите ключ, который будет использоваться для распознавания элемента данных.
Мастер-элемент данных	Выберите мастер-элемент данных. Значение мастер-элемента данных будет использоваться для заполнения значения зависимого элемента данных.
Тип информации	Выберите тип информации, который будет соответствовать формату сохраняемых данных.

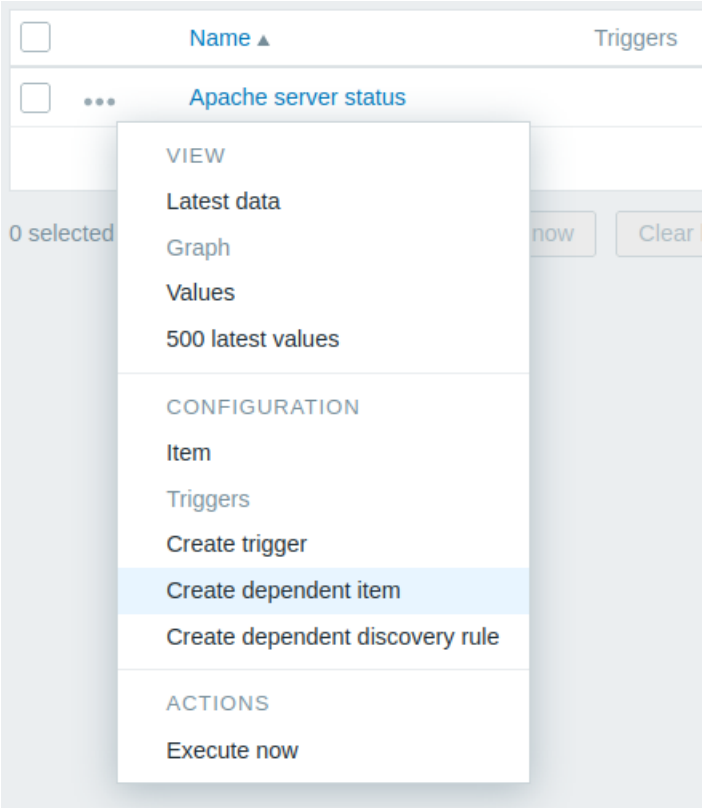
Вы можете использовать предобработку значения элемента данных, чтобы извлечь необходимую часть значения мастер-элемента данных.



Без предобработки значение зависимого элемента данных будет точно таким же, как значение мастер-элемента данных.

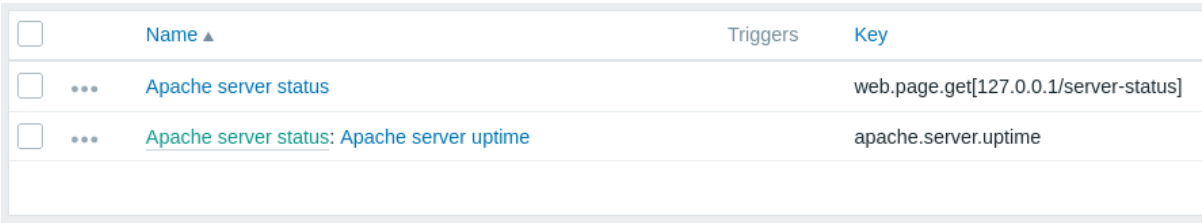
Нажмите *Добавить*, чтобы сохранить зависимый элемент данных.

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.

16 HTTP агент

Обзор

Этот тип элемента данных позволяет опрашивать данные по протоколу HTTP/HTTPS. Также возможен траппинг с использованием утилиты Zabbix sender или протокола Zabbix sender (для отправки данных на сервер или прокси Zabbix), либо с использованием метода API `history.push` (для отправки данных на сервер Zabbix).

Проверки HTTP-элементов данных выполняются сервером Zabbix. Однако, когда узлы сети контролируются прокси Zabbix, проверки HTTP-элементов данных выполняются прокси.

Для проверок HTTP-элементов данных не требуется, чтобы на контролируемом узле сети был запущен какой-либо агент.

HTTP агент поддерживает как HTTP, так и HTTPS. Zabbix при необходимости будет следовать перенаправлениям (см. параметр *Follow redirects* ниже). Максимальное количество перенаправлений жёстко задано и равно 10 (с использованием параметра cURL CURLOPT_MAXREDIRS).

> **Attention:**
> Сервер/прокси Zabbix должен быть изначально настроен с поддержкой cURL (libcurl).

HTTP-проверки выполняются асинхронно — не требуется получать ответ на один запрос до начала других проверок. Разрешение DNS-имен также выполняется асинхронно.

Максимальная параллельность асинхронных проверок составляет 1000 (определяется параметром MaxConcurrentChecksPer-Poller).

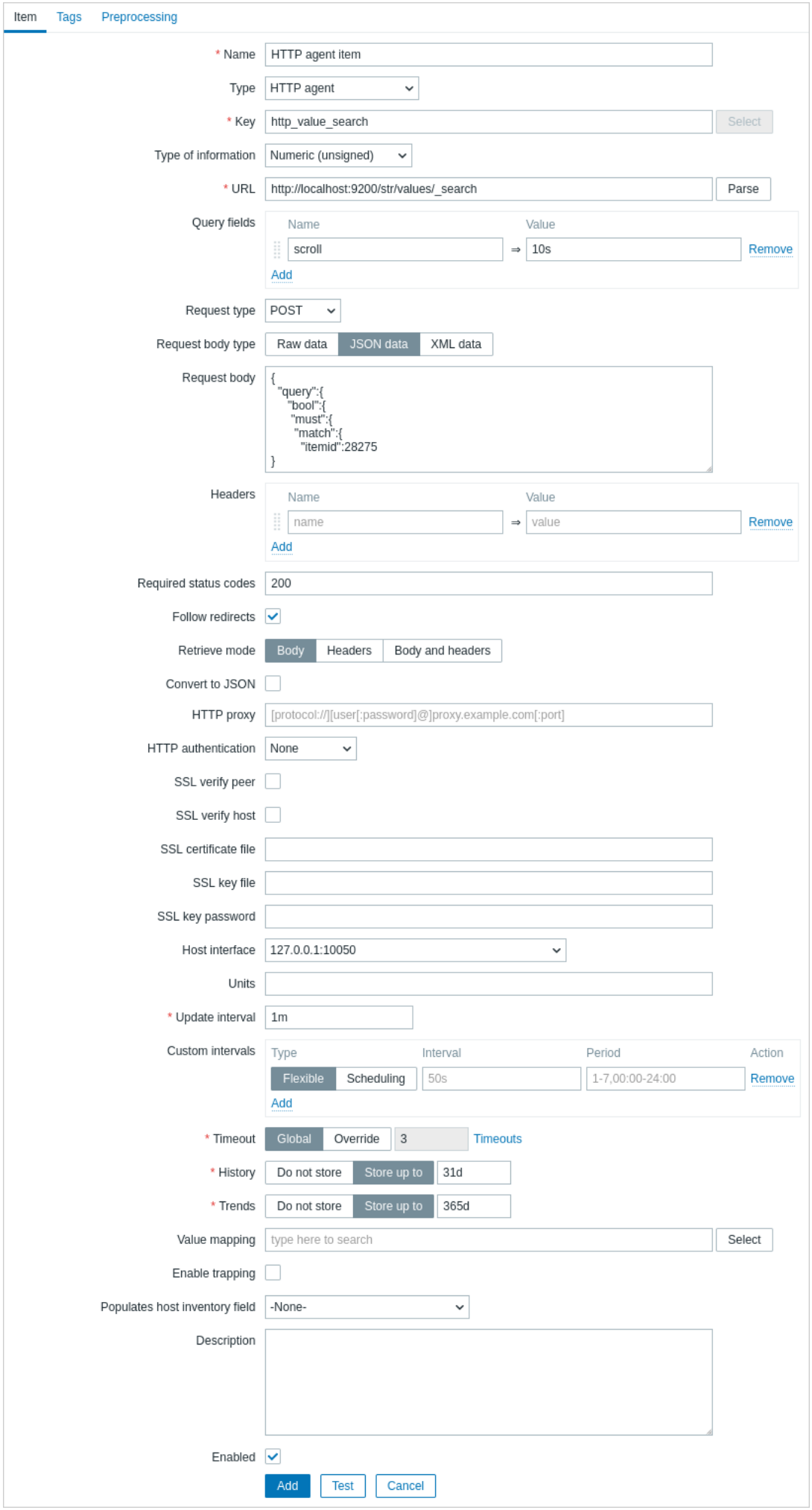
Количество асинхронных poller-процессов HTTP агента задаётся параметром StartHTTPAgentPollers.

Поддержка функции постоянных соединений cURL была добавлена в проверки HTTP агента, начиная с Zabbix 7.0.

Конфигурация

Чтобы настроить HTTP-элемент данных:

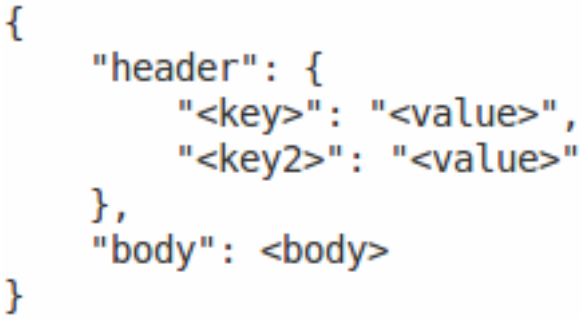
- Перейдите в: *Сбор данных → Узлы сети*
- Нажмите *Элементы данных* в строке узла сети
- Нажмите *Создать элемент данных*
- Введите параметры элемента данных в форме


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
Action
Flexible
Scheduling
50s
1-7,00:00-24:00
Remove
Add
* Timeout
Global
Override
3
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
Populates host inventory field
-None-
Description
Enabled
Add
Test
Cancel

Все обязательные поля ввода отмечены красной звездочкой.

Поля, требующие специальной информации для HTTP-элементов данных:

Параметр	Описание
Тип	Выберите здесь **HTTP агент**.
Ключ	Введите уникальный ключ элемента данных.
URL	URL для подключения и получения данных. Например: https://www.example.com http://www.example.com/download Доменные имена могут быть указаны символами Unicode. При выполнении HTTP-проверки они автоматически преобразуются в punycode и затем в ASCII. Кнопка *Parse* может использоваться для отделения необязательных полей запроса (например, ?name=Admin&password=mypassword) от URL, перемещая атрибуты и значения в *Поля запроса* для автоматического URL-кодирования. 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 низкоуровневого обнаружения. Устанавливает параметр cURL CURLOPT_URL.
Поля запроса	Переменные для URL (см. выше). Указываются как пары атрибут-значение. Значения автоматически URL-кодируются. Значения из макросов сначала разрешаются, а затем автоматически URL-кодируются. Поддерживаемые макросы: {HOST.IP}, {HOST.CONN}, {HOST.DNS}, {HOST.HOST}, {HOST.NAME}, {HOST.PORT}, {ITEM.ID}, {ITEM.KEY}, {ITEM.KEY.ORIG}, пользовательские макросы, макросы низкоуровневого обнаружения. Устанавливает параметр cURL CURLOPT_URL.
Тип запроса	Выберите тип метода запроса: *GET*, *POST*, *PUT* или *HEAD*
Тип тела запроса	Выберите тип тела запроса: **Raw data** - пользовательское тело HTTP-запроса, макросы подставляются, но кодирование не выполняется **JSON data** - тело HTTP-запроса в формате JSON. Макросы могут использоваться как string, number, true и false; макросы, используемые как строки, должны быть заключены в двойные кавычки. Значения из макросов сначала разрешаются, а затем автоматически экранируются. Если "Content-Type" не указан в заголовках, по умолчанию будет использоваться "Content-Type: application/json" **XML data** - тело HTTP-запроса в формате XML. Макросы могут использоваться как текстовый узел, атрибут или секция CDATA. Значения из макросов сначала разрешаются, а затем автоматически экранируются в текстовом узле и атрибуте. Если "Content-Type" не указан в заголовках, по умолчанию будет использоваться "Content-Type: application/xml"
Тело запроса	Введите 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 низкоуровневого обнаружения.
Заголовки	Пользовательские HTTP-заголовки, которые будут отправлены при выполнении запроса. Указываются как пары атрибут-значение. Поддерживаемые макросы: {HOST.IP}, {HOST.CONN}, {HOST.DNS}, {HOST.HOST}, {HOST.NAME}, {HOST.PORT}, {ITEM.ID}, {ITEM.KEY}, {ITEM.KEY.ORIG}, пользовательские макросы, макросы низкоуровневого обнаружения. Устанавливает параметр cURL CURLOPT_HTTPHEADER.
Требуемые коды состояния	Список ожидаемых HTTP-кодов состояния. Если Zabbix получит код, которого нет в списке, элемент данных станет неподдерживаемым. Если поле пустое, проверка не выполняется. Например: 200,201,210-299 Поддерживаемые макросы в списке: пользовательские макросы, макросы низкоуровневого обнаружения. Использует параметр cURL CURLINFO_RESPONSE_CODE.
Следовать перенаправлениям	Установите этот флажок, чтобы следовать HTTP-перенаправлениям. Устанавливает параметр cURL CURLOPT_FOLLOWLOCATION.
Режим получения	Выберите часть ответа, которую необходимо получить: **Тело** - только тело **Заголовки** - только заголовки **Тело и заголовки** - тело и заголовки

Параметр	Описание
Преобразовать в JSON	Заголовки сохраняются как пары атрибут-значение под ключом ”header”, если *Режим получения* не установлен в *Тело*. Если обнаружен ’Content-Type: application/json’, тогда тело сохраняется как объект, в противном случае оно сохраняется как строка, например: 
HTTP прокси	Вы можете указать HTTP-прокси для использования в формате `[protocol://][username[:password]@]proxy.example.com[:port]`. Необязательный префикс `protocol://` может использоваться для указания альтернативных протоколов прокси (например, https, socks4, socks5; см. documentation; поддержка префикса протокола была добавлена в cURL 7.21.7). Если протокол не указан, прокси будет рассматриваться как HTTP-прокси. Если вы укажете неверный протокол, соединение завершится ошибкой, и элемент данных станет неподдерживаемым. По умолчанию будет использоваться порт 1080. Если указано, прокси переопределит связанные с прокси переменные окружения, такие как http_proxy, HTTPS_PROXY. Если не указано, прокси не будет переопределять связанные с прокси переменные окружения. Введенное значение передается ”как есть”, без какой-либо проверки корректности. *Обратите внимание*, что с 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 низкоуровневого обнаружения. Устанавливает 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 в Zabbix); **Digest** - используется аутентификация Digest. Устанавливает параметр cURL CURLOPT_HTTPAUTH.
Имя пользователя	Введите имя пользователя (до 255 символов). Это поле доступно, если для *HTTP аутентификация* установлено значение Basic, NTLM, Kerberos или Digest. Поддерживаются пользовательские макросы и макросы низкоуровневого обнаружения.
Пароль	Введите пароль пользователя (до 255 символов). Это поле доступно, если для *HTTP аутентификация* установлено значение Basic, NTLM, Kerberos или Digest. Поддерживаются пользовательские макросы и макросы низкоуровневого обнаружения.
Проверять SSL-сертификат узла	Установите этот флажок, чтобы проверять SSL-сертификат веб-сервера. Сертификат сервера будет автоматически взят из общесистемного расположения центра сертификации (CA). Вы можете переопределить расположение файлов CA с помощью параметра конфигурации сервера или прокси SSLCALocation. Устанавливает параметр cURL CURLOPT_SSL_VERIFYPEER.
Проверять SSL-имя узла	Установите этот флажок, чтобы проверять совпадение поля Common Name или поля Subject Alternate Name сертификата веб-сервера. Устанавливает 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 файла SSL-ключа пустым. Если ключ зашифрован, укажите пароль в поле пароля SSL-ключа. Каталог, содержащий этот файл, задается параметром конфигурации сервера 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 низкоуровневого обнаружения. 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 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 низкоуровневого обнаружения. Устанавливает параметр cURL CURLOPT_SSLKEY.
Пароль SSL-ключа	Пароль файла закрытого SSL-ключа. Поддерживаемые макросы: пользовательские макросы, макросы низкоуровневого обнаружения. Устанавливает параметр cURL CURLOPT_KEYPASSWD.
Тайм-аут	Zabbix не будет тратить на обработку URL больше заданного времени (1-600 секунд). Фактически этот параметр определяет максимальное время для установления соединения с URL и максимальное время для выполнения HTTP-запроса. Таким образом, Zabbix не будет тратить на одну проверку более 2 x *Тайм-аут* секунд. Устанавливает параметр cURL CURLOPT_TIMEOUT. Дополнительную информацию о параметре *Тайм-аут* см. в разделе общие атрибуты элемента данных.
Включить траппинг	Если этот флажок установлен, элемент данных также будет функционировать как элемент данных trapper и будет принимать данные, отправленные на сервер или прокси Zabbix с помощью утилиты Zabbix sender или протокола Zabbix sender, либо будет принимать данные, отправленные на сервер Zabbix с помощью метода API `history.push`. Дополнительную информацию об отправке данных см. в разделе: Элементы данных trapper.
Разрешенные узлы сети	Отображается только если установлен флажок *Включить траппинг*. Список IP-адресов, разделенных запятыми, при необходимости в нотации CIDR, или DNS-имен. Если указано, входящие соединения будут приниматься только от узлов сети, перечисленных здесь. Если поддержка IPv6 включена, тогда '127.0.0.1', '::127.0.0.1', '::ffff:127.0.0.1' рассматриваются одинаково, 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 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 допускаются пробелы и пользовательские макросы. В этом поле допускаются макросы узла сети: {HOST.HOST}, {HOST.NAME}, {HOST.IP}, {HOST.DNS}, {HOST.CONN}.

Note:
Если поле *HTTP прокси* оставлено пустым, другим способом использования HTTP-прокси является настройка связанных с прокси переменных окружения.
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`. Более подробная информация доступна при выполнении команды оболочки: *# man curl*.

Attention:
[1] Zabbix поддерживает только файлы сертификатов и закрытых ключей в формате PEM. Если ваши данные сертификата и закрытого ключа находятся в файле формата PKCS #12 (обычно с расширением *.p12 или *.pfx), вы можете сгенерировать из него PEM-файл 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 GET элемент данных 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 предобработки JSONPath: `$.version.number`

Пример 2

Отправляйте простые POST-запросы для получения данных от таких сервисов, как Elasticsearch:

- Создайте POST-элемент данных с URL: `http://localhost:9200/str/values/_search?scroll=10s`
- Настройте следующее тело POST-запроса, 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,
```

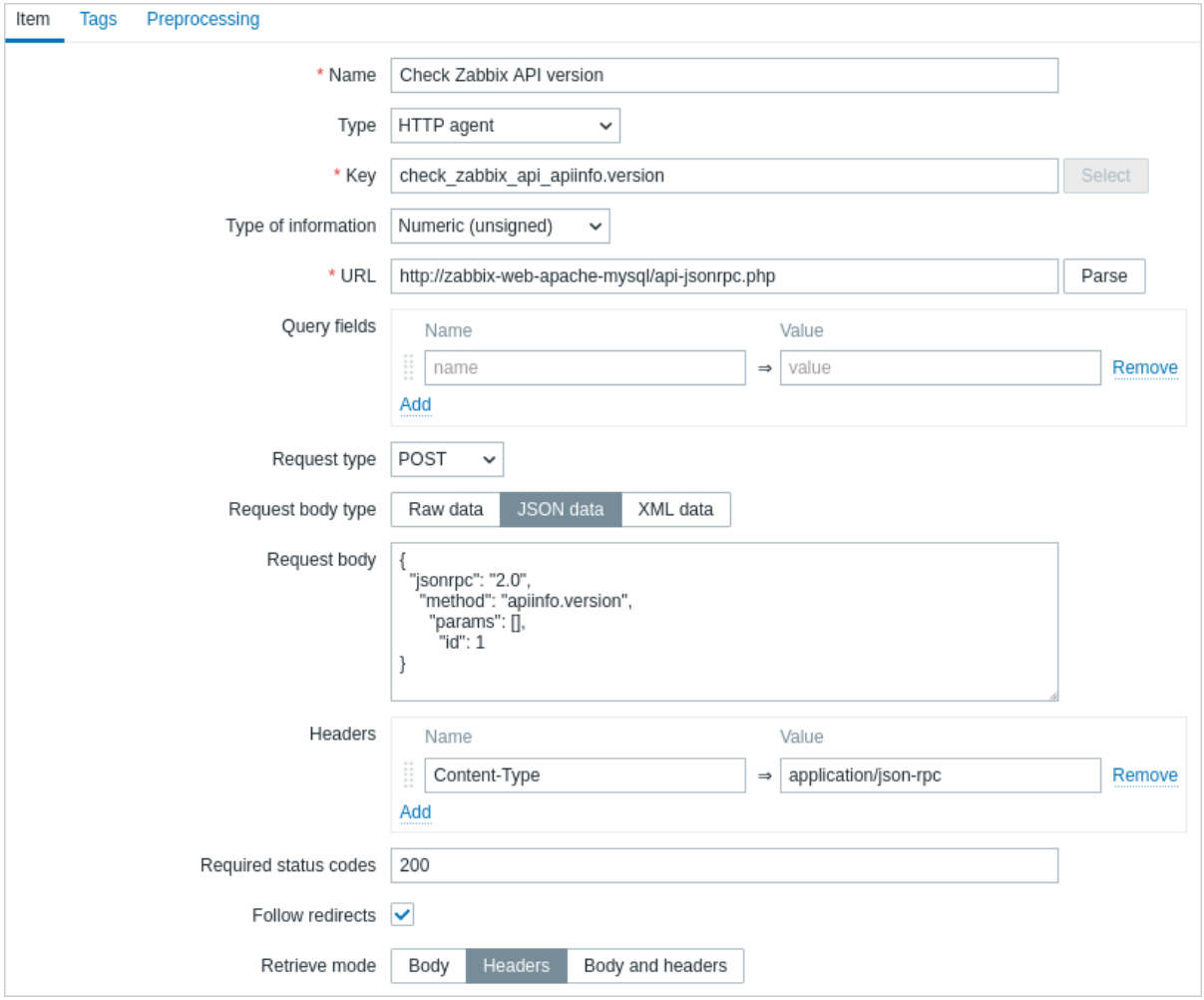
```
        "_shards": {
            "total": 5,
            "successful": 5,
            "skipped": 0,
            "failed": 0
        },
        "hits": {
            "total": 1,
            "max_score": 1.0,
            "hits": [{
                "_index": "dbl",
                "_type": "values",
                "_id": "dqX9VWEBV6sEKSMyk6sw",
                "_score": 1.0,
                "_source": {
                    "itemid": 28275,
                    "value": "0.138750",
                    "clock": 1517566136,
                    "ns": 25388713,
                    "ttl": 604800
                }
            }]
        }
    }
```


- Теперь используйте шаг предобработки JSONPath, чтобы получить значение элемента данных: `$.hits.hits[0]._source.val`

Пример 3

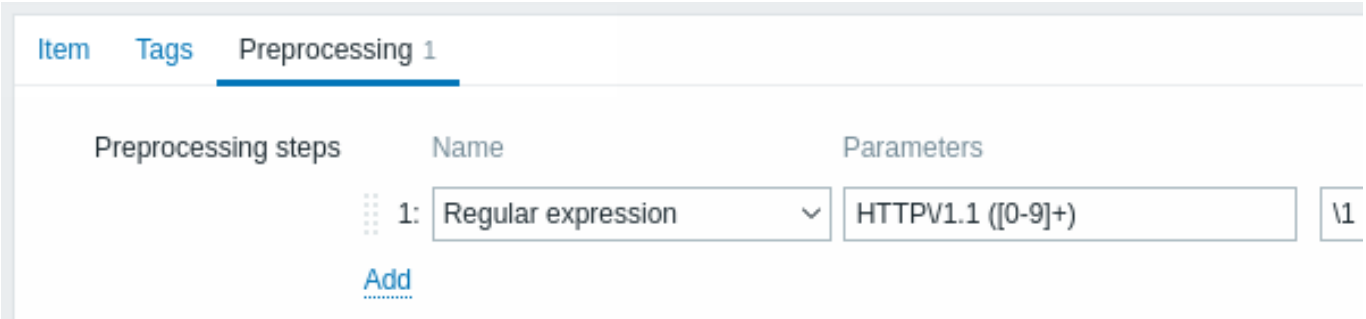
Проверка доступности Zabbix API с использованием apiinfo.version.

- Конфигурация элемента данных:

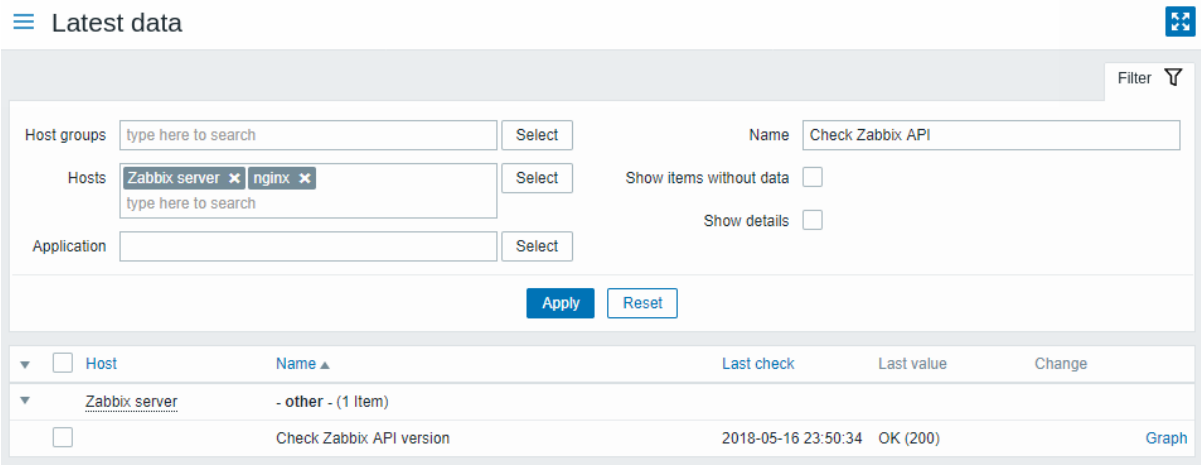


Обратите внимание на использование метода POST с данными JSON, настройку заголовков запроса и указание возвращать только заголовки:

- Предобработка значения элемента данных с помощью регулярного выражения для получения HTTP-кода:



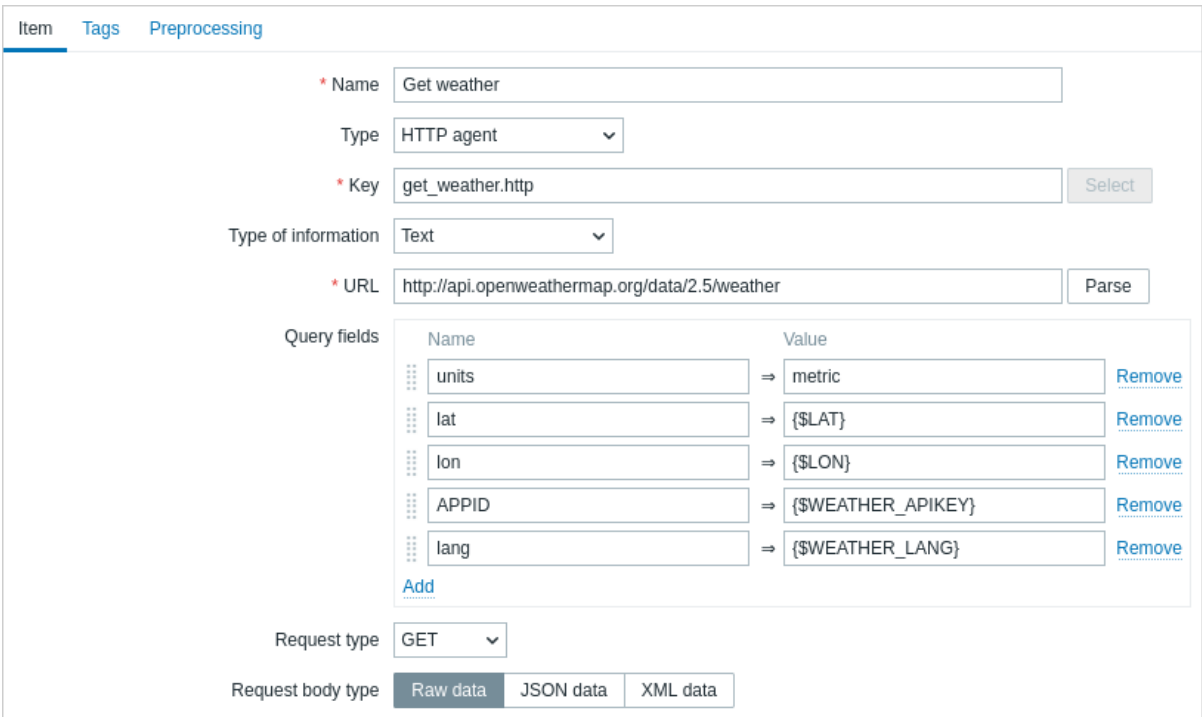
- Проверка результата в *Последних данных*:



Пример 4

Получение информации о погоде путем подключения к общедоступному сервису Openweathermap.

- Настройте мастер-элемент данных для массового сбора данных в одном JSON:



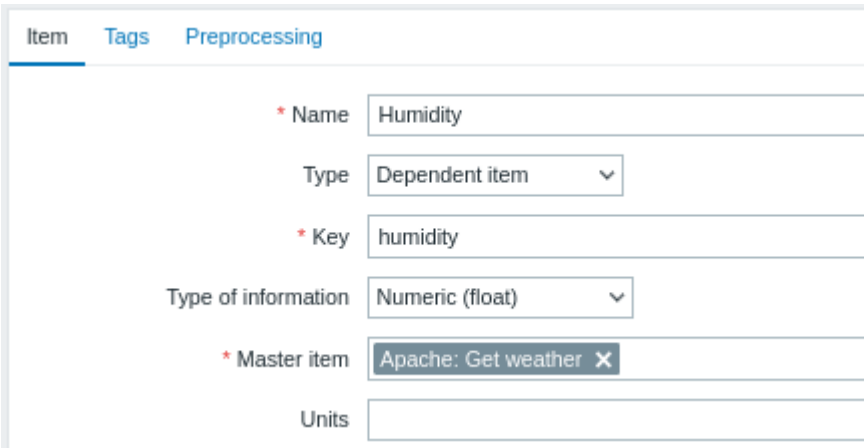
Обратите внимание на использование макросов в полях запроса. См. Openweathermap API, чтобы узнать, как их заполнять.

Пример JSON, возвращаемого в ответ на HTTP агент:

```
{
    "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
    }
}
```

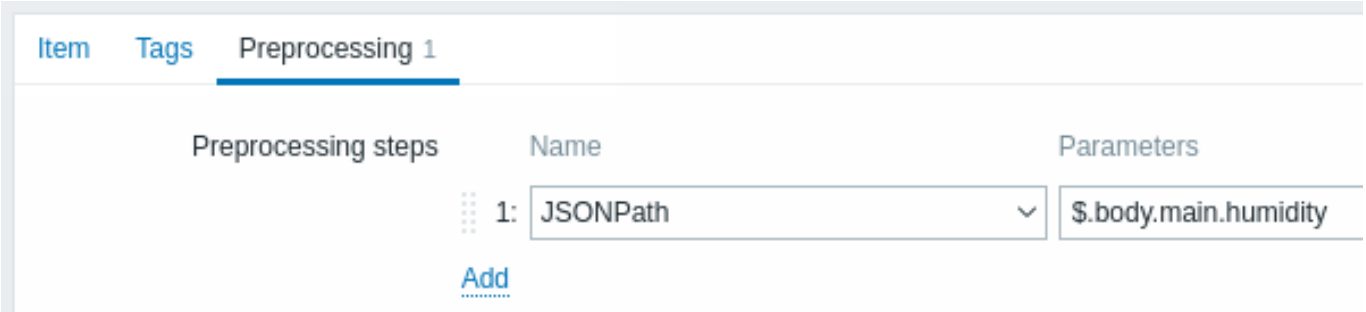

Следующая задача — настроить зависимые элементы данных, которые извлекают данные из JSON.

- Настройте пример зависимого элемента данных для влажности:

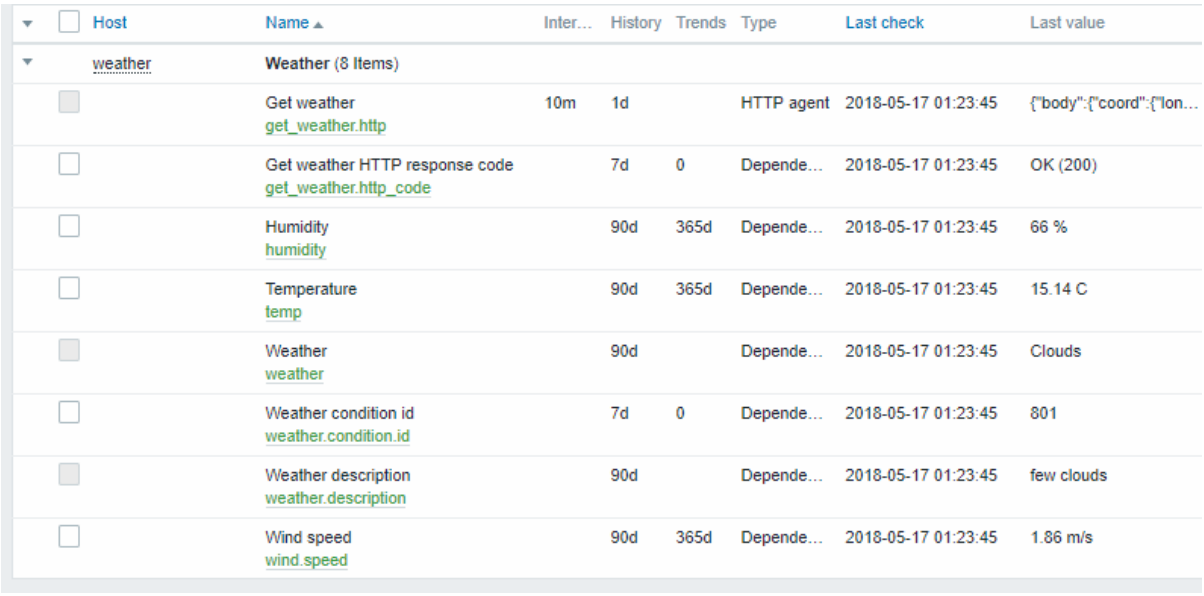


Другие погодные показатели, такие как «Температура», добавляются аналогичным образом.

- Пример предобработки значения зависимого элемента данных с помощью JSONPath:



- Проверьте результат получения данных о погоде в *Последние данные*:

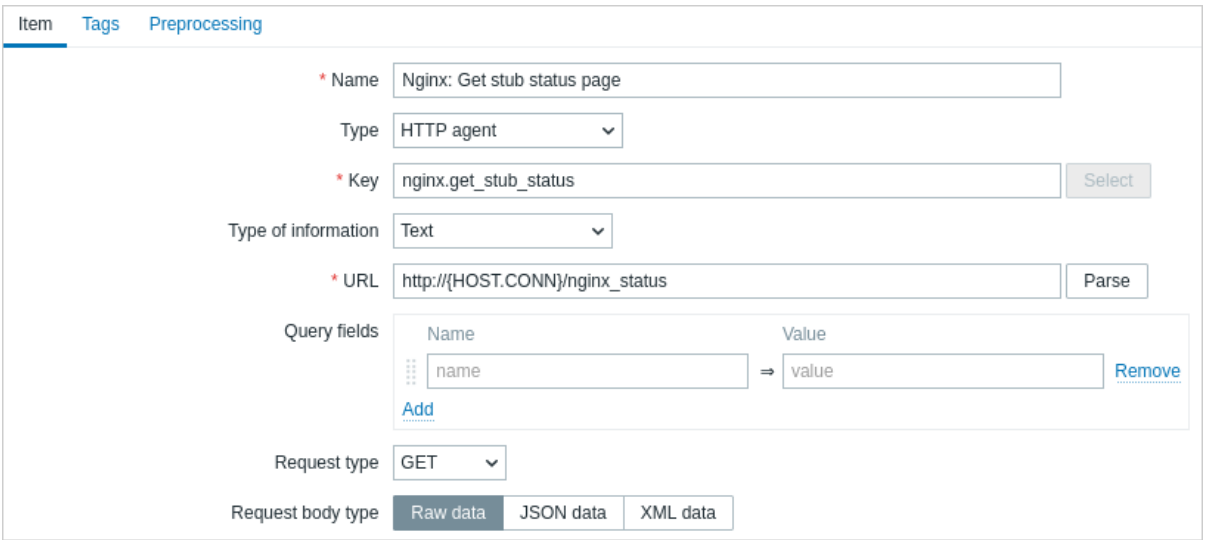


Host	Name	Inter...	History	Trends	Type	Last check	Last value
weather	**Weather** (8 Items)						
	Get weather get_weather.http	10m	1d		HTTP agent	2018-05-17 01:23:45	{"body":{"coord":{"lon...
	Get weather HTTP response code get_weather.http_code		7d	0	Depende...	2018-05-17 01:23:45	OK (200)
	Humidity 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 массовое получение её метрик.

- Настройте Nginx, следуя официальному руководству.
- Настройте мастер-элемент данных для массового сбора данных:

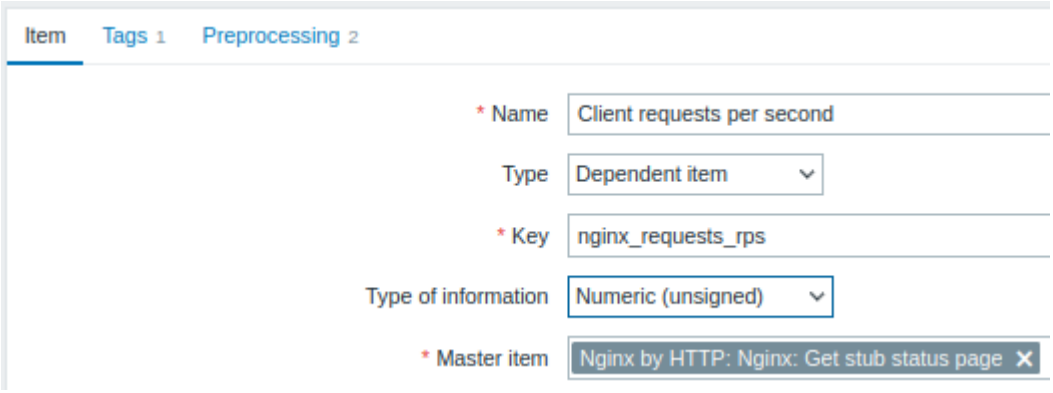


Пример вывода состояния Nginx stub:

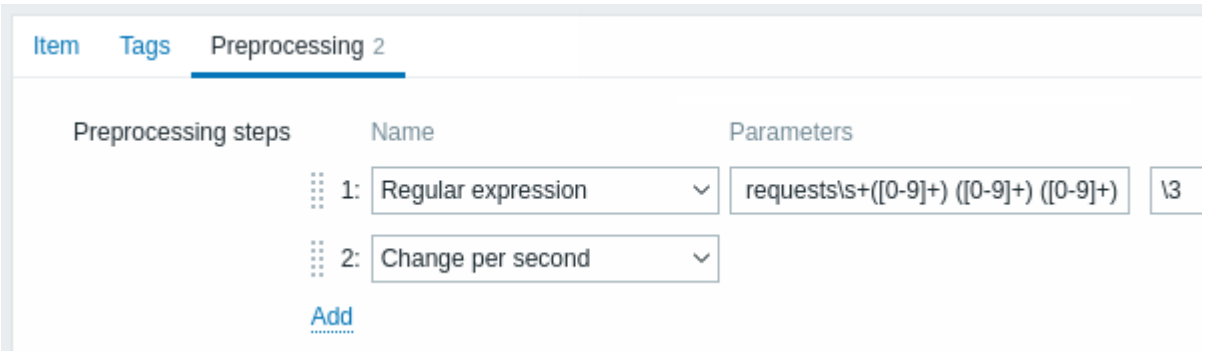
```
Active connections: 1 Active connections:
server accepts handled requests
 52 52 52
Reading: 0 Writing: 1 Waiting: 0
```

Следующая задача — настроить зависимые элементы данных, которые извлекают данные.

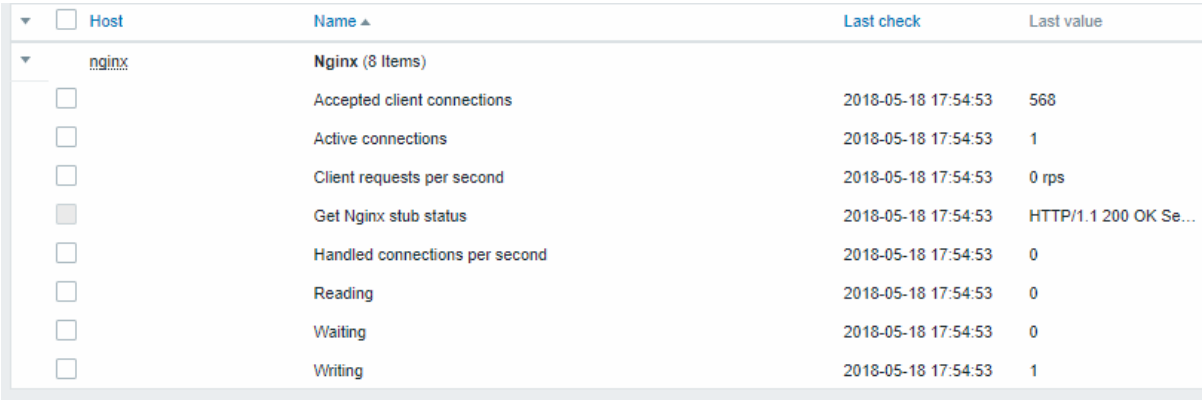
- Настройте пример зависимого элемента данных для запросов в секунду:



- Пример предобработки значения зависимого элемента данных с регулярным выражением `server accepts handled requests\s+([0-9]+) ([0-9]+) ([0-9]+)`:



- Проверьте полный результат из stub-модуля в разделе *Последние данные*:



17 Проверки Prometheus

Обзор

Zabbix может запрашивать метрики, предоставляемые в строковом формате Prometheus.

Чтобы начать сбор данных Prometheus, необходимо выполнить два шага:

- HTTP мастер-элемент данных, указывающий на соответствующую конечную точку данных, например `https://<prometheus host>/metrics`
- зависимые элементы данных, использующие вариант предобработки Prometheus для запроса необходимых данных из метрик, собранных мастер-элементом данных

Существует два варианта предобработки данных Prometheus:

- *Шаблон Prometheus* — используется в обычных элементах данных для запроса данных Prometheus
- *Prometheus в JSON* — используется в обычных элементах данных и для низкоуровневого обнаружения. В этом случае запрошенные данные Prometheus возвращаются в формате JSON.

Массовая обработка

Для зависимых метрик поддерживается массовая обработка. Чтобы включить кэширование и индексацию, **первым** шагом предварительной обработки должен быть *Шаблон Prometheus*. Когда *Шаблон Prometheus* является первым шагом предварительной обработки, тогда проанализированные данные Prometheus кэшируются и индексируются по первому

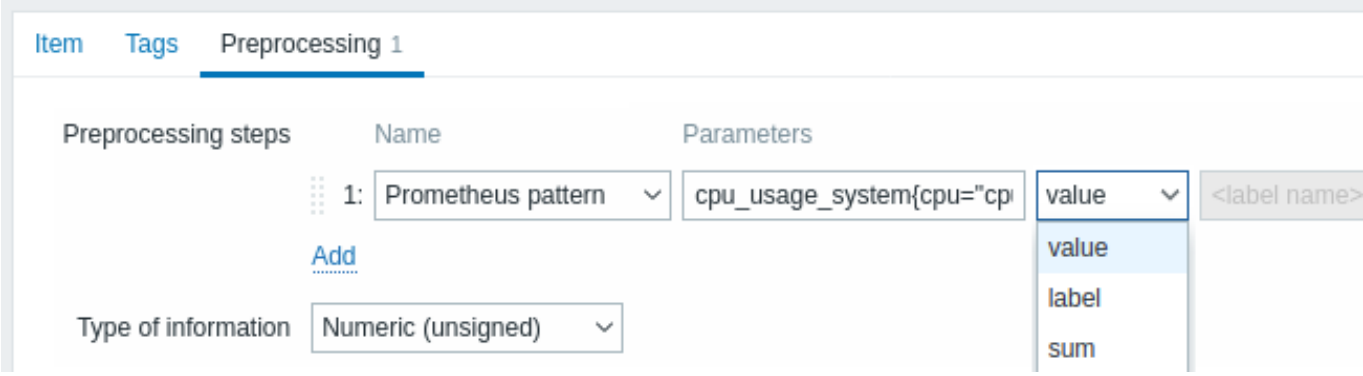
условию < >==< > в шаге предварительной обработки *Шаблон Prometheus*. Этот кэш используется повторно при обработке остальных зависимых элементов данных в этом пакете. Для достижения оптимальной производительности первая метка должна быть одной из тех, у которых наиболее разные значения.

Если другие операции предварительные обработки необходимо выполнить до первого шага, то их необходимо переместить либо в основной элемент данных, либо в новый зависимый элемент данных, который затем можно использовать в качестве основного элемента данных для этих зависимых элементов данных.

Настройка

Если у вас настроен основной элемент данных HTTP, вам нужно создать зависимый элемент данных, который использует шаг предварительной обработки Prometheus:

- введите общие параметры зависимого элемента данных в диалоге настройки
- перейдите на вкладку **Предобработка**
- выберите шаг предобработки Prometheus (*Шаблон Prometheus* или *Prometheus в JSON*)



Следующие параметры специфичны для опции предобработки *Шаблон Prometheus*:

Параметр	Описание	Примеры
Шаблон	Чтобы определить необходимый шаблон данных вы можете использовать язык запросов, который похож на язык запросов Prometheus (см. таблицу сравнения), например: <имя метрики> - выбор по названию метрики {__name__=”<имя метрики>”} - выбор по названию метрики {__name__=~”<регулярное выражение>”} - выбор по названию метрики, совпадающему с регулярным выражением {<имя метки>=”<значение метки>”,...} - выбор по имени метки {<имя метки>=~”<регулярное выражение>”,...} - выбор по имени метки, совпадающему с регулярным выражением {__name__=~”.*”}==<значение> - выбор по значению метрики Или сочетание вышеперечисленного: <имя метрики>{<имя метки1>=”<значение метки1>”,<имя метки2>=~”<регулярное выражение>”,...}==<значение> Значением метки может быть любая последовательность символов UTF-8, однако символы обратной косой черты, двойных кавычек и перевода строки необходимо экранированы как \\, \" и \n соответственно; другие символы не требуется экранировать.	*wmi_os_physical_memory_free_bytes* *cpu_usage_system{cpu=”cpu-total”}* *cpu_usage_system{cpu=~”.*”}* *cpu_usage_system{cpu=”cpu-total”,host=~”.*”}* *wmi_service_state{name=”dhcp”}==1* *wmi_os_timezone{timezone=~”.*”}==1*

Параметр	Описание	Примеры
Результат обработки	Укажите, следует ли вернуть значение, метку или применить соответствующую функцию (если шаблон соответствует нескольким строкам и результат необходимо агрегировать): **значение** - получение значения метрики (ошибка, если совпадут несколько строк) **метка** - получение значения метки, указанной в поле *Метка* (ошибка, если совпадут несколько метрик) **сумма** - получение суммы значений **мин** - получение минимального значения **макс** - получение максимального значения **сред** - получение среднего значения **кол-во** - получение количества значений Это поле доступно только для опции *Шаблон Prometheus*.	Смотрите также примеры использования ниже.
Вывод	Задайте название метки (опционально). В этом случае вернется значение, соответствующее названию метки. Это поле доступно только для шага *Шаблон Prometheus*, если в поле *Результат обработки* выбрано 'Метка'.	

Примеры использования параметров

1. Наиболее распространенный вариант использования - получение **значения**. Чтобы получить значение `/var/db` из:

   `node_disk_usage_bytes{path="/var/cache"} 2.1766144e+09<br> node_disk_usage_bytes{path="/var/db"} 20480<br> node_disk_usage_bytes{path="/var/dpkg"} 8192<br> node_disk_usage_bytes{path="/var/empty"} 4096`

используйте следующие параметры:

- *Шаблон* - `node_disk_usage_bytes{path="/var/db"}`
- *Результат обработки* - выберите 'значение'

2. Вас также может заинтересовать **усреднённое** значение по всем параметрам `node_disk_usage_bytes`:

- *Шаблон* - `node_disk_usage_bytes`
- *Результат обработки* - выберите 'сред'

3. Хотя Prometheus поддерживает только числовые данные, часто используется обходное решение, которое также позволяет получить соответствующее текстовое значение. Это можно сделать при помощи фильтра и указания метки. Так, чтобы получить значение метки 'color' из

   `elasticsearch_cluster_health_status{cluster="elasticsearch",color="green"} 1<br> elasticsearch_cluster 0<br> elasticsearch_cluster_health_status{cluster="elasticsearch",color="yellow"} 0`

используйте следующие параметры:

- *Шаблон* - `elasticsearch_cluster_health_status {cluster="elasticsearch"} == 1`
- *Результат обработки* - выберите 'метка'
- *Метка* - укажите 'color'

Фильтр (на основе числового значения '1') совпадёт с соответствующей строкой, а метка вернет описание состояния работоспособности (в настоящий момент 'green'; но возможны также 'red' или 'yellow').

Prometheus в JSON

Данные из Prometheus можно использовать для низкоуровневого обнаружения. В этом случае данные необходимы в формате JSON, и опция предобработки *Prometheus в JSON* вернет их в нужном формате.

Для получения более подробных сведений смотрите Обнаружение с использованием данных Prometheus.

Сравнение языка запросов

В следующей таблице перечислены различия и сходства между PromQL и языком запросов предобработки Prometheus в Zabbix.

Мгновенный векторный селектор PromQL [en]		Предобработка Prometheus в Zabbix
Различия		
Цель запроса	Сервер Prometheus	Простой текст в формате представления Prometheus
Возвращает	Мгновенный вектор	Значение метрики или метки (Шаблон Prometheus) Массив метрик для одиночного значения в JSON (Prometheus в JSON)
Операторы сопоставления меток	**=**, **!=**, **=~**, **!~**	**=**, **!=**, **=~**, **!~**
Регулярное выражение, используемое при сопоставлении метки или имени метрики	RE2	PCRE
Операторы сравнения	Смотрите список [en]	Только **==** (равно) поддерживается для фильтрации значений
Сходства		
Выбор по названию метрики, равному строке	<имя метрики> или {__name__=”<имя метрики>”}	<имя метрики> или {__name__=”<имя метрики>”}
Выбор по имени метрики, которое соответствует регулярному выражению	{__name__=~”<регулярное выражение>”}	{__name__=~”<регулярное выражение>”}
Выбор по значению <имя метки>, равному строке	{<имя метки>=”<значение метки>”,...}	{<имя метки>=”<значение метки>”,...}
Выбор по значению <имя метки>, которое соответствует регулярному выражению	{<имя метки>=~”<регулярное выражение>”,...}	{<имя метки>=~”<регулярное выражение>”,...}
Выбор по значению, равному строке	{__name__=~”.*”} == <значение>	{__name__=~”.*”} == <значение>

18 Скриптовые элементы данных

Обзор

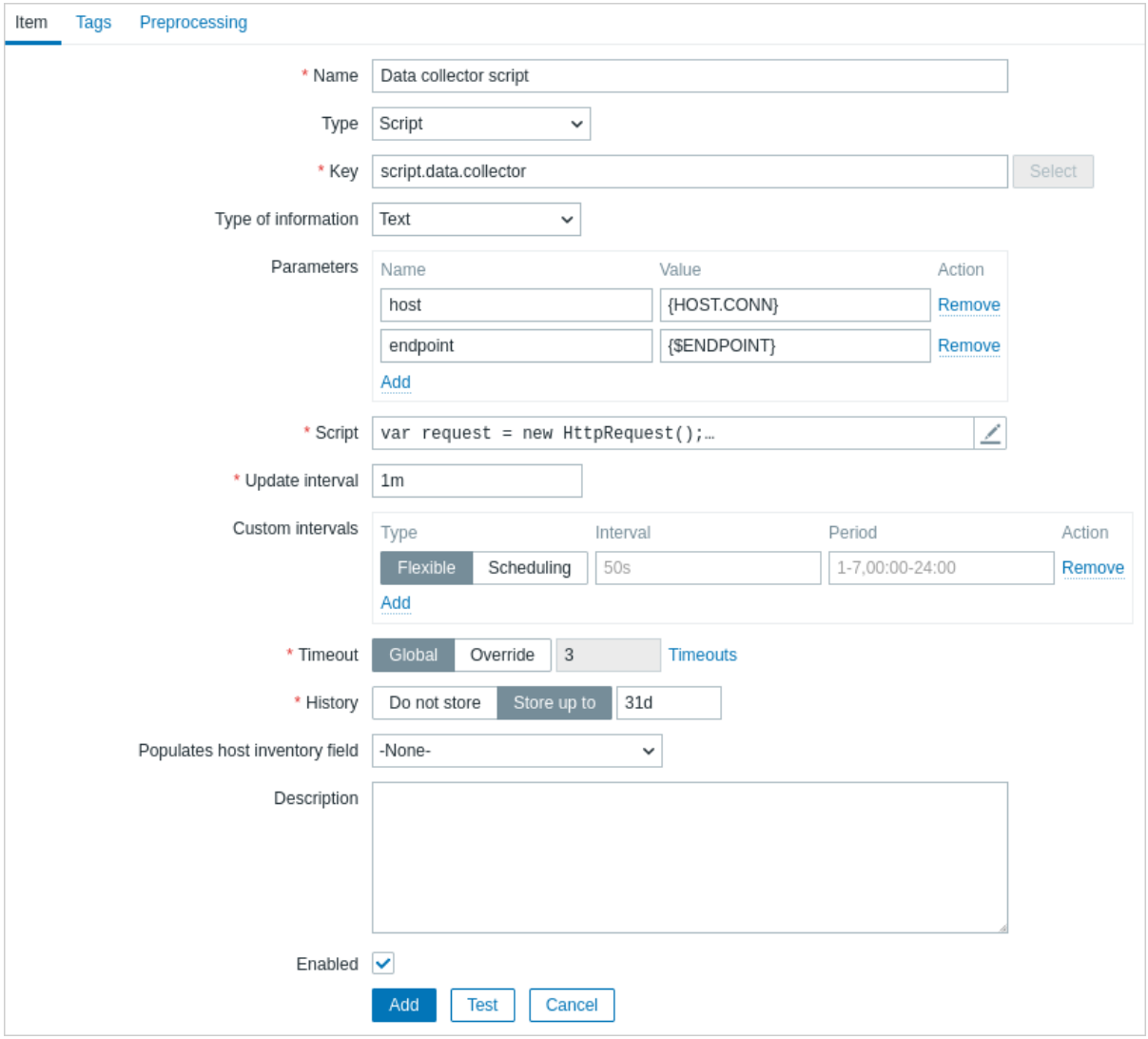
Элементы данных типа «Скрипт» можно использовать для сбора данных путем выполнения пользовательского кода JavaScript с возможностью получения данных по HTTP/HTTPS. В дополнение к скрипту можно указать необязательный список параметров (пар имя-значение) и тайм-аут.

Этот тип элемента данных может быть полезен в сценариях сбора данных, требующих нескольких шагов или сложной логики. Например, элемент данных типа «Скрипт» можно настроить так, чтобы он выполнял HTTP-вызов, затем каким-либо образом обрабатывал данные, полученные на первом шаге, и передавал преобразованное значение во второй HTTP-вызов.

Элементы данных типа «Скрипт» обрабатываются опрашивателями сервера Zabbix или прокси.

Настройка

В поле *Тип* формы настройки элемента данных выберите Script, затем заполните обязательные поля.



Все обязательные поля ввода отмечены красной звёздочкой.

Поля, требующие специальной информации для элементов данных типа Script:

Поле	Описание
Ключ	Введите уникальный ключ, который будет использоваться для идентификации элемента данных.
Параметры	Укажите переменные, которые будут переданы в скрипт как пары атрибут-значение. Поддерживаются пользовательские макросы. Чтобы узнать, какие встроенные макросы поддерживаются, выполните поиск по строке ”Script-type item” в таблице поддерживаемых макросов.
Скрипт	Введите код JavaScript в модальном редакторе, который открывается при щелчке в поле параметра или по значку карандаша рядом с ним. Этот код должен содержать логику возврата значения метрики. Код имеет доступ ко всем параметрам и всем дополнительным объектам JavaScript, добавленным Zabbix. См. также: Руководство по JavaScript.
Тайм-аут	Тайм-аут выполнения JavaScript (1-600s; при его превышении будет возвращена ошибка). Обратите внимание, что в зависимости от скрипта срабатывание тайм-аута может занять больше времени. Дополнительную информацию о параметре *Timeout* см. в разделе общие атрибуты элемента данных.

Примеры

Простой сбор данных

Получение содержимого *https://www.example.com/release_notes*:

- Создайте элемент данных с типом ”Скрипт”.

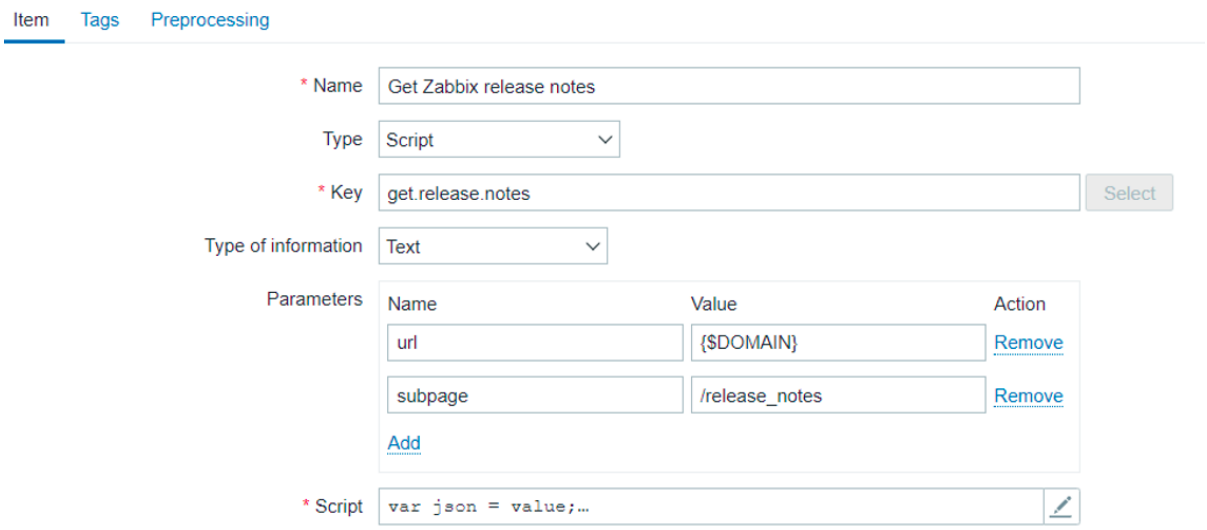
- В поле *Скрипт* введите следующий код:

```
var request = new HttpRequest();
return request.get("https://www.example.com/release_notes");
```

Сбор данных с параметрами

Соберите содержимое определенной страницы и используйте параметры:

- Создайте элемент данных типа ”Script” и два параметра:
  - **url : {$DOMAIN}** (пользовательский макрос {$DOMAIN} должен быть определен, предпочтительно на уровне узла сети)
  - **subpage : /release_notes**



- В поле *Script* введите:

```
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 значением элемента данных ”1”:

- Создайте элемент данных с типом ”Скрипт”.

- В поле *Скрипт* введите следующий код:

```
Zabbix.log(3, 'Log test');
return 1;
```

19 Элементы данных браузера

Обзор

Элементы данных браузера позволяют выполнять мониторинг сложных веб-сайтов и веб-приложений с использованием браузера.

> **Attention:**
> Поддержка элементов данных браузера в настоящее время является экспериментальной.

Элементы данных браузера собирают данные, выполняя определяемый пользователем код JavaScript и получая данные по HTTP/HTTPS. Этот элемент данных может имитировать такие действия, связанные с браузером, как щелчки, ввод текста, переход по веб-страницам и другие взаимодействия пользователя с веб-сайтами или веб-приложениями.

Помимо скрипта, можно указать необязательный список параметров (пар имя-значение) и тайм-аут.

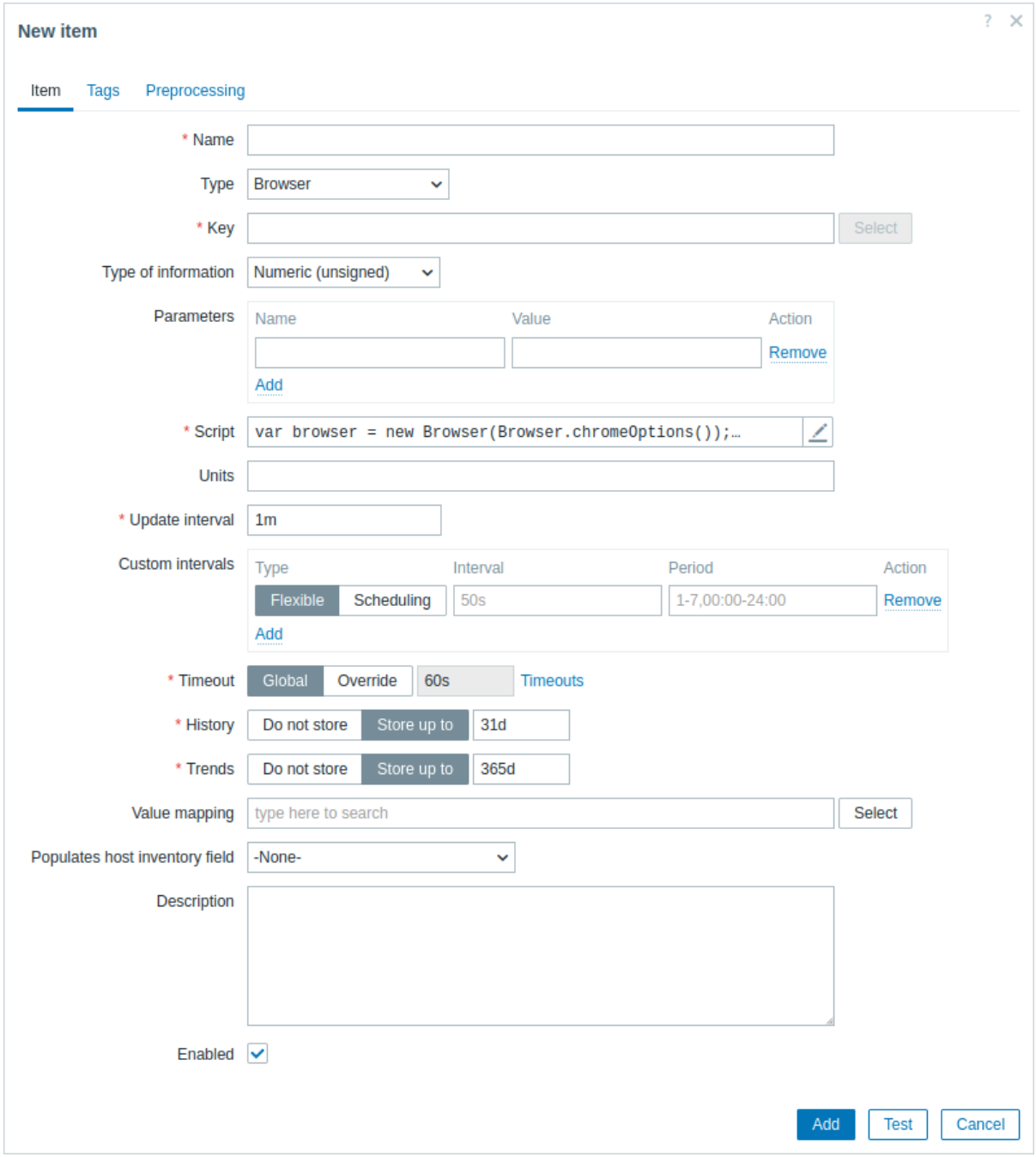
> **Attention:**
> Элемент данных частично реализует стандарт W3C WebDriver с использованием либо Selenium Server, либо обычного WebDriver (например, ChromeDriver) в качестве конечной точки веб-тестирования. Чтобы элемент данных работал, задайте конечную точку в параметре `WebDriverURL` файла конфигурации Zabbix сервера/прокси (при использовании ChromeDriver см. соображения безопасности). Для повышения производительности рекомендуется использовать выделенный сервер для среды веб-тестирования.

Проверки элементов данных браузера выполняются и обрабатываются browser poller'ами сервера или прокси Zabbix. При необходимости вы можете настроить количество предварительно созданных экземпляров browser poller'ов в параметре `StartBrowserPollers` файла конфигурации Zabbix сервера/прокси.

Для мониторинга сложных веб-сайтов и веб-приложений доступен шаблон Website by Browser как готовый шаблон.

Настройка

В поле *Тип* формы настройки элемента данных выберите Browser, а затем заполните обязательные поля.



Все обязательные поля ввода отмечены красной звездочкой.

Поля, требующие специальной информации для элементов данных Browser:

Поле	Описание
Ключ	Введите уникальный ключ, который будет использоваться для идентификации элемента данных.
Параметры	Укажите переменные, которые будут переданы в скрипт как пары атрибут-значение. Поддерживаются пользовательские макросы. Чтобы увидеть, какие встроенные макросы поддерживаются, выполните поиск по строке ”Browser-type item” в таблице поддерживаемых макросов.
Скрипт	Введите код JavaScript в модальном редакторе, который открывается при щелчке в поле параметра или по значку карандаша рядом с ним. Этот код должен содержать логику возврата значения метрики. Код имеет доступ ко всем параметрам, всем дополнительным объектам JavaScript и объектам JavaScript элемента данных Browser, добавленным Zabbix. См. также: Руководство по JavaScript.

Поле	Описание
Тайм-аут	Тайм-аут выполнения JavaScript (1-600 с; при превышении будет возвращена ошибка). Обратите внимание, что в зависимости от скрипта срабатывание тайм-аута может занять больше времени. Дополнительную информацию о параметре *Timeout* см. в разделе общие атрибуты элемента 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.
3. Собирает записи производительности и статистику сессии и возвращает их в виде строки JSON.

В поле *Скрипт* 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 capabilities

Следующий скрипт:

1. Инициализирует сеанс браузера для доступного браузера на основе первого совпавшего браузера в порядке, указанном в скрипте.
2. Определяет capabilities 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 (кроме Safari) инициализируются в режиме headless, то есть графический пользовательский интерфейс (GUI) браузера не отображается.

Следующий скрипт инициализирует сеанс браузера с включённым 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 = [];

// To initialize a Firefox session with GUI, uncomment the following lines:
// var opts = Browser.firefoxOptions();
// opts.capabilities.alwaysMatch['moz:firefoxOptions'].binary = 'usr/bin/firefox';
// opts.capabilities.alwaysMatch['moz:firefoxOptions'].args = [];

// To initialize a Microsoft Edge session with GUI, uncomment the following lines:
// 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 удалённом сервере или в контейнере, вы можете использовать клиент Virtual Network Computing (VNC) для подключения к VNC-серверу машины. Это позволит вам удалённо просматривать GUI 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 (задаётся как параметры, см. ниже).
3. Переходит по URL-адресу (задаётся как параметр, см. ниже).
4. Собирает статистику сеанса, делает снимок экрана и добавляет его к собранной статистике.
5. Обрабатывает ошибки, сохраняя сообщения об ошибках и снимок экрана.
6. Возвращает собранные результаты в виде JSON-строки.

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 сеанс браузера.
2. Переходит на страницу (указывается как параметр, см. ниже).
3. Вводит имя пользователя и пароль (указываются как параметры, см. 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, сохраняя сообщения об ошибках и снимок экрана.
8. Возвращает собранные результаты в виде JSON-строки.

Скрипт также использует параметры из формы настройки элемента данных:

- webURL - http://{HOST.CONN}/index.php
- username - {$USERNAME}
- password - {$PASSWORD}

В поле *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 повторяющихся элементов из массива (см. шаг 5).
3. Переходит на страницу (указанную в параметрах, см. ниже).
4. Находит ссылки на странице.
5. Удаляет повторяющиеся ссылки, чтобы обеспечить их уникальность.
6. Извлекает только ссылки, начинающиеся с "http".
7. Форматирует извлеченные ссылки в определенную структуру.
8. Обрабатывает ошибки, сохраняя сообщения об ошибках и снимок экрана.
9. Возвращает собранные результаты в виде JSON-строки.

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);
}
```

4 История и динамика изменений

Обзор

История и тренды — это два способа хранения собранных данных в Zabbix.

Если история хранит каждое собранное значение, то тренды хранят усреднённую информацию на почасовой основе и поэтому требуют меньше ресурсов.

Хранение истории

Вы можете указать, сколько дней будет храниться история:

- в форме свойств элемента данных
- при массовом обновлении элементов данных
- при настройке задач housekeeper

Все более старые данные будут удаляться housekeeper.

В общем случае настоятельно рекомендуется хранить историю как можно меньшее количество дней, чтобы не перегружать базу данных большим количеством исторических значений.

Вместо длительного хранения истории вы можете дольше хранить данные трендов. Например, можно хранить историю 14 дней, а тренды — 5 лет.

Чтобы получить хорошее представление о том, сколько места требуется для данных истории по сравнению с данными трендов, обратитесь к странице расчета размера базы данных.

Даже при более коротком сроке хранения истории вы все равно сможете просматривать старые данные на графиках, так как для отображения более старых данных графики будут использовать значения трендов.

Attention:
Если для истории установлено значение '0', элемент данных будет обновлять только зависимые элементы данных и инвентарные данные. Функции триггеров не будут вычисляться, поскольку вычисление триггеров основано только на данных истории.

Note:
В качестве альтернативного способа сохранения истории рассмотрите возможность использования функции экспорта истории загружаемых модулей.

Хранение трендов

Тренды — это встроенный механизм сокращения объема исторических данных, который сохраняет минимальное, максимальное, среднее значение и общее количество значений за каждый час для числовых типов данных.

Вы можете указать, сколько дней будут храниться тренды:

- в форме свойств элемента данных
- при массовом обновлении элементов данных
- при настройке задач Housekeeper

Обычно тренды могут храниться значительно дольше, чем история. Все более старые данные будут удалены housekeeper.

Сервер Zabbix накапливает данные трендов во время работы в кэше трендов по мере поступления данных. Сервер записывает в базу данных тренды за **предыдущий час** каждого элемента данных (откуда их может получить веб-интерфейс) в следующих случаях:

- сервер получает первое значение элемента данных за текущий час
- до конца текущего часа остается 5 минут или меньше, и при этом для элемента данных все еще нет значений за текущий час
- сервер останавливается

Чтобы увидеть тренды на графике, необходимо подождать как минимум до начала следующего часа (если элемент данных обновляется часто) и как максимум до конца следующего часа (если элемент данных обновляется редко), то есть не более 2 часов.

Когда сервер записывает кэш трендов и в базе данных уже есть тренды за этот час (например, если сервер был перезапущен в середине часа), серверу необходимо использовать операторы update вместо простых insert. Поэтому в более крупной установке, если требуется перезапуск, желательно останавливать сервер в конце одного часа и запускать в начале следующего часа, чтобы избежать перекрытия данных трендов.

Таблицы истории никак не участвуют в генерации трендов.

Attention:
Если для трендов установлено значение '0', сервер Zabbix вообще не вычисляет и не сохраняет тренды.

Note:
Тренды вычисляются и сохраняются с тем же типом данных, что и исходные значения. В результате вычисления среднего значения для значений беззнакового типа данных округляются, и чем меньше интервал значений, тем менее точным будет результат. Например, если элемент данных имеет значения 0 и 1, среднее значение будет 0, а не 0.5.
Кроме того, перезапуск сервера может привести к потере точности при вычислении среднего значения для беззнакового типа данных за текущий час.

5 Пользовательские параметры

Обзор

Иногда может потребоваться выполнить проверку агента, которая не предопределена в Zabbix. В этом случае помогут пользовательские параметры.

Вы можете написать команду, которая получает нужные вам данные, и включить её в пользовательский параметр в файле конфигурации агента (параметр конфигурации 'UserParameter').

Пользовательский параметр имеет следующий синтаксис:

```
UserParameter=<key>,<command>
```

Как видите, пользовательский параметр также содержит ключ. Ключ будет необходим при настройке элемента данных. Укажите ключ по своему выбору, на который будет удобно ссылаться (он должен быть уникальным в пределах узла сети).

Перезапустите агент или используйте опцию управления во время выполнения агента, чтобы применить новый параметр, например:

```
zabbix_agentd -R userparameter_reload
```

Затем при настройке элемента данных укажите ключ, чтобы сослаться на команду из пользовательского параметра, которую необходимо выполнить.

Пользовательские параметры — это команды, выполняемые агентом Zabbix. Обратите внимание, что до этапов предобработки значений элемента данных может быть возвращено до 16 МБ данных.

/bin/sh используется в качестве интерпретатора командной строки в операционных системах UNIX. Пользовательские параметры подчиняются тайм-ауту проверки агента; если время ожидания истекает, порождённый процесс пользовательского параметра завершается.

См. также:

- Пошаговое руководство по использованию пользовательских параметров
- Выполнение команд

Примеры простых пользовательских параметров

Простая команда:

```
UserParameter=ping,echo 1
```

Агент будет всегда возвращать '1' для элемента данных с ключом 'ping'.

Более сложный пример:

```
UserParameter=mysql.ping,mysqladmin -uroot ping|grep -c alive
```

Агент будет возвращать '1', если MySQL сервер доступен, '0' - в противном случае.

Гибкие пользовательские параметры

Гибкие пользовательские параметры принимают параметры вместе с ключом. Таким образом, гибкий пользовательский параметр может служить основой для создания нескольких элементов данных.

Гибкие пользовательские параметры имеют следующий синтаксис:

```
UserParameter=key[*],command
```

Параметр	Описание
Key	Уникальный ключ элемента данных. [*] определяет, что этот ключ принимает параметры в квадратных скобках. Параметры задаются при настройке элемента данных.
Command	Команда, которая будет выполнена для получения значения ключа. *Только для гибких пользовательских параметров*: Вы можете использовать позиционные ссылки $1…$9 в команде для обращения к соответствующему параметру в ключе элемента данных. Zabbix разбирает параметры, заключённые в [ ] ключа элемента данных, и соответствующим образом подставляет $1,…,$9 в команду. $0 будет заменён исходной командой (до раскрытия $0,…,$9), которая должна быть выполнена. Позиционные ссылки интерпретируются независимо от того, заключены ли они в двойные (") или одинарные (') кавычки. Чтобы использовать позиционные ссылки без изменений, укажите двойной знак доллара — например, awk '{print $$2}'. В этом случае $$2 фактически превратится в $2 при выполнении команды.

Attention:
Позиционные ссылки со знаком $ ищутся и заменяются агентом Zabbix только для гибких пользовательских параметров. Для простых пользовательских параметров такая обработка ссылок пропускается, и, следовательно, экранирование любого знака $ не требуется.

Attention:
Некоторые символы по умолчанию не допускаются в пользовательских параметрах. Полный список символов см. в UnsafeUserParameters.

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 подсчета количества строк в файле.

```
wc[/etc/passwd,root]
wc[/etc/services,zabbix]
```

Результат команды

Возвращаемое значение команды представляет собой стандартный вывод вместе со стандартным выводом ошибок, сформированным командой.

> **Attention:**
> Элемент данных, возвращающий текст (символьный, журнал или информацию типа text), не станет неподдерживаемым в случае вывода в стандартный поток ошибок.

Возвращаемое значение ограничено 16 МБ (включая завершающие пробельные символы, которые усекаются); также применяются ограничения базы данных.

Пользовательские параметры, возвращающие текст (символьный, журнал или информацию типа text), также могут возвращать пробельные символы. В случае недопустимого результата элемент данных станет неподдерживаемым.

1 Расширение Zabbix агентов

В этом руководстве приведены пошаговые инструкции, как расширить функциональность Zabbix агента с помощью пользовательского параметра.

Шаг 1

Напишите скрипт или составьте командную строку для получения необходимого параметра.

Например, мы можем 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 zabbix_agentd.conf:

```
UserParameter=mysql.questions,mysqladmin -uroot status | cut -f4 -d":" | cut -f1 -d"S"
```

mysql.questions - это уникальный идентификатор. Здесь может быть любой допустимый идентификатор, например, *queries*.

Протестируйте этот параметр, используя Zabbix агент с флагом ”-t” (если запускаете под root, стоит отметить, что агент может иметь отличающиеся права при работе демоном):

```
zabbix_agentd -t mysql.questions
```

Шаг 3

Перезагрузите пользовательские параметры из файла конфигурации, выполнив:

```
zabbix_agentd -R userparameter_reload
```

Вы также можете перезапустить агента вместо выполнения команды управления работой.

Протестируйте этот параметр, используя утилиту zabbix_get.

Шаг 4

Добавьте новый элемент данных с Ключом=mysql.questions к наблюдаемому узлу сети. Тип этого элемента данных должен быть или Zabbix агент, или Zabbix агент (активный).

Помните, очень важно указать корректный тип возвращаемого значения в Zabbix сервере. Иначе Zabbix не будет принимать значения.

6 Счётчики производительности Windows

Обзор

Вы можете эффективно мониторить счетчики производительности Windows используя ключ *perf_counter[]*.

Например:

```
perf_counter["\Processor(0)\Interrupts/sec"]
```

или

```
perf_counter["\Processor(0)\Interrupts/sec", 10]
```

Для получения более подробной информации об этом ключе и его эквиваленте только для Английского языка `perf_counter_en`, смотрите специфичные ключи элементов данных для Windows.

Чтобы получить полный список счетчиков производительности для мониторинга, вы можете выполнить:

```
typeperf -qx
```

Вы также можете использовать низкоуровневое обнаружение для обнаружения нескольких экземпляров объектов счетчиков производительности Windows и автоматизировать создание элементов данных perf_counter по множеству экземпляров объектов.

Числовое представление

Windows поддерживает числовые представления (индексы) для имен объектов и счетчиков производительности. Zabbix поддерживает эти числовые представления в качестве параметров для ключей элементов данных `perf_counter`, `perf_counter_en` и в параметрах конфигурации `PerfCounter`, `PerfCounterEn`.

Однако использовать их не рекомендуется, если только вы не можете гарантировать, что ваши числовые индексы сопоставляются с правильными строками на конкретных узлах сети. Если вам нужно создать переносимые элементы данных, которые работают на разных узлах сети с различными локализованными версиями Windows, вы можете использовать ключ `perf_counter_en` или параметр конфигурации `PerfCounterEn`, которые позволяют использовать английские имена независимо от системной локали.

Чтобы узнать числовые эквиваленты, запустите **regedit**, затем найдите *Counter* в *HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows NT\CurrentVersion\Perflib\009*.

Запись реестра содержит информацию примерно такого вида:

```
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 числа для каждой строковой части счетчика производительности, например в '\System\% Processor Time':

```
System → 2
% Processor Time → 6
```

Затем эти числа можно использовать для представления пути в числовом виде:

```
\2\6
```

Параметры счетчиков производительности

Вы можете использовать некоторые параметры PerfCounter для мониторинга счётчиков производительности Windows.

Например, вы можете добавить следующее содержимое в файл конфигурации Zabbix агента:

```
PerfCounter=UserPerfCounter1,"\Memory\Page Reads/sec",30

PerfCounter=UserPerfCounter2,"\4\24",30
```

С такими параметрами, вы можете просто использовать ключи *UserPerfCounter1* или *UserPerfCounter2* при создании соответствующих элементов данных.

Не забудьте перезапустить Zabbix агента после внесения изменений в файл конфигурации.

7 Массовое обновление

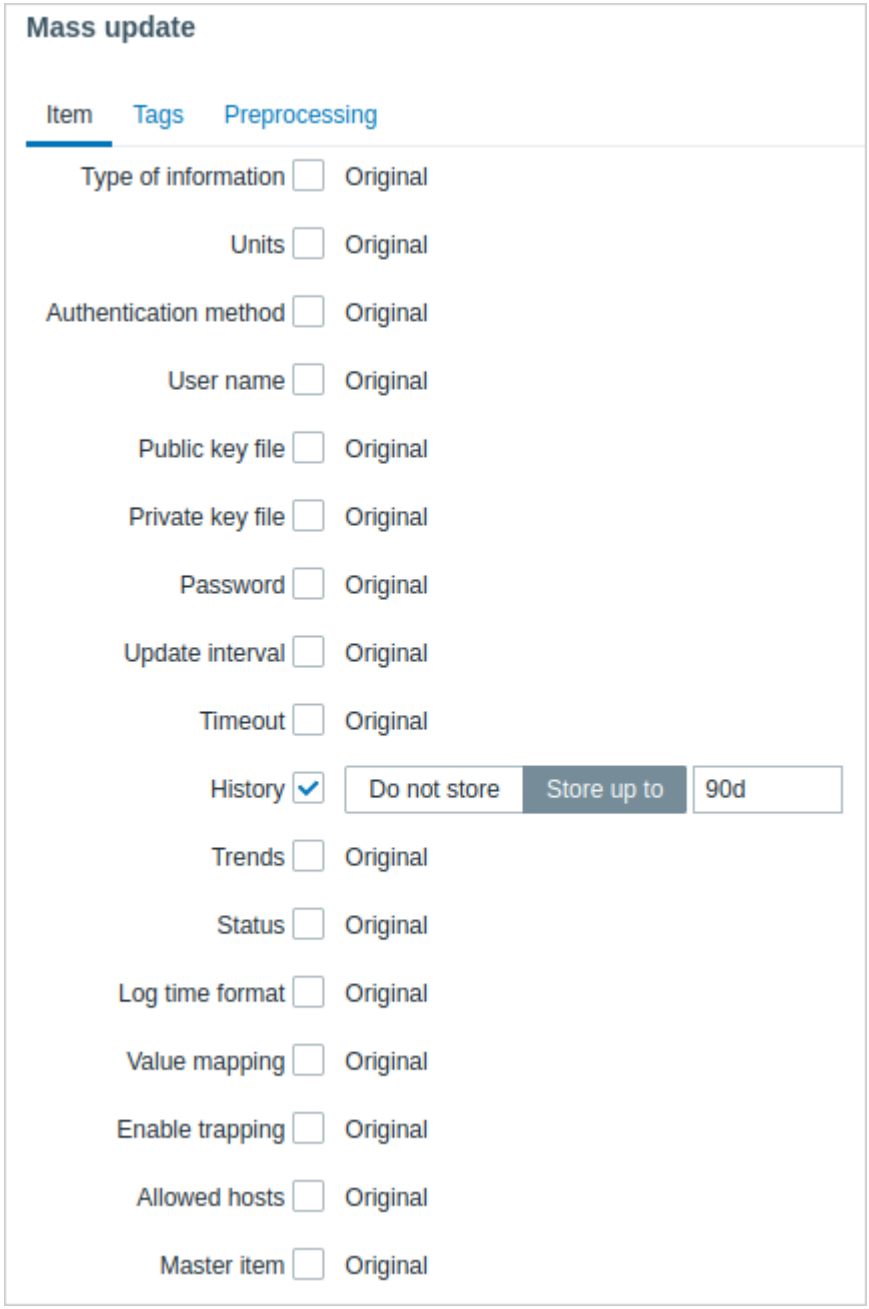
Обзор

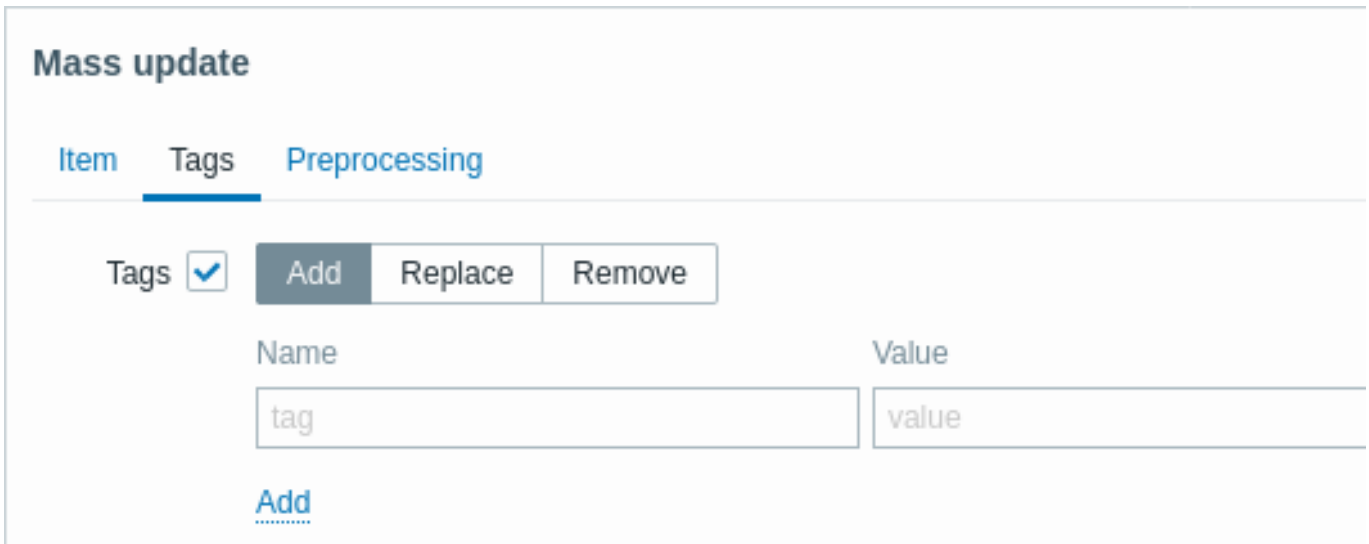
Иногда вы можете захотеть изменить какие-нибудь атрибуты у нескольких элементов данных одновременно. Вместо открытия каждого элемента данных для редактирования, вы можете воспользоваться функцией массового обновления.

Массовое обновление

Чтобы массово обновить некоторые элементы данных, выполните следующие действия:

- Отметьте флажками элементы данных, которые нужно обновить, в списке
- Нажмите *Mass update* под списком
- Перейдите на вкладку с требуемыми атрибутами (*Item*, *Tags* или *Preprocessing*)
- Отметьте флажками атрибуты, которые нужно обновить
- Введите новые значения для атрибутов



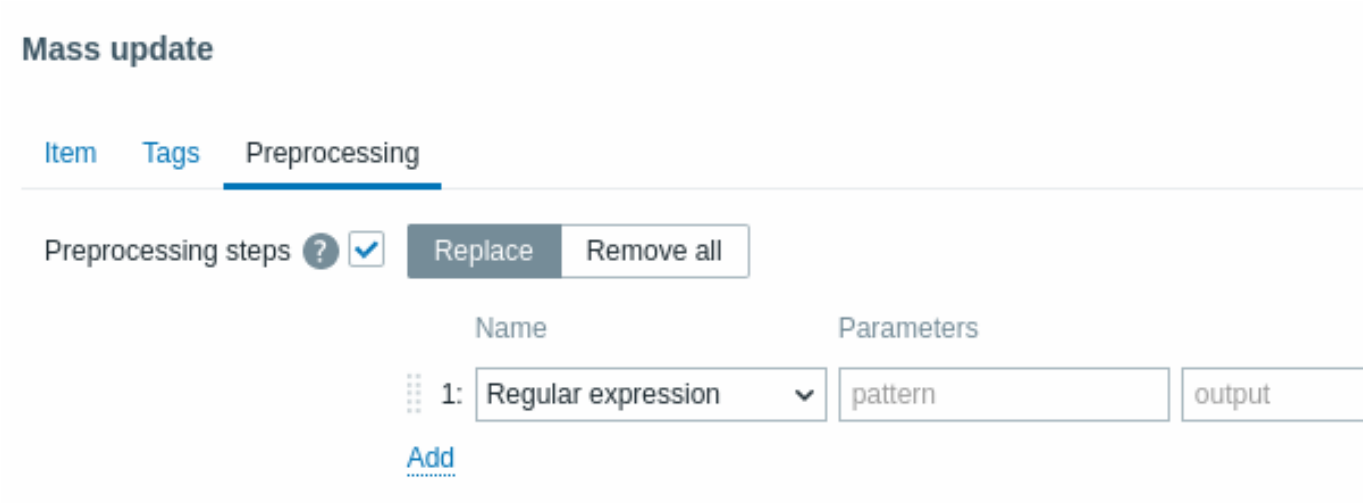


Параметр *Tags* позволяет:

- *Add* — добавить указанные теги к элементам данных (теги с одинаковым именем, но разными значениями не считаются «дубликатами» и могут быть добавлены к одному и тому же элементу данных).
- *Replace* — удалить указанные теги и добавить теги с новыми значениями

- *Remove* — 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 *Preprocessing steps*, чтобы активировать массовое обновление шагов предобработки.

Массовое обновление *Preprocessing* позволяет:

- *Replace* — заменить существующие шаги предобработки у элементов данных шагами предобработки, указанными ниже
- *Remove all* — удалить все шаги предобработки у элементов данных

После завершения нажмите *Update*.

8 Преобразование значений

Обзор

Сопоставление значений позволяет настроить более удобное для пользователя представление полученных значений с помощью сопоставлений между числовыми/строковыми значениями и их строковыми представлениями.

Например, когда значение элемента данных равно ”0” или ”1”, сопоставления значений можно использовать для более удобного представления этих значений:

- 0 → Недоступно
- 1 → Доступно

Сопоставления значений для типов резервного копирования данных могут быть настроены следующим образом:

- F → Полное
- D → Дифференциальное
- I → Инкрементальное

Диапазоны значений напряжения могут быть настроены следующим образом:

- <=209 → Низкое
- 210-230 → Норма
- >=231 → Высокое

Сопоставление значений используется в веб-интерфейсе Zabbix и в уведомлениях, отправляемых способами оповещения.

Attention:
Подстановка полученного значения настроенным представлением выполняется как в веб-интерфейсе Zabbix, так и на сервере; однако сервер выполняет подстановку только в следующих случаях:

- при заполнении инвентарных данных узла сети;
- при раскрытии поддерживаемых макросов {ITEM.VALUE}, {ITEM.LASTVALUE}, {EVENT.OPDATA} и {EVENT.CAUSE.OPDATA}.

Сопоставления значений настраиваются в шаблонах или на узлах сети. После настройки они доступны для всех элементов данных в соответствующем шаблоне или на узле сети. При настройке элементов данных укажите имя ранее настроенного сопоставления значений в параметре *Value mapping*.

Note:
Наследование сопоставлений значений отсутствует — узлы сети и шаблоны не наследуют сопоставления значений из связанных шаблонов. Элементы данных шаблона на узле сети продолжат использовать сопоставления значений, настроенные в шаблоне.

Note:
Сопоставления значений можно использовать с элементами данных, имеющими тип информации *Numeric (unsigned)*, *Numeric (float)* и *Character*.

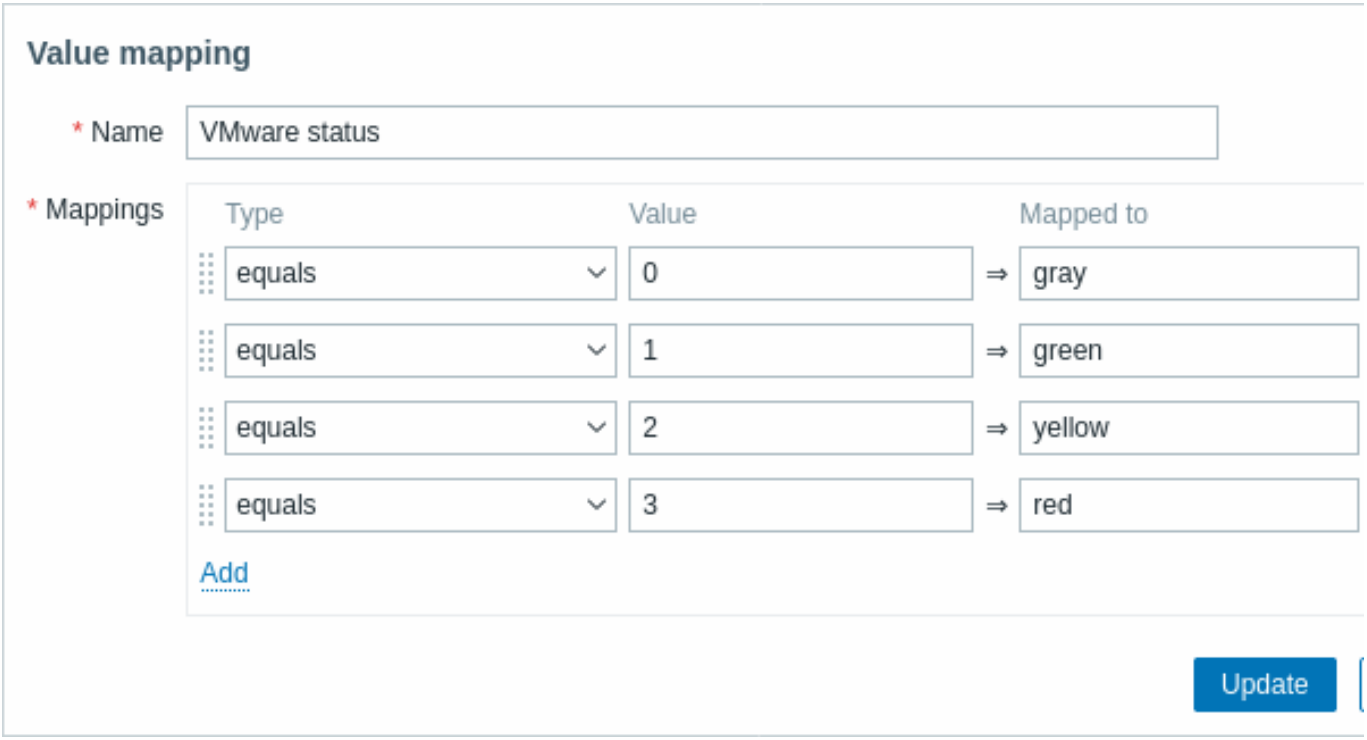
Сопоставления значений экспортируются/импортируются вместе с соответствующим шаблоном или узлом сети. Их также можно массово обновлять с помощью форм массового обновления узлов сети и шаблонов.

Настройка

Чтобы настроить сопоставление значений, выполните следующие шаги:

1. Откройте форму настройки узла сети или шаблона.

2. На вкладке *Value mapping* нажмите *Add*, чтобы добавить новое сопоставление значений, или нажмите на имя существующего сопоставления, чтобы изменить его.



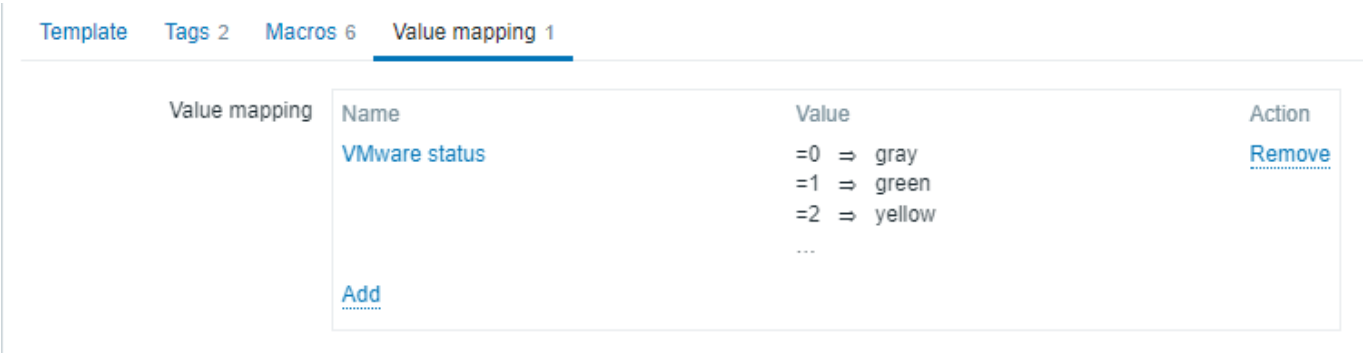
Параметры сопоставления значений:

Параметр	Описание
Name	Уникальное имя для набора сопоставлений значений.
Mappings	Отдельные правила для сопоставления числовых/строковых значений со строковыми представлениями. Сопоставление применяется в порядке правил, который можно изменять перетаскиванием.

Параметр	Описание
Type	Тип сопоставления: **equals** - будут сопоставлены равные значения; **is greater than or equals** - будут сопоставлены равные или большие значения; **is less than or equals** - будут сопоставлены равные или меньшие значения; **in range** - будут сопоставлены значения в диапазоне; диапазон задается как <number1>-<number2> или <number>; поддерживается несколько диапазонов (например, 1-10,101-110,201); **regexp** - будут сопоставлены значения, соответствующие регулярному выражению (глобальные регулярные выражения не поддерживаются); **default** - будут сопоставлены все остальные значения, кроме тех, для которых заданы конкретные сопоставления. Для сопоставления диапазонов поддерживаются только числовые типы значений (*is greater than or equals*, *is less than or equals*, *in range*).
Value	Входящее значение (может содержать диапазон или регулярное выражение, в зависимости от типа сопоставления).
Mapped to	Строковое представление (до 64 символов) для входящего значения.

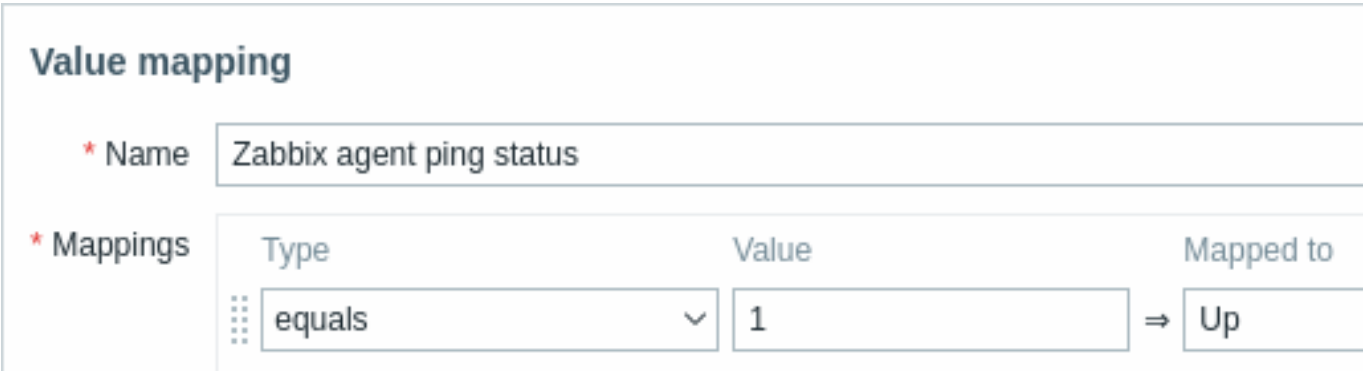
Все обязательные поля ввода отмечены красной звездочкой.

При просмотре сопоставления значений в списке видны только первые три сопоставления, а три точки указывают на наличие дополнительных сопоставлений.



Пример сопоставления значений

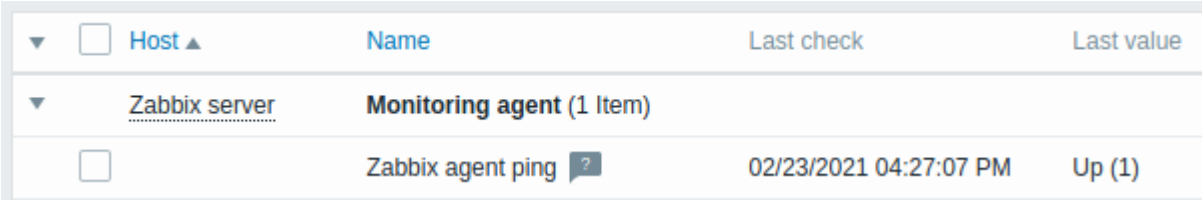
Один из предопределённых элементов данных агента *Zabbix agent ping* использует сопоставление значений уровня шаблона "Zabbix agent ping status" для отображения своих значений.



В форме настройки элемента данных вы можете найти ссылку на это сопоставление значений в поле *Value mapping*:



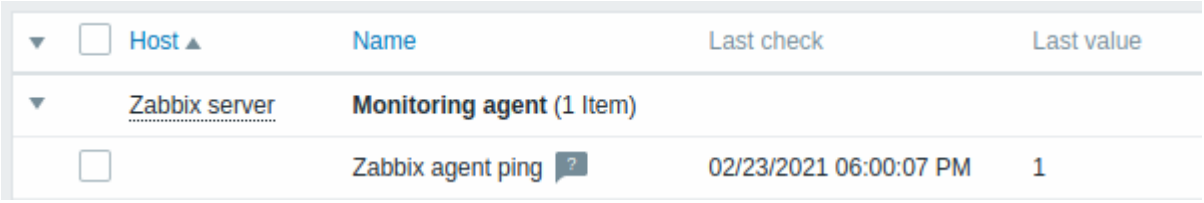
Это сопоставление используется в разделе *Мониторинг* → *Последние данные* для отображения "Up" (с исходным значением в скобках).



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, это сокращение применяется не к сопоставленному значению, а только к исходному значению (отображаемому в скобках).

Без предопределённого сопоставления значений вы бы увидели только ”1”, что могло бы быть затруднительно для понимания.



Host	Name	Last check	Last value
Zabbix server	**Monitoring agent** (1 Item)		
	Zabbix agent ping	02/23/2021 06:00:07 PM	1

9 Очередь

Обзор

Очередь отображает элементы данных, ожидающие обновления. Очередь является лишь **логическим** представлением данных. В Zabbix нет IPC-очереди или какого-либо другого механизма очередей.

Элементы данных, которые мониторятся прокси, также включаются в очередь — они будут учитываться как находящиеся в очереди на период обновления исторических данных прокси.

В очереди отображаются только элементы данных с запланированным временем обновления. Это означает, что следующие типы элементов данных исключаются из очереди:

- активные элементы данных агента Zabbix типов log, logrt и event log
- элементы данных SNMP trap
- элементы данных trapper
- элементы данных веб-мониторинга
- зависимые элементы данных

Статистика, отображаемая очередью, является хорошим индикатором производительности сервера Zabbix.

Очередь извлекается напрямую с сервера Zabbix с использованием протокола JSON. Информация доступна только в том случае, если сервер Zabbix запущен.

Attention:
Элементы данных не учитываются в очереди, если интерфейс элемента данных становится недоступным из-за проблем с подключением или некорректной работы агента.

Чтение очереди

Чтобы просмотреть очередь, перейдите в *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

Изображение здесь в целом «нормальное», поэтому можно предположить, что сервер работает нормально.

Очередь показывает некоторые элементы данных, ожидающие до 30 секунд. Было бы полезно узнать, что это за элементы данных.

Чтобы сделать именно это, выберите *Подробности очереди* в выпадающем списке заголовка. Теперь вы можете увидеть 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

Имея эти подробности, можно попытаться выяснить, почему эти элементы данных могут задерживаться.

Если задерживаются один или два элемента данных, возможно, причин для беспокойства нет. Они могут обновиться через секунду. Однако если вы видите, что множество элементов данных задерживаются слишком долго, это может указывать на более серьёзную проблему.

См. также: Выравнивание 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

Можно использовать специальный внутренний элемент данных **zabbix[queue,<от>,<до>]** для наблюдения за состоянием очереди Zabbix. Он будет возвращать количество элементов данных, задерживающихся за указанный период времени. Для получения более подробной информации смотрите Внутренние элементы данных.

10 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, и кэш обновляется соответственно.

Для включения функционала кэша значений поддерживается дополнительный параметр **ValueCacheSize** в файле конфигурации Zabbix сервера.

Значения элементов данных остаются в кэше значений до тех пор, пока:

- элемент данных удален (кешированные значения удаляются после следующей синхронизации конфигурации);
- значение элемента находится за пределами диапазона времени или количества, указанного в выражении триггера/вычисляемого элемента (кэшированное значение удаляется при получении нового значения);
- изменен диапазон времени или счетчика, указанный в выражении триггера/вычисляемого элемента, таким образом, чтобы для расчета требовалось меньше данных (ненужные кэшированные значения удаляются через 24 часа).

Note:
Статус кэша значений можно наблюдать с помощью опции сервера runtime controldiaginfo (или `diaginfo=valuecache`) и проверкой раздела на наличие диагностической информации о кэше значений. Это может быть полезно для определения неправильно настроенных триггеров или вычисляемых элементов.

Чтобы включить функцию кэширования значений, необходимо использовать необязательный параметр **ValueCacheSize**. параметр поддерживается сервером Zabbix файл конфигурации.

Для мониторинга кэша значений поддерживаются два внутренних элемента: **zabbix[vcache,buffer,<режим>]** и **zabbix[vcache,cache,<параметр>]**. Смотрите подробности в описании внутренних элементов данных.

11 Выполнить сейчас

Обзор

Опция *Выполнить сейчас* позволяет немедленно выполнять пассивные проверки.

Сбор значений элементов данных в Zabbix выполняется в соответствии с настроенными интервалами обновления. Некоторые элементы данных (например, правила низкоуровневого обнаружения) имеют большие интервалы, и вам может потребоваться немедленно получить новое значение — например, чтобы обнаружить изменения в обнаруживаемых ресурсах.

Опция *Выполнить сейчас* поддерживается для следующих типов элементов данных:

- Zabbix агент (пассивный)
- Простая проверка
- SNMP агент (v1/v2/v3)
- Внутренний элемент Zabbix
- Внешняя проверка
- Монитор базы данных
- HTTP агент
- IPMI агент
- SSH агент
- TELNET агент
- JMX агент
- Вычисляемый
- Зависимый элемент данных
- Скрипт
- Браузер

Если элемент данных имеет тип ”Зависимый элемент данных”, то его мастер-элемент также должен быть одним из перечисленных выше типов.

Конфигурация

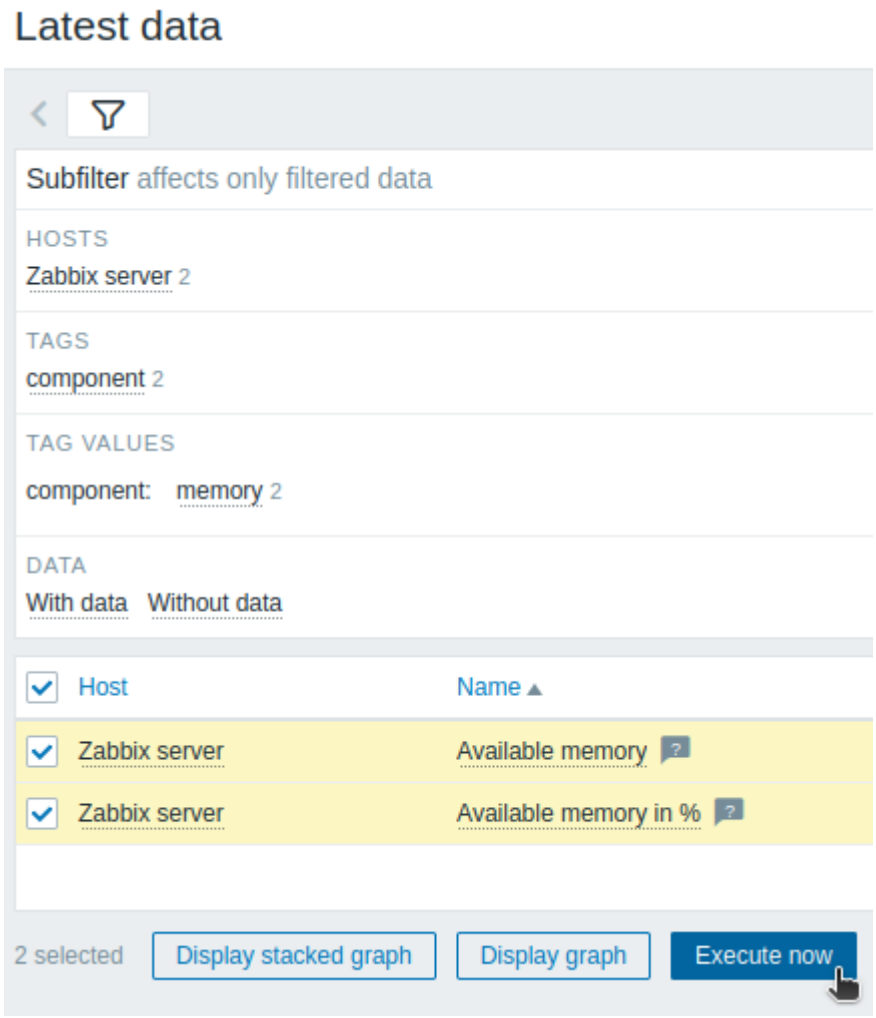
Вы можете выполнять пассивные проверки из:

- *Мониторинг* > *Последние данные*
- *Сбор данных* > *Узлы сети* > *Элементы данных* или *Обнаружение*
- Меню элемента данных
- При редактировании элементов данных или правил обнаружения низкого уровня (в их формах конфигурации)

Attention:
Проверка должна существовать в кэше конфигурации, чтобы быть выполненной; см. также CacheUpdateFrequency. Кэш **не обновляется** перед выполнением проверки, поэтому недавние изменения в конфигурации элемента данных или правила обнаружения низкого уровня не будут учтены. Для тестирования создаваемых или недавно созданных элементов данных или правил LLD вместо этого можно использовать опцию Test.

Например, чтобы выполнить пассивную проверку в *Мониторинг* > *Последние данные*:

1. Выберите элементы данных; вы можете выполнить проверку сразу для нескольких элементов данных.
2. Нажмите *Выполнить сейчас*.



Note:
В разделе *Последние данные* пользователи могут *Выполнить сейчас* элементы данных на узлах сети, для которых у них есть разрешение *чтение-запись* или разрешение *чтение* и для их роли включено действие *Invoke "Execute now" on read-only hosts*.

12 Ограничение проверок агента

Обзор

Вы можете управлять тем, какие ключи элементов данных Zabbix агент или агент 2 может использовать или не использовать при выполнении проверок элементов данных, удалённых команд или скриптов.

Для этого используйте следующие параметры конфигурации агента для определения правил разрешения/запрета:

- `AllowKey=<pattern>`

- DenyKey=<pattern>

<pattern> должен содержать один ключ элемента данных и 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.*[*]).

Attention:
Для повышения безопасности рекомендуется использовать точные ключи элементов данных вместо подстановочных знаков. Подробнее см. в разделе Защита правил разрешения/запрета.

В отличие от других параметров конфигурации агента, вы можете указать неограниченное количество параметров AllowKey или DenyKey.

Важные примечания

- Все элементы данных system.run по умолчанию отключены (даже если DenyKey пуст), как если бы DenyKey=system.run[*] был задан как последнее правило. Благодаря этому вы можете разрешить определённые элементы данных system.run, не запрещая явно другие элементы данных system.run.
- Элемент данных, указанный в AllowKey, также должен быть указан в DenyKey (кроме элементов данных system.run); в противном случае агент Zabbix **не запустится**.
- По возможности используйте AllowKey, чтобы разрешить только необходимые элементы данных и запретить всё остальное. Некоторые ключи могут использоваться для чтения непредусмотренных файлов через обход пути (например, vfs.file.contents["../../../../etc/passwd"]), а новые версии агента Zabbix могут вводить ключи, не охваченные вашими правилами DenyKey.
- Конфигурация AllowKey и DenyKey не влияет на параметры агента HostnameItem, HostMetadataItem или HostInterfaceItem.
- Запрещённые элементы данных становятся неподдерживаемыми без каких-либо подсказок или сообщений об ошибках; например:
  - Параметр командной строки агента Zabbix --print (-p) не будет показывать ключи запрещённых элементов данных.
  - Параметр командной строки агента Zabbix --test (-t) будет возвращать "Unsupported item key." для ключей запрещённых элементов данных.
  - Когда ведение журнала включено (LogRemoteCommands=1), файл журнала агента Zabbix не будет регистрировать запрещённые удалённые команды.

Порядок правил Allow/deny

Вы можете указать неограниченное количество правил AllowKey или DenyKey, 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 останавливается.

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 распространённые шаблоны конфигурации для AllowKey и DenyKey.

Разрешение определённых проверок и команд

Разрешите только две проверки элементов данных vfs.file и две команды system.run:

```
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.*[*]
```

Note:
Указывать `DenyKey=system.run[*]` не требуется, поскольку все остальные команды `system.run` по умолчанию запрещены.

Разрешение скриптов

Разрешите агенту Zabbix выполнять скрипты на узлах сети всеми доступными способами:

- Глобальные скрипты, которые можно выполнять в веб-интерфейсе или через API (этот метод всегда использует ключ `system.run[myscript.sh])`
- Удаленные команды из операций действий (этот метод всегда использует ключ `system.run[myscript.sh,nowait])`
- Элементы данных агента Zabbix `system.run` со скриптом, например:
  - `system.run[myscript.sh]`
  - `system.run[myscript.sh,wait]`
  - `system.run[myscript.sh,nowait]`

```
AllowKey=system.run[myscript.sh,*]
```

Чтобы управлять параметром `wait/nowait`, необходимо задать другое правило. Например, можно разрешить только элементы данных `system.run[myscript.sh,wait]`, исключив таким образом другие методы:

```
AllowKey=system.run[myscript.sh,wait]
```

Защита правил allow/deny

В этом примере показано, как защитить чрезмерно разрешающие правила `AllowKey` или `DenyKey`.

Рассмотрим следующие правила:

```
AllowKey=system.run["C:\Program^ Files\Zabbix^ Agent^ 2\scripts\test.bat*"]
DenyKey=vfs.file.*
DenyKey=system.cpu.load[*]
```

Note:
В Windows необходимо экранировать пробелы в пути с помощью символа каретки (^).

Эти правила содержат подстановочный знак (*), которым можно злоупотребить:

- Скрипт `test.bat` можно выполнить с любыми аргументами, включая непредусмотренные.
- Шаблон `vfs.file.*` соответствует только ключам элементов данных без параметров; однако все элементы данных `vfs.file` требуют параметры.
- Шаблон `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 определёнными аргументами и запретите корректные шаблоны ключей элементов данных; например:

```
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[*]
```

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

В следующей таблице показано, как сопоставляются шаблоны ключей элементов данных:

- Ключ соответствует шаблону только в том случае, если он удовлетворяет **всем** условиям в столбце *Соответствует*.
- Параметры должны быть полностью заключены в квадратные скобки (например, `vfs.file.contents[*` и `vfs.file.contents*utf8]` — недопустимые шаблоны).

Шаблон	Соответствует	Примеры
*	Любой ключ с параметрами или без них	
vfs.file.*	Ключ начинается с vfs.file. Без параметров	Соответствует: vfs.file.size vfs.file.contents Не соответствует: vfs.file.contents[] vfs.file.size[/var/log/app.log]
vfs.*.contents	Ключ начинается с vfs. Ключ заканчивается на .contents Без параметров	Соответствует: vfs..contents vfs.mount.point.file.contents Не соответствует: vfs.contents vfs.file.contents[]
vfs.file.*[*]	Ключ начинается с vfs.file. Любые или пустые параметры	Соответствует vfs.file.get.custom[] vfs.file.size[/var/log/app.log, utf8] Не соответствует: vfs.file.get.custom
vfs.file.contents	Ключ — vfs.file.contents Без параметров	Соответствует: vfs.file.contents Не соответствует: vfs.file.contents[/etc/passwd]
vfs.file.contents[]	Ключ — vfs.file.contents[] Пустые параметры	Соответствует: vfs.file.contents[] Не соответствует: vfs.file.contents
vfs.file.contents[*]	Ключ — vfs.file.contents Любые или пустые параметры	Соответствует: vfs.file.contents[/path/to/file] Не соответствует: vfs.file.contents
vfs.file.contents[/etc/passwd,*]	Ключ — vfs.file.contents Первый параметр — /etc/passwd Любой или пустой второй параметр	Соответствует: vfs.file.contents[/etc/passwd,] vfs.file.contents[/etc/passwd,utf8] Не соответствует: vfs.file.contents[] vfs.file.contents[/etc/passwd]
vfs.file.contents[*passwd*]	Ключ — vfs.file.contents Первый параметр содержит passwd Без второго параметра	Соответствует: vfs.file.contents[/etc/passwd] Не соответствует: vfs.file.contents[/etc/passwd,] vfs.file.contents[/etc/passwd,utf8]
vfs.file.contents[*passwd*,*]	Ключ — vfs.file.contents Первый параметр содержит passwd Любой или пустой второй параметр	Соответствует: vfs.file.contents[/etc/passwd,] vfs.file.contents[/etc/passwd,utf8] Не соответствует: vfs.file.contents[/etc/passwd] vfs.file.contents[/tmp/test]

Шаблон	Соответствует	Примеры
vfs.file.contents[/etc/passwd,utf8]	Контент файла. Первый параметр — /etc/passwd Второй параметр — utf8	Соответствует: vfs.file.contents[/etc/passwd,utf8] Не соответствует: vfs.file.contents[/etc/passwd,] vfs.file.contents[/etc/passwd,utf16]

3 Обнаружение проблем с помощью триггеров

Обнаружение проблем в Zabbix основано на **триггерах**.

Триггер описывает условие проблемы, позволяя Zabbix реагировать, если проблема возникает.

Полный 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 просто собирает данные, но никогда на них не реагирует.
3. Наблюдайте за обнаруженными проблемами:
    - В *Мониторинг* > *Проблемы*
    - В виджете *Проблемы*
4. Получайте уведомления, выполняйте удаленные команды (если они определены).

Обратите внимание, что официальные шаблоны Zabbix содержат предопределенные триггеры. См. руководство по применению шаблона для мониторинга 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 является частью выражения.
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 (процессом history syncer), если выражение содержит одно из следующего:

- функции даты и времени
- функцию nodata()

Триггеры, содержащие только trend функции, вычисляются один раз за наименьший период времени в выражении. Хотя многие функции триггеров используют данные истории значений элемента данных, trend-функции используют данные трендов.

1 Примеры триггеров

На этой странице собрана коллекция примеров триггеров.

Примеры отсортированы по вариантам использования:

- Недоступный узел сети
- Недоступный прокси
- Недоступный SMTP-кластер
- Недостижимый узел сети
- Неожиданная перезагрузка
- Изменения в важных файлах
- Изменения в DNS-запросе
- Несоответствие программного обеспечения на разных узлах сети
- Несинхронизированные часы
- Устаревший агент
- Нестабильный интерфейс
- Высокий входящий трафик
- Мало места на диске
- Мало места на диске (динамический порог)
- Высокая загрузка CPU
- Сравнение загрузки CPU
- Долгосрочное сравнение загрузки CPU
- 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[192.0.2.0,{$WEBSITE_NAME},{$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[192.0.2.0,{$WEBSITE_NAME},{$DNS_RESOURCE_RECORD_TYPE},2,1], с определёнными макросами:

```
{$WEBSITE_NAME} = example.com
{$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`

2 Настройка триггера

На этой странице описано, как настроить триггер в веб-интерфейсе Zabbix.

Обратите внимание, что официальные шаблоны Zabbix содержат предопределенные триггеры. См. руководство по применению шаблона для мониторинга Linux. Также можно массово обновить настроенные триггеры.

Другие ресурсы:

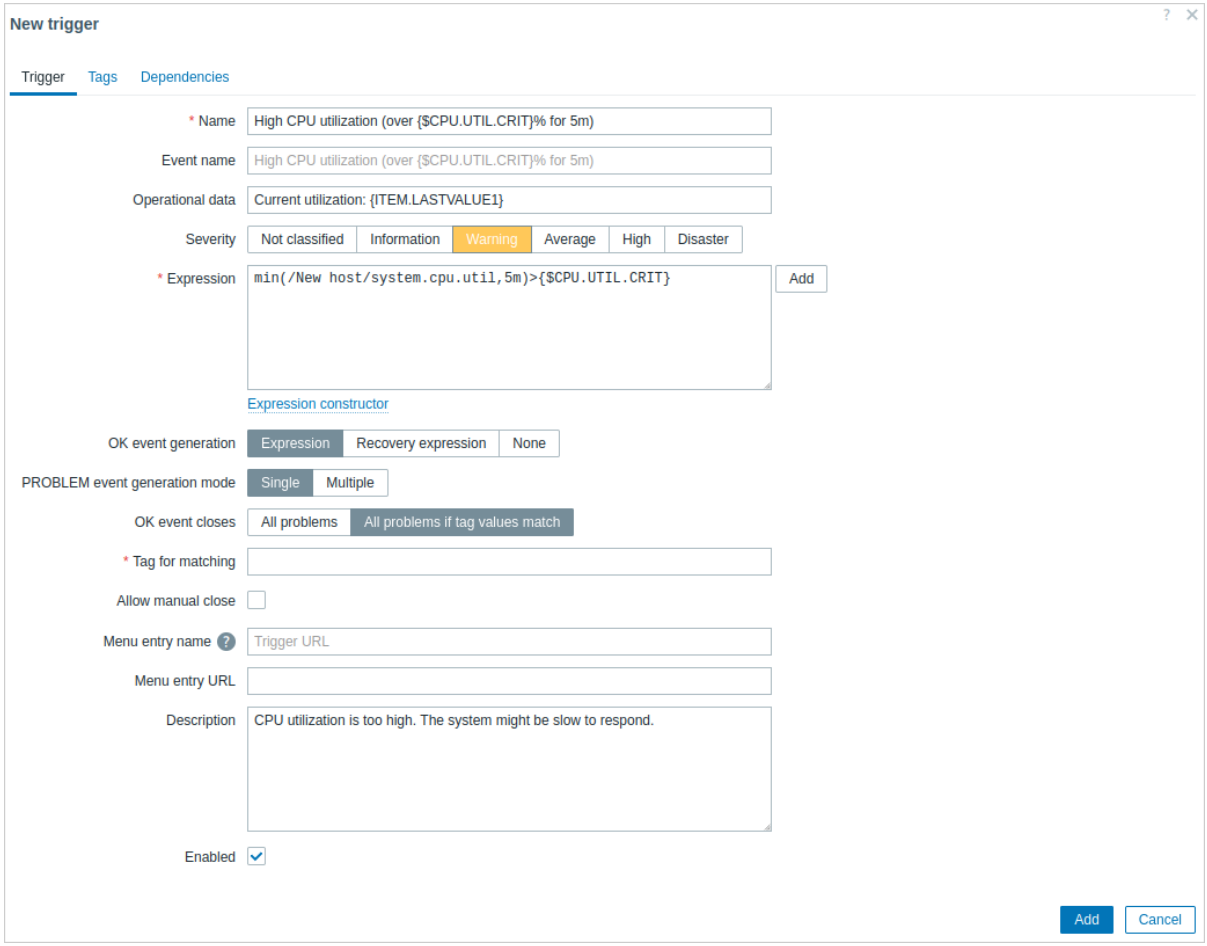
- Обнаружение проблем с помощью триггеров (общее введение)
- Примеры триггеров (по вариантам использования)
- Выражение триггера (подробности синтаксиса)
- Рекомендации

Конфигурация

Чтобы настроить триггер, выполните следующие действия:

- Перейдите в: *Сбор данных* > *Узлы сети*
- Нажмите *Триггеры* в строке узла сети
- Нажмите *Создать триггер* справа (или на имя триггера, чтобы редактировать существующий триггер)
- Введите параметры триггера в форме

Вкладка **Триггер** содержит все основные атрибуты триггера.

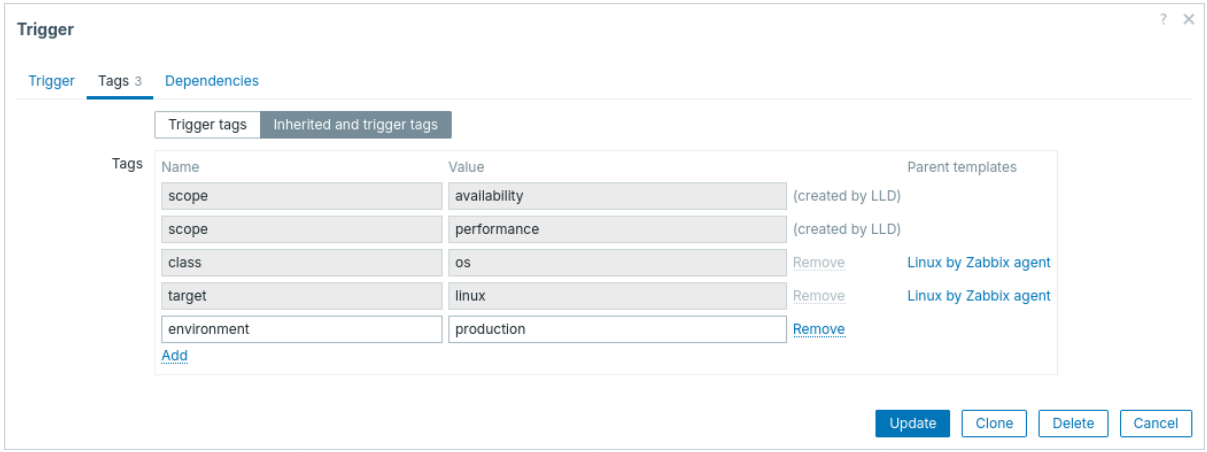


Все обязательные поля ввода отмечены красной звездочкой.

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 при ссылке на константы в относительно простых, прямолинейных выражениях. Например, имя "Загрузка процессора выше $1 на {HOST.NAME}" автоматически изменится на "Загрузка процессора выше 5 на New host", если выражение имеет вид last(/New host/system.cpu.load[percpu,avg1])>5.
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 построения информативных оповещений, содержащих данные о проблеме (см. пример). Поддерживается тот же набор макросов, что и в имени триггера, а также макросы выражений {TIME}, {TIMESTAMP} и {?EXPRESSION}.
Operational data	Оперативные данные позволяют задавать произвольные строки вместе с макросами. Макросы будут динамически разрешаться в данные реального времени в *Мониторинг* > *Проблемы*. В то время как макросы в имени триггера (см. выше) разрешаются в свои значения в момент возникновения проблемы и становятся основой статического имени проблемы, макросы в оперативных данных сохраняют возможность динамически отображать самую актуальную информацию. Если оперативные данные не настроены на уровне триггера, будут отображаться последние значения всех элементов данных из выражения. Поддерживается тот же набор макросов, что и в имени триггера.
Severity	Установите требуемую важность триггера, нажимая кнопки.
Expression	Логическое выражение, используемое для определения условий 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 устранена, как только выражение примет значение FALSE, если только в *Выражение восстановления* не заданы дополнительные условия восстановления.
OK event generation	Параметры генерации OK-событий: **Expression** - OK-события генерируются на основе того же выражения, что и события проблемы; **Recovery expression** - OK-события генерируются, если выражение проблемы принимает значение FALSE, а выражение восстановления принимает значение 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 устранения проблемы, после того как исходное выражение проблемы уже приняло значение FALSE. Выражение восстановления полезно для гистерезиса триггера. **Невозможно** устранить проблему только выражением восстановления, если выражение проблемы все еще имеет значение TRUE. Это поле доступно только если для *OK event generation* выбрано значение 'Recovery expression'.
PROBLEM event generation mode	Режим генерации событий проблемы: **Single** - одно событие генерируется, когда триггер впервые переходит в состояние 'Проблема'; **Multiple** - событие генерируется при *каждой* оценке триггера как 'Проблема'.
OK event closes	Выберите, что закрывает OK-событие: **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 значение 'All problems if tag values match', и в этом случае является обязательным.
Allow manual close	Установите флажок, чтобы разрешить закрытие вручную событий проблемы, созданных этим триггером. Закрытие вручную возможно при подтверждении событий проблемы.

Parameter	Description
Menu entry name	Если не пусто, введенное здесь имя (до 64 символов) используется в нескольких местах веб-интерфейса как метка для URL триггера, указанного в параметре *Menu entry URL*. Если пусто, используется имя по умолчанию *Trigger URL*. Поддерживается тот же набор макросов, что и в имени триггера, а также {EVENT.ID}, {HOST.ID} и {TRIGGER.ID}.
Menu entry URL	Если не пусто, введенный здесь URL (до 2048 символов) доступен как ссылка в меню события в нескольких местах веб-интерфейса, например, при нажатии на имя проблемы в *Мониторинг* > *Проблемы* или в виджете панели мониторинга *Проблемы*. Поддерживается тот же набор макросов, что и в имени триггера, а также {EVENT.ID}, {HOST.ID} и {TRIGGER.ID}. Примечание: пользовательские макросы с секретными значениями не будут разрешаться в URL.
Description	Текстовое поле, используемое для предоставления дополнительной информации об этом триггере. Может содержать инструкции по устранению конкретной проблемы, контактные данные ответственного персонала и т. д. Поддерживается тот же набор макросов, что и в имени триггера.
Enabled	Снятие этого флажка отключит триггер при необходимости. Проблемы отключенного триггера больше не отображаются в веб-интерфейсе, но не удаляются.

Вкладка **Теги** позволяет определить теги на уровне триггера. Все проблемы этого триггера будут помечены введенными здесь значениями.



Кроме того, опция *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 маркировки событий триггера. Теги представляют собой пару из имени тега и значения. Можно использовать только имя или пару имя-значение. 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 функции макросов с {{ITEM.VALUE}}, {{ITEM.LASTVALUE}}. Макросы низкоуровневого обнаружения можно использовать внутри контекста макроса. В значениях тегов триггера поддерживается макрос {TRIGGER.ID}. 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 символов, оно будет обрезано до 255 символов. См. все макросы, поддерживаемые для тегов событий. Теги событий могут использоваться для корреляции событий, в условиях действий, а также отображаются в *Мониторинг* > *Проблемы* или в виджете *Проблемы*.

Вкладка **Зависимости** содержит все зависимости триггера.

Нажмите *Добавить*, чтобы добавить новую зависимость.

> **Note:**
> Вы также можете настроить триггер, открыв существующий, нажав кнопку *Клонировать*, а затем сохранив его под другим именем.

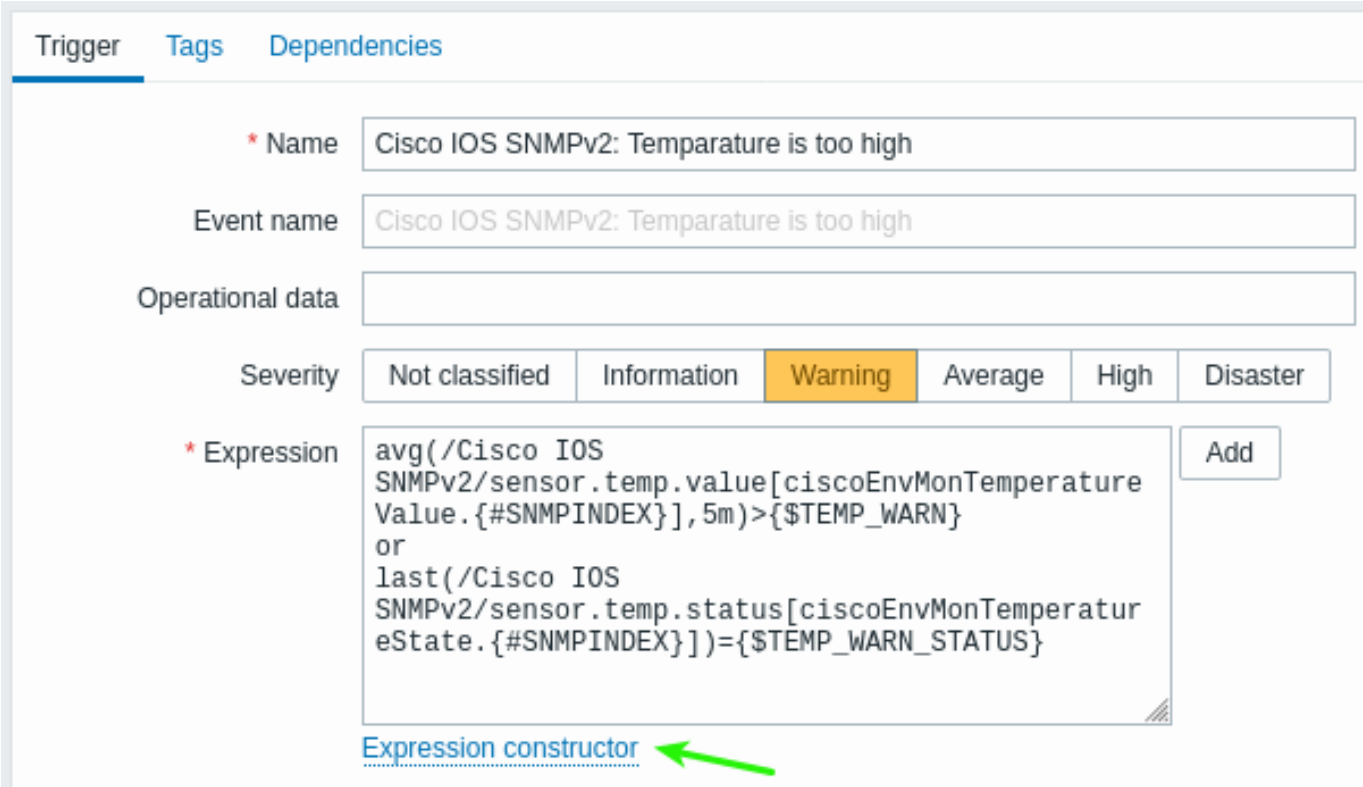
Тестирование выражения триггера

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}
```

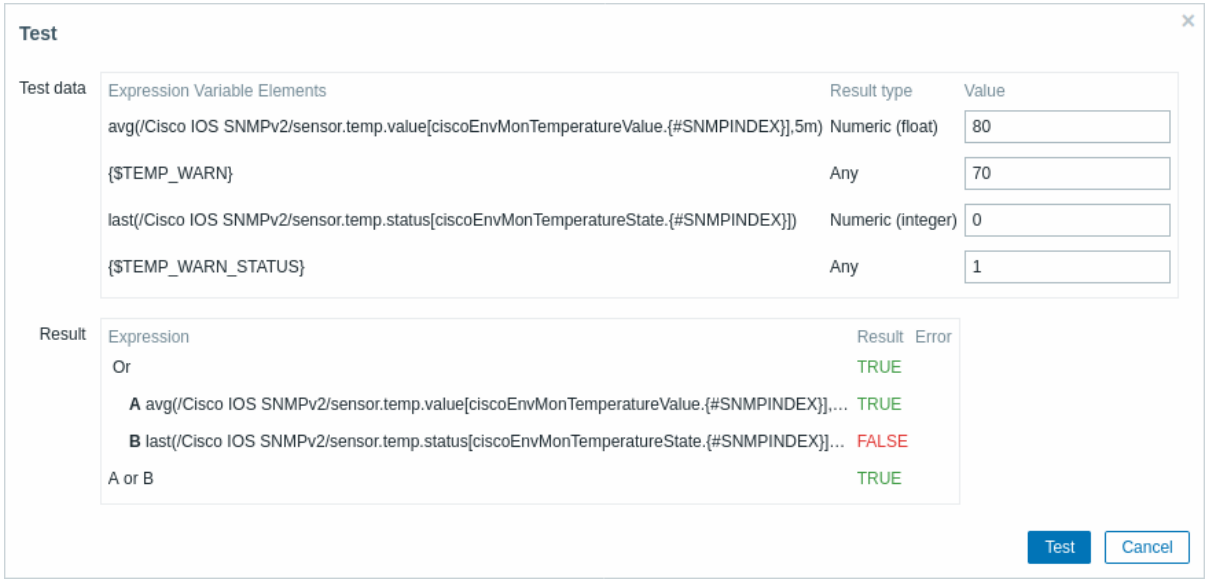
Чтобы протестировать выражение, нажмите *Конструктор выражений* под полем выражения.



В конструкторе выражений 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 увидеть результат выражения, нажав кнопку *Тест*.



Можно увидеть результат отдельных выражений, а также всего выражения целиком.

”TRUE” означает, что указанное выражение истинно. 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 ложно. В данном случае B значение {$TEMP_WARN_STATUS} ”1” должно быть равно указанному значению, которое в этом примере равно ”0”. Как и ожидалось, отображается результат ”FALSE”.

Выбранный тип выражения — ”OR”. Если хотя бы одно из указанных условий (в данном случае A или B) истинно, общий результат также будет TRUE. Это означает, что текущее значение превышает предупреждающее значение и возникла проблема.

3 Выражение триггера

Обзор

На этой странице описываются синтаксис выражений триггеров и подробности их вычисления.

Синтаксис простого выражения:

`function(/host/item,time_period)<operator><constant>`

В этом выражении первый операнд (слева от оператора) — это функция с её параметрами в круглых скобках (обычно это элемент данных и период времени).

Функция используется для анализа полученных данных за указанный период времени, в результате чего вычисляется значение.

Затем это значение сравнивается со вторым операндом с помощью оператора. В этом примере второй операнд является константой, но он также может быть другой функцией.

Например:

`min(/Zabbix server/net.if.in[eth0,bytes],5m)>100K`

Этот триггер срабатывает, если количество полученных байтов на `eth0` в течение последних пяти минут всегда было больше 100 килобайт. В этом случае выражение истинно и создаётся 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 или находить определенную строку.

Нажмите на соответствующую группу функций, чтобы увидеть более подробную информацию.

Группа функций	Функции
Агрегирующие 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, last, lastclock, logeventid, logseverity, logsource, logtimestamp, monodec, monoinc, nodata, percentile, rate
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
Операторные функции	between, in
Прогнозирующие функции	forecast, timeleft
Строковые функции	ascii, bitlength, bytelength, char, concat, insert, jsonpath, left, length, ltrim, mid, repeat, replace, right, rtrim, trim, xmlxpath

Если не указано иное, эти функции поддерживаются в:

- Выражениях триггеров
- Вычисляемых элементах данных
- Макросах выражений

Функции foreach поддерживаются только для агрегирующих вычислений.

Обычно функции возвращают числовые значения для сравнения. Если возвращаются строки, сравнение возможно с помощью операторов **=** и **<>** (см. пример Обнаружение несовпадающего программного обеспечения на разных узлах сети).

Параметры функции

Параметры функции позволяют указать:

- ключ элемента данных (в виде / /) для функций, ссылающихся на историю элемента данных узла сети
- период времени (и другие специфичные для функции параметры)
- 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, специфичные для функции, размещаются после ключа элемента данных и отделяются от ключа элемента данных запятой.

Большинство числовых функций принимают **период времени** в качестве параметра. Он позволяет указать интересующий нас интервал. Его можно указать как период времени (30s, 10m, 1h) или как диапазон значений (#5 — для пяти последних значений).

Для указания периода времени можно использовать секунды или суффиксы времени. Если параметру предшествует знак решетки, он имеет другое значение:

Expression	Description
sum(/host/key,**10m)**	Сумма значений за последние 10 минут.
sum(/host/key,**#10)**	Сумма десяти последних значений.

Параметры со знаком решетки имеют другое значение для функции **last** — они обозначают N-е предыдущее значение, поэтому при значениях 30, 70, 20, 60, 50 (от самого нового к самому старому):

- `last(/host/key,#2)` вернет '70'
- `last(/host/key,#5)` вернет '50'

Период времени измеряется до "текущего момента" — где "текущий момент" является временем последнего пересчета триггера (см. Частота вычисления); "текущий момент" не является временем "сейчас" сервера.

Период времени определяет одно из следующего:

- Учитывать все значения между "текущий момент - период времени" и "текущий момент" (или, при сдвиге времени, между "текущий момент - сдвиг времени - период времени" и "текущий момент - сдвиг времени")
- Учитывать не более num значений из прошлого, вплоть до "текущего момента"
  - Если для указанного периода времени или количества num доступно 0 значений, то 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 является временем последнего полученного значения. Например, если последнее значение было получено час назад, период времени будет считаться до последнего значения, полученного час назад.
- Новый триггер вычисляется сразу после получения первого значения (функции истории); для функций дата и время и nodata() он будет вычислен в течение 30 секунд. Таким образом, триггер будет вычислен, даже если, возможно, заданный период времени (например, один час) еще не прошел с момента создания триггера. Триггер также будет вычислен после первого значения, даже если диапазон был задан, например, как десять последних значений.

Сдвиг по времени

Поддерживается необязательный сдвиг по времени с указанием времени или количества значений в качестве параметра функции. Этот параметр позволяет ссылаться на данные за период времени в прошлом.

Сдвиг по времени начинается с `now`, обозначающего текущее время, после чего указывается `+N<time unit>` или `-N<time unit>` — для добавления или вычитания N единиц времени.

Например, `avg(/host/key,1h:now-1d)` вернет среднее значение за один час сутки назад.

Attention:
Сдвиг по времени, заданный 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).

Сдвиг по времени с абсолютными периодами времени

В параметре сдвига по времени поддерживаются абсолютные периоды времени, например, от полуночи до полуночи для дня, с понедельника по воскресенье для недели, с первого по последний день месяца для месяца.

Сдвиг по времени для абсолютных периодов времени начинается с `now`, обозначающего текущее время, после чего может следовать любое количество операций со временем: `/<time unit>` — задает начало и конец единицы времени, например, от полуночи до полуночи для дня; `+N<time unit>` или `-N<time unit>` — для добавления или вычитания N единиц времени.

Обратите внимание, что значение сдвига по 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	Текущая 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 нельзя использовать, если функция ссылается на историю элемента данных. Например, следующий синтаксис **не** допускается:

~~`min(/host/key,#5*10)`~~

Хотя другие выражения триггеров в качестве параметров функций в триггерах ограничены функциями, не использующими историю, это ограничение не применяется в вычисляемых элементах данных.

Операторы

Для триггеров поддерживаются следующие операторы **(в порядке убывания приоритета выполнения)**:

Приоритет	Оператор	Определение	**Примечания для неизвестных значений**	Принудительно приводить 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 потерять Unknown в арифметических операциях) 1.2 * Unknown → Unknown	Да
	/	Деление	Unknown / 0 → error Unknown / 1.2 → Unknown 0.0 / Unknown → Unknown	Да
4	+	Арифметический плюс	1.2 + Unknown → Unknown	Да
	-	Арифметический минус	1.2 - Unknown → Unknown	Да
5	<	Меньше чем. 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 чем. Оператор определяется как: A>B ⇔ (A>B+0.000001)		Да

Приоритет	Оператор	Определение	**Примечания для неизвестных значений**	Принудительно приводить 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 всё равно приводится к числовому, если:

- другой операнд является числовым
- к операнду применяется оператор, отличный от **=** или **<>**

(Если приведение не удаётся, числовой операнд приводится к строковому, и затем оба операнда сравниваются как строки.)

Операторы **not**, **and** и **or** чувствительны к регистру и должны быть в нижнем регистре. Они также должны быть отделены пробелами или круглыми скобками.

Все операторы, кроме унарных **-** и **not**, имеют левостороннюю ассоциативность. Унарные **-** и **not** неассоциативны (то есть следует использовать **-(-1)** и **not (not 1)** вместо **--1** и **not not 1**).

Результат вычисления:

- операторы **<**, **<=**, **>**, **>=**, **=**, **<>** должны возвращать ’1’ в выражении триггера, если указанное отношение истинно, и ’0’, если оно ложно. Если хотя бы один операнд имеет значение Unknown, результатом будет Unknown;
- **and** для известных операндов должно возвращать ’1’, если оба его операнда не равны ’0’; в противном случае возвращается ’0’; для неизвестных операндов **and** возвращает ’0’ только если один из операндов равен ’0’; в противном случае возвращается ’Unknown’;
- **or** для известных операндов должно возвращать ’1’, если хотя бы один из его операндов не равен ’0’; в противном случае возвращается ’0’; для неизвестных операндов **or** возвращает ’1’ только если один из операндов не равен ’0’; в противном случае возвращается ’Unknown’;
- результат оператора логического отрицания **not** для известного операнда равен ’0’, если значение его операнда не равно ’0’; и ’1’, если значение его операнда равно ’0’. Для неизвестного операнда **not** возвращает ’Unknown’.

Неизвестное состояние выражения

Неизвестный операнд может появиться в выражении триггера, если:

- используется неподдерживаемый элемент данных
- вычисление функции для поддерживаемого элемента данных приводит к ошибке

В этом случае выражение триггера обычно вычисляется в Unknown (так как его невозможно вычислить)

Можно получать уведомления о триггерах в состоянии unknown.

Исключения

Несмотря на наличие неизвестного операнда, в некоторых случаях выражения триггеров могут вычисляться в известный результат (Проблема/OK):

- Функция nodata() вычисляется независимо от того, поддерживается ли указанный элемент данных.
- Выражения с AND/OR могут вычисляться 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 "Проблема"),
  - **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 приводит к ошибке, значение функции становится Unknown и 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 (Проблема/OK). Поэтому, если состояние триггера было "Проблема" (см. Случай 1), оно остается в состоянии проблемы, даже если известная часть разрешилась ('1' становится '0'), потому что теперь выражение вычисляется в Unknown, а это не изменяет состояние триггера.

Если выражение триггера с несколькими неподдерживаемыми элементами данных вычисляется в Unknown, сообщение об ошибке в веб-интерфейсе ссылается на последний вычисленный неподдерживаемый элемент данных.

Кэширование значений

Значения, необходимые для вычисления триггеров, кэшируются сервером Zabbix. Из-за этого вычисление триггеров в течение некоторого времени после перезапуска сервера создает повышенную нагрузку на базу данных.

Кэш значений не очищается при удалении значений истории элементов данных (вручную или с помощью housekeeper), поэтому сервер будет использовать кэшированные значения, пока они не станут старше периодов времени, заданных в функциях триггеров, или пока сервер не будет перезапущен.

Note:
Если в кэше нет недавних данных и в функции не задан период запроса, Zabbix по умолчанию будет запрашивать из базы данных исторические значения вплоть до одной недели в прошлом.

1 Агрегатные функции

Если не указано иное, все перечисленные здесь функции поддерживаются в:

- Выражениях триггеров
- Вычисляемых элементах 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 вычисляемых элементах данных; не может использоваться в триггерах)

Функции перечислены без дополнительной информации. Нажмите на функцию, чтобы увидеть полные сведения.

Function	Description
avg	Среднее значение элемента данных в пределах заданного периода 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 в пределах заданного периода вычисления.
mad	Медианное абсолютное отклонение в собранных значениях в пределах заданного периода вычисления.
max	Наибольшее значение элемента данных в пределах заданного периода вычисления.
min	Наименьшее значение элемента данных в пределах заданного периода вычисления.
skewness	Асимметрия распределения вероятностей в собранных значениях в пределах заданного периода вычисления.
stddevpop	Среднеквадратичное отклонение генеральной совокупности в собранных значениях в пределах заданного периода вычисления.
stddevsamp	Выборочное среднеквадратичное отклонение в собранных значениях в пределах заданного периода вычисления.
sum	Сумма собранных значений в пределах заданного периода вычисления.
sumofsquares	Сумма квадратов собранных значений в пределах заданного периода вычисления.
varpop	Дисперсия генеральной совокупности собранных значений в пределах заданного периода вычисления.
varsamp	Выборочная дисперсия собранных значений в пределах заданного периода вычисления.

Общие параметры

- `/host/key` — это общий обязательный первый параметр для функций, ссылающихся на историю элемента данных узла сети
- `(sec|#num)<:time shift>` — это 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** (необязательно) позволяет сдвинуть точку вычисления назад во времени. Подробнее о задании time shift см. здесь.

Подробности о функциях

Некоторые общие замечания о параметрах функций:

- Параметры функций разделяются запятой
- Необязательные параметры функций (или части параметров) обозначаются с помощью < >
- Специфические для функции параметры описаны для каждой функции отдельно
- Параметры `/host/key` и `(sec|#num)<:time shift>` никогда не должны заключаться в кавычки

avg(/host/key,(sec|#num)<:time shift>)

Среднее значение элемента данных в пределах заданного периода вычисления.
 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 необходимо сравнить текущее среднее значение со средним значением за некоторое время в прошлом.

Примеры:

```
avg(/host/key,1h) #
avg(/host/key,1h:now-1d) #                    25          24
avg(/host/key,#5) #
avg(/host/key,#5:now-1d) #                    ,           ,           24
```

bucket_percentile(item filter,time period,percentage)

Вычисляет процентиль по бакетам гистограммы.

Параметры:

- **item filter** — см. синтаксис фильтра элемента данных;
- **time period** — см. период времени;
- **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))`.

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 необходимо 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. фильтр элементов данных;
- **time period** - см. период времени;
- **operator** (должен быть заключён в двойные кавычки). Поддерживаемые :
eq - равно
ne - не равно
gt - больше
ge - больше или равно
lt - меньше
le - меньше или равно
like - соответствует, если содержит шаблон (с учётом регистра)
bitand - побитовое И
regexp - чувствительное к регистру соответствие регулярному выражению, заданному в `pattern`
iregexp - нечувствительное к регистру соответствие регулярному выражению, заданному в `pattern`
- **pattern** - требуемый шаблон (строковые аргументы должны быть заключены в двойные кавычки); поддерживается, если *operator* указан в третьем параметре.

Комментарии:

- Использование **count()** с функцией foreach, связанной с историей данных (max_foreach, avg_foreach и т. д.), может повлиять на производительность, тогда как использование **exists_foreach()**, работающей только с данными конфигурации, такого эффекта не оказывает.
- Необязательные параметры *operator* или *pattern* нельзя оставлять пустыми после запятой, их можно только полностью опустить.
- Если в качестве третьего параметра используется *bitand*, четвёртый параметр `pattern` может быть задан как два числа, разделённые символом '/': **number_to_compare_with/mask**. count() вычисляет "побитовое И" 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, можно указать только *mask* (без '/').
- Если в качестве третьего параметра используется *regexp* или *iregexp*, четвёртый параметр `pattern` может быть обычным или глобальным (начинающимся с '@') регулярным выражением. В случае глобальных регулярных выражений чувствительность к регистру наследуется из настроек глобального регулярного выражения. Для целей сопоставления по regexp значения с плавающей точкой всегда будут представлены с 4 десятичными знаками после '.'. Также обратите внимание, что для больших чисел различие между десятичным представлением (хранимым в базе данных) и двоичным представлением (используемым сервером Zabbix) 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 ≤ φ ≤ 1;
- **bucketN, valueN** - вручную введённые пары (>=2) параметров 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, соответствующих критериям фильтра.
 Поддерживаемый тип значения: *Integer*.

Параметр:

- **item filter** — критерий выбора элементов данных; допускает ссылки по группе узлов сети, узлу сети, ключу элемента данных и тегам. Поддерживаются подстановочные знаки. Подробнее см. в разделе item filter.

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"]) #                    *agent.ping*   "Host group 1"
```

kurtosis(/host/key,(sec|#num)<:time shift>)

«Хвостатость» распределения вероятностей собранных значений в пределах заданного периода вычисления. См. также: Kurtosis.
 Поддерживаемые типы значений: *Float*, *Integer*.
 Поддерживаемая функция foreach: *last_foreach*.

Параметры: см. общие параметры.

Пример:

```
kurtosis(/host/key,1h) #kurtosis
```

mad(/host/key,(sec|#num)<:time shift>)

Медианное абсолютное отклонение собранных значений в пределах заданного периода вычисления. См. также: Медианное абсолютное отклонение.
 Поддерживаемые типы значений: *Float*, *Integer*.
 Поддерживаемая функция foreach: *last_foreach*.

Параметры: см. общие параметры.

Пример:

```
mad(/host/key,1h) #
```

max(/host/key,(sec|#num)<:time shift>)

Наибольшее значение элемента данных в пределах заданного периода вычисления.
 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.

Пример:

```
max(/host/key,1h) - min(/host/key,1h) #                                                    (
```

min(/host/key,(sec|#num)<:time shift>)

Наименьшее значение элемента данных в пределах заданного периода вычисления.
 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.

Пример:

```
max(/host/key,1h) - min(/host/key,1h) #                                                    (
```

skewness(/host/key,(sec|#num)<:time shift>)

Асимметрия распределения вероятностей собранных значений в пределах заданного периода вычисления. См. также: Skewness.
 Поддерживаемые типы значений: *Float*, *Integer*.
 Поддерживаемая функция foreach: *last_foreach*.

Параметры: см. общие параметры.

Пример:

```
skewness(/host/key,1h) #
```

stddevpop(/host/key,(sec|#num)<:time shift>)

Стандартное отклонение генеральной совокупности по собранным значениям в пределах заданного периода вычисления. См. также: Стандартное отклонение.
 Поддерживаемые типы значений: *Float*, *Integer*.
 Поддерживаемая функция foreach: *last_foreach*.

Параметры: см. общие параметры.

Пример:

```
stddevpop(/host/key,1h) #
```

stddevsamp(/host/key,(sec|#num)<:time shift>)

Выборочное стандартное отклонение собранных значений в пределах заданного периода вычисления. См. также: Стандартное отклонение.
 Поддерживаемые типы значений: *Float*, *Integer*.
 Поддерживаемая функция foreach: *last_foreach*.

Параметры: см. общие параметры.

Для работы этой функции требуется как минимум два значения данных.

Пример:

```
stddevsamp(/host/key,1h) #
```

sum(/host/key,(sec|#num)<:time shift>)

Сумма собранных значений в пределах заданного периода вычисления.
 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.

Пример:

```
sum(/host/key,1h) #
```

sumofsquares(/host/key,(sec|#num)<:time shift>)

Сумма квадратов собранных значений в пределах заданного периода вычисления.
 Поддерживаемые типы значений: *Float*, *Integer*.
 Поддерживаемая функция foreach: *last_foreach*.

Параметры: см. общие параметры.

Пример:

```
sumofsquares(/host/key,1h) #
```

varpop(/host/key,(sec|#num)<:time shift>)

Дисперсия генеральной совокупности собранных значений в пределах заданного периода вычисления. См. также: Variance.
 Поддерживаемые типы значений: *Float*, *Integer*.
 Поддерживаемая функция foreach: *last_foreach*.

Параметры: см. общие параметры.

Пример:

```
varpop(/host/key,1h) #
```

varsamp(/host/key,(sec|#num)<:time shift>)

Выборочная дисперсия собранных значений в пределах заданного периода вычисления. См. также: Variance.
 Поддерживаемые типы значений: *Float*, *Integer*.
 Поддерживаемая функция foreach: *last_foreach*.

Параметры: см. общие параметры.

Для работы этой функции требуется как минимум два значения данных.

Пример:

```
varsamp(/host/key,1h) #
```

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.

> **Attention:**
> Функции foreach поддерживаются только в вычисляемых элементах данных как часть формул агрегатных вычислений. Их нельзя вызывать в обычных ключах элементов данных или выражениях триггеров вне этого контекста.

Например, функция *avg_foreach* вернет массив значений, где каждое значение является *средним* значением истории выбранного элемента данных за указанный интервал времени.

Фильтр элементов данных является частью синтаксиса, используемого функциями foreach. В фильтре элементов данных поддерживается использование подстановочных знаков, поэтому необходимые элементы данных можно выбирать достаточно гибко.

Поддерживаемые функции

Функция	Описание
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, определённого <parameter number> параметром.
count_foreach	Возвращает количество значений для каждого элемента данных.
exists_foreach	Возвращает '1' для каждого включённого 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 функции

Функции foreach поддерживают два общих параметра: (подробности см. ниже) и :

```
foreach_function(item filter,time period)
```

Например:

```
avg_foreach(/*/mysql.qps?[group="MySQL Servers"],5m)
```

вернёт среднее значение за пять минут для каждого элемента данных 'mysql.qps' в группе серверов MySQL.

Обратите внимание, что некоторые функции поддерживают дополнительные параметры.

Синтаксис фильтра элементов данных

Фильтр элементов данных:

```
/host/key[parameters]?[conditions]
```

состоит из четырёх частей, где:

- host - имя узла сети
- key - ключ элемента данных (без параметров)
- parameters - параметры ключа элемента данных
- conditions - условия на основе группы узлов сети и/или тега элемента данных (в виде выражения)

Пробелы допускаются только внутри выражения conditions.

Использование подстановочного знака

- Подстановочный знак можно использовать для замены имени узла сети, ключа элемента данных или отдельного параметра ключа элемента данных.
- Либо host, либо key должны быть указаны без подстановочного знака. Таким образом, `/host/*` и `/*/key` являются допустимыми фильтрами, а `/*/*` — недопустимым.
- Подстановочный знак нельзя использовать для *части* имени узла сети, ключа элемента данных или параметра ключа элемента данных.

- Подстановочный знак не соответствует более чем одному параметру ключа элемента данных. Поэтому подстановочный знак должен быть указан отдельно для каждого параметра (то есть `key[abc,*,*]`).

Выражение conditions

Выражение conditions поддерживает:

- операнды:
  - group - группа узлов сети
  - tag - тег элемента данных
  - "<text>" - строковая константа с символом экранирования \ для экранирования " и \
- операторы сравнения строк с учётом регистра: =, <>
- логические операторы: and, or, not
- группировку с помощью скобок: ( )

Заключение строковых констант в кавычки обязательно. Поддерживается только полное сравнение строк с учётом регистра.

Warning:
При указании тегов в фильтре (то есть `tag="tagname:value"`) двоеточие ":" используется как разделитель. Всё, что следует после него, считается значением тега. Поэтому в настоящее время не поддерживается указание имени тега, содержащего ":".

Примеры

Можно использовать сложный фильтр со ссылками на ключ элемента данных, группу узлов сети и теги, как показано в примерах:

Пример синтаксиса	Описание
`/host/key[abc,*]`	Соответствует похожим элементам данных на этом узле сети.
`/*/key`	Соответствует одному и тому же элементу данных любого узла сети.
`/*/key?[group="ABC" and tag="tagname:value"]`	Соответствует одному и тому же элементу данных любого узла сети из группы ABC, имеющему теги 'tagname:value'.
`/*/key[a,*,c]?[(group="ABC" and tag="Tag1") or (group="DEF" and (tag="Tag2" or tag="Tag3:value"))]`	Соответствует похожим элементам данных любого узла сети из группы ABC или DEF с соответствующими тегами.

Все элементы данных, на которые есть ссылки, должны существовать и собирать данные. В вычисления включаются только включённые элементы данных на включённых узлах сети. Элементы данных в неподдерживаемом состоянии не включаются.

Attention:
Если ключ элемента данных, на который есть ссылка, изменён, фильтр необходимо обновить вручную.

Указание родительской группы узлов сети включает родительскую группу и все вложенные группы узлов сети вместе с их элементами данных.

Период времени

Второй параметр позволяет указать период времени для агрегации. Период времени может быть выражен только как время; количество значений (с префиксом **#**) не поддерживается.

Для удобства в этом параметре можно использовать поддерживаемые обозначения единиц, 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), который можно опустить:

`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,<          >)
```

где <номер параметра> — 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 <номер параметра> — ’1’.

count_foreach

Функция *count_foreach* поддерживает третий и четвёртый необязательные параметры:

```
count_foreach(item filter,time period,<operator>,<pattern>)
```

Где:

- **operator** — условный оператор для значений элемента данных (должен быть заключён 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 учётом регистра)
bitand — побитовое AND
regexp — чувствительное к регистру совпадение с регулярным выражением, заданным в pattern
iregexp — нечувствительное к регистру совпадение с регулярным выражением, заданным в pattern
- **pattern** — требуемый шаблон (строковые аргументы должны быть заключены в двойные кавычки); поддерживается, если *operator* указан в третьем параметре.

Комментарии:

- Необязательные параметры *operator* или *pattern* нельзя оставлять пустыми после запятой, их можно только полностью опустить.
- Если в качестве третьего параметра используется *bitand*, четвёртый параметр pattern можно указать как два числа, разделённые символом ’/’: **число_для_сравнения/маска**. count_foreach() вычисляет ”побитовое AND” значения и *маски* и сравнивает результат с *число_для_сравнения*. Если результат ”побитового AND” равен *число_для_сравнения*, значение учитывается.
Если *число_для_сравнения* и *маска* равны, достаточно указать только *маску* (без ’/’).
- Если в качестве третьего параметра используется *regexp* или *iregexp*, четвёртый параметр pattern может быть обычным или глобальным (начинающимся с ’@’) регулярным выражением. В случае глобальных регулярных выражений чувствительность к регистру наследуется из настроек глобального регулярного выражения. Для целей сопоставления regexp значения с плавающей точкой всегда будут представлены с 4 десятичными знаками после ’.’. Также обратите внимание, что для больших чисел различие между десятичным представлением (хранимым в базе данных) и двоичным представлением (используемым сервером Zabbix) может влиять на 4-й знак после запятой.

См. агрегированные вычисления для получения дополнительных сведений и примеров использования функций foreach.

Поведение в зависимости от доступности

Следующая таблица показывает, как ведет себя каждая функция в случаях ограниченной доступности узла сети/элемента данных и исторических данных.

Функция	Отключенный узел сети	Недоступный узел сети с данными	Недоступный узел сети без данных	Отключенный элемент данных	Неподдерживаемый элемент данных	Ошибка получения 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	н/д
last_foreach	игнорировать	вернуть последнее значение	игнорировать	игнорировать	игнорировать	игнорировать
max_foreach	игнорировать	вернуть максимум	игнорировать	игнорировать	игнорировать	игнорировать
min_foreach	игнорировать	вернуть минимум	игнорировать	игнорировать	игнорировать	игнорировать
sum_foreach	игнорировать	вернуть сумму	игнорировать	игнорировать	игнорировать	игнорировать

Если элемент данных *игнорируется*, в агрегирование ничего не добавляется.

2 Побитовые функции

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
bitand	Значение результата ”побитового И” для значения элемента данных и маски.
bitlshift	Побитовый сдвиг влево значения элемента данных.
bitnot	Значение результата ”побитового НЕ” для значения элемента данных.
bitor	Значение результата ”побитового ИЛИ” для значения элемента данных и маски.
bitrshift	Побитовый сдвиг вправо значения элемента данных.
bitxor	Значение результата ”побитового исключающего ИЛИ” для значения элемента данных и маски.

Подробности о функциях

Некоторые общие замечания о параметрах функций:

- Параметры функций разделяются запятой
- В качестве параметров допускаются выражения
- Необязательные параметры функций (или части параметров) обозначаются < >

bitand(value,mask)

Значение побитового ”AND” значения элемента данных и маски.
 Поддерживаемые типы значений: *Целое число*.

Параметры:

- **value** — проверяемое значение;
- **mask** (обязательный) — 64-битное целое число без знака (0 - 18446744073709551615).

Хотя сравнение выполняется побитово, все значения должны передаваться и возвращаются в десятичном виде. Например, проверка 3-го бита выполняется сравнением с 4, а не с 100.

Примеры:

```
bitand(last(/host/key),12)=8 or bitand(last(/host/key),12)=4 #       3-    4-    ,
bitand(last(/host/key),20)=16 # 3-           ,   5-
```

bitlshift(value,bits to shift)

Побитовый сдвиг влево значения элемента данных.
 Поддерживаемые типы значений: *Integer*.

Параметры:

- **value** — значение для проверки;
- **bits to shift** (обязательный) — количество битов для сдвига.

Хотя сравнение выполняется побитово, все значения должны передаваться и возвращаются в десятичном виде. Например, проверка 3-го бита выполняется сравнением с 4, а не с 100.

bitnot(value)

Значение операции ”побитовое НЕ” для значения элемента данных.
 Поддерживаемые типы значений: *Integer*.

Параметр:

- **value** — значение для проверки.

Хотя сравнение выполняется побитово, все значения должны быть указаны и возвращаются в десятичном виде. Например, проверка 3-го бита выполняется сравнением с 4, а не с 100.

bitor(value,mask)

Значение операции ”побитовое ИЛИ” для значения элемента данных и маски.
 Поддерживаемые типы значений: *Целое число*.

Параметры:

- **value** - значение для проверки;
- **mask** (обязательный) - 64-битное беззнаковое целое число (0 - 18446744073709551615).

Хотя сравнение выполняется побитово, все значения должны передаваться и возвращаются в десятичном виде. Например, проверка 3-го бита выполняется сравнением с 4, а не с 100.

bitrshift(value,bits to shift)

Побитовый сдвиг вправо значения элемента данных.
 Поддерживаемые типы значений: *Integer*.

Параметры:

- **value** — значение для проверки;
- **bits to shift** (обязательный) — количество битов для сдвига.

Хотя сравнение выполняется побитово, все значения должны передаваться и возвращаются в десятичном виде. Например, проверка 3-го бита выполняется сравнением с 4, а не с 100.

bitxor(value,mask)

Значение операции ”исключающее ИЛИ по битам” для значения элемента данных и маски.
 Поддерживаемые типы значений: *Целое число*.

Параметры:

- **value** — значение для проверки;
- **mask** (обязательный) — 64-битное беззнаковое целое число (0 - 18446744073709551615).

Хотя сравнение выполняется побитово, все значения должны передаваться и возвращаются в десятичном виде. Например, проверка 3-го бита выполняется сравнением с 4, а не с 100.

См. все поддерживаемые функции.

3 Функции даты и времени

Все перечисленные здесь функции поддерживаются в:

- Выражениях триггеров
- Вычисляемых элементах данных
- Макросах выражений

Attention:
Функции даты и времени не могут использоваться 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, смотрите в разделе Время вычисления.

Функции перечислены без дополнительной информации. Нажмите на функцию, чтобы увидеть полные сведения.

Функция	Описание
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, прошедших с начала эпохи (00:00:00 UTC, 1 января 1970 года).
time	Текущее время в формате HHMMSS.

Подробности функции

date

Текущая дата в формате YYYYMMDD.

Пример:

`date()<20220101`

dayofmonth

День 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 будние дни):

`dayofweek()<6`

Пример (только выходные):

`dayofweek()>5`

now

Количество секунд, прошедших с начала эпохи (00:00:00 UTC, 1 января 1970 года).

Пример:

`now()<1640998800`

time

Текущее время в формате HHMMSS.

Пример (только ночное время, 00:00-06:00):

`time()<060000`

См. все поддерживаемые функции.

4 Функции истории

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
change	Величина разницы между предыдущим и последним значением.
changecount	Количество изменений между соседними значениями в пределах заданного периода вычисления.
count	Количество значений в пределах заданного периода вычисления.
countunique	Количество уникальных значений в пределах заданного периода вычисления.
find	Поиск совпадения значения в пределах заданного периода вычисления.
first	Первое (самое старое) значение в пределах заданного периода вычисления.
firstclock	Метка времени первого (самого старого) значения в пределах заданного периода вычисления.
fuzzytime	Проверка того, насколько время пассивного агента отличается от времени сервера/прокси Zabbix.
last	Самое последнее значение.
lastclock	Метка времени N-го самого последнего значения в пределах заданного периода вычисления.
logeventid	Проверка того, соответствует ли ID события последней записи журнала регулярному выражению.
logseverity	Уровень важности журнала для последней записи журнала.
logsource	Проверка того, соответствует ли источник журнала последней записи журнала регулярному выражению.
logtimestamp	Метка времени сообщения журнала для N-го самого последнего значения элемента данных журнала.
monodec	Проверка, было ли монотонное уменьшение значений.
monoinc	Проверка, было ли монотонное увеличение значений.
nodata	Проверка отсутствия полученных данных.
percentile	P-й процентиль периода, где P (процент) задаётся третьим параметром.
rate	Средняя скорость прироста в секунду для монотонно увеличивающегося счётчика в пределах заданного периода времени.

Общие параметры

- `/host/key` — это общий обязательный первый параметр для функций, ссылающихся на историю элемента данных узла сети
- `(sec|#num)<:time shift>` — это общий второй параметр для функций, ссылающихся на историю элемента данных узла сети, где:
  - **sec** — максимальный период вычисления в секундах (могут использоваться суффиксы времени), или
  - **#num** — максимальный диапазон вычисления в последних собранных значениях (если перед ним стоит знак решётки)
  - **time shift** (необязательно) позволяет сдвинуть точку вычисления назад во времени. См. подробности о задании time shift.

Подробности о функциях

Некоторые общие замечания о параметрах функций:

- Параметры функций разделяются запятой
- Необязательные параметры функций (или части параметров) обозначаются с помощью < >
- Специфические для функции параметры описаны для каждой функции
- Параметры `/host/key` и `(sec|#num)<:time shift>` 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:

- Будет вычислена числовая разница, как показано в следующих примерах входящих значений (’предыдущее’ и ’последнее’ значение = 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 счётчиков, переходов числового состояния.
- См. также: abs для сравнения.

Примеры:

```
change(/host/system.uptime)<0 #                                                    (                )
change(/host/system.cpu.load[all,avg1])>2 #      CPU (         )              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 вычисления.
 Поддерживаемые типы значений: *Float*, *Integer*, *String*, *Text*, *Log*.

Параметры:

- См. общие параметры
- **mode** (должен быть заключён в двойные кавычки) — возможные значения: *all* — учитывать все изменения (по умолчанию); *dec* — учитывать уменьшения; *inc* — учитывать увеличения

Комментарии:

- Для нечисловых типов значений параметр *mode* игнорируется.
- В триггерах полезно для обнаружения 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 #          eth0          5
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 вычисления.
 Поддерживаемые типы значений: *Float*, *Integer*, *String*, *Text*, *Log*.

Параметры:

- См. общие параметры

- **operator** (должен быть заключён 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 учётом регистра)
bitand - побитовое AND
regexp - чувствительное к регистру совпадение с регулярным выражением, заданным в pattern
iregexp - нечувствительное к регистру совпадение с регулярным выражением, заданным в pattern
- **pattern** - требуемый шаблон (строковые аргументы должны быть заключены в двойные кавычки)

Комментарии:

- Float элементы данных сопоставляются с точностью 2.22e-16.
- *like* не поддерживается в качестве оператора для целочисленных значений.
- *like* и *bitand* не поддерживаются в качестве операторов для значений float.
- Для значений string, text и log поддерживаются только операторы *eq*, *ne*, *like*, *regexp* и *iregexp*.
- При использовании *bitand* в качестве оператора четвёртый параметр pattern может быть указан как два числа, разделённые символом '/': **number_to_compare_with/mask**. count() вычисляет "побитовое AND" значения и *mask* и сравнивает результат с *number_to_compare_with*. Если результат "побитового AND" 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 четвёртый параметр pattern может быть обычным или глобальным (начинающимся с '@') регулярным выражением. В случае глобальных регулярных выражений чувствительность к регистру наследуется из настроек глобального регулярного выражения. Для целей сопоставления по regexp значения float всегда будут представлены с 4 десятичными знаками после '.'. Также обратите внимание, что для больших чисел различие между десятичным представлением (хранится в базе данных) и двоичным представлением (используется сервером Zabbix) может влиять на 4-й знак после запятой.

Примеры:

```
count(/host/icmpping,30m,,"0")>5 #ping failed more than 5 times in 30 minutes
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 в пределах заданного периода вычисления.
 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 — чувствительное к регистру совпадение с регулярным выражением, заданным в pattern
iregexp — нечувствительное к регистру совпадение с регулярным выражением, заданным в pattern
- **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 в качестве оператора для целочисленных значений.
- *like* и *bitand* не поддерживаются в качестве операторов 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 может быть указан как два числа, разделенные символом '/': **number_to_compare_with/mask**. countunique() вычисляет "побитовое И" 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, можно указать только *mask* (без '/').
- При использовании *regexp* или *iregexp* в качестве оператора четвертый параметр pattern может быть обычным или глобальным (начинающимся с '@') регулярным выражением. В случае глобальных регулярных выражений чувствительность к регистру наследуется из настроек глобального регулярного выражения. Для целей сопоставления regexp значения float всегда будут представлены с 4 десятичными знаками после '.'. Также

обратите внимание, что для больших чисел различие между десятичным представлением (хранимым в базе данных) и двоичным представлением (используемым сервером Zabbix) 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 вычисления.
 Поддерживаемые типы значений: *Float*, *Integer*, *String*, *Text*, *Log*.
 Возвращает: 1 — найдено; 0 — в противном случае.

Параметры:

- См. общие параметры
- **sec** или **#num** (необязательно) — если не указано, по умолчанию используется последнее значение
- **operator** (должен быть заключён 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 строку, указанную в `pattern` (с учётом регистра)
bitand — побитовое И
regexp — чувствительное к регистру совпадение с регулярным выражением, указанным в `pattern`
iregexp — нечувствительное к регистру совпадение с регулярным выражением, указанным в `pattern`
- **pattern** — требуемый шаблон (строковые аргументы должны быть заключены в двойные кавычки); регулярное выражение Perl Compatible Regular Expression (PCRE), если `operator` — *regexp*, *iregexp*

Комментарии:

- Если обрабатывается более одного значения, возвращается '1', если есть хотя бы одно совпадающее значение.
- *like* не поддерживается в качестве оператора для целочисленных значений.
- *like* и *bitand* не поддерживаются в качестве операторов для значений с плавающей точкой.
- Для значений string, text и log поддерживаются только операторы *eq*, *ne*, *like*, *regexp* и *iregexp*.
- При использовании *regexp* или *iregexp* в качестве оператора четвёртый параметр `pattern` может быть обычным или глобальным (начинающимся с '@') регулярным выражением. В случае глобальных регулярных выражений чувствительность к регистру наследуется из настроек глобального регулярного выражения.

Примеры:

```
find(/host/key,10m,"like","error") #find a value that contains 'error' within the last 10 minutes until no
find(/host/agent.version,,"like","beta")=1 #Zabbix agent has beta version, must be upgraded
find(/host/log[/var/log/nginx/access.log],,"regexp"," 500 ")=1 #internal web server error has been found
```

first(/host/key,sec<:time shift>)

Первое (самое старое) значение в пределах заданного периода вычисления.
 Поддерживаемые типы значений: *Float*, *Integer*, *String*, *Text*, *Log*.

Параметры:

- См. общие параметры

См. также last().

Пример:

```
first(/host/key,1h) #retrieve the oldest value within the last hour until now
```

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 пассивного элемента данных (в виде временной метки) и временной меткой сервера/прокси Zabbix (временем сбора значения) меньше или равна *sec* секунд; 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 агенте 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.
- Эту функцию не рекомендуется использовать в сложных выражениях триггеров (с несколькими элементами 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 #
```

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 счёту последнее значение

Комментарии:

- Обратите внимание, что период времени с решёткой (#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 счёту последнее значение (*а не три последних значения*).
- Zabbix не гарантирует точный порядок значений, если в истории в пределах одной секунды 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(/host/key,<#num<:time shift»,<pattern>)

Проверяет, соответствует ли ID события последней записи журнала регулярному выражению.
 Поддерживаемые типы значений: *Журнал*.
 Возвращает: 0 — не соответствует; 1 — соответствует.

Параметры:

- См. общие параметры
- **#num** (необязательно) — N-е по свежести значение
- **pattern** (необязательно) — регулярное выражение, описывающее требуемый шаблон, в стиле Perl Compatible Regular Expression (PCRE) (строковые аргументы должны быть заключены в двойные кавычки)

Примеры:

```
logeventid(/host/eventlog[Security],,"4625")=1 #               ID,          "4625" (             )
logeventid(/host/eventlog[System],,"6008|41")=1 #               ID,          "6008"    "41"
```

logseverity(/host/key,<#num<:time shift»)

Серьёзность последней записи журнала.
 Поддерживаемые типы значений: *Log*.
 Возвращает: 0 — серьёзность по умолчанию; N — серьёзность (целое число, полезно для журналов событий Windows: 1 — Information, 2 — Warning, 4 — Error, 7 — Failure Audit, 8 — Success Audit, 9 — Critical, 10 — Verbose).

Параметры:

- См. общие параметры
- **#num** (необязательно) — N-е по счёту самое последнее значение

Zabbix получает серьёзность журнала из поля **Information** журнала событий Windows.

Примеры:

```
logseverity(/host/log[/var/log/syslog],10m)>3 #a log entry with severity above "3" found
logseverity(/host/eventlog[System],10m)=4 #a log entry with severity equaling "Error" found
```

logsource(/host/key,<#num<:time shift»,<pattern>)

Проверяет, соответствует ли источник журнала последней записи журнала регулярному выражению.
 Поддерживаемые типы значений: *Log*.
 Возвращает: 0 — не соответствует; 1 — соответствует.

Параметры:

- См. общие параметры
- **#num** (необязательно) — N-е по счёту последнее значение
- **pattern** (необязательно) — регулярное выражение, описывающее требуемый шаблон, в стиле Perl Compatible Regular Expression (PCRE) (строковые аргументы должны быть заключены в двойные кавычки)

Обычно используется для журналов событий Windows.

Примеры:

```
logsource(/host/eventlog[Application],,"MSSQLSERVER")=1 #                    ,          "MSSQLSERVER"
logsource(/host/eventlog[System],,"Service Control Manager")=1 #                     ,          "Service C
logsource(/host/eventlog[System],,"Service Control Manager")=1 and logeventid(/host/eventlog[System],,"703
```

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(/host/key,(sec|#num)<:time shift>,<mode>)

Проверяет, было ли монотонное уменьшение значений.
 Поддерживаемые типы значений: *Целое число*.
 Возвращает: 1 — если все элементы данных за период времени непрерывно уменьшаются; 0 — в противном случае.

Параметры:

- См. общие параметры
- **mode** (должен быть заключён в двойные кавычки) — *weak* (каждое значение меньше или равно предыдущему; по умолчанию) или *strict* (каждое значение уменьшилось)

Примеры:

```
monodec(/host/system.swap.size[all,free],60s) + monodec(/host2/system.swap.size[all,free],60s) + monodec(/
monodec(/host/proc.num[nginx],10m,"strict")=1 #            nginx                    10
```

monoinc(/host/key,(sec|#num)<:time shift>,<mode>)

Проверяет, было ли монотонное увеличение значений.
 Поддерживаемые типы значений: *Integer*.
 Возвращает: 1 — если все элементы данных за период времени непрерывно увеличиваются; 0 — в противном случае.

Параметры:

- См. общие параметры
- **mode** (должен быть заключён в двойные кавычки) — *weak* (каждое значение больше или равно предыдущему; по умолчанию) или *strict* (каждое значение увеличилось)

Примеры:

```
monoinc(/host/system.localtime,#3,"strict")=0 #
monoinc(/host/vfs.dir.count[/mnt/data/logs],24h,"weak")=0 #    ,                          24     (
```

nodata(/host/key,sec,<mode>)

Проверяет отсутствие полученных данных.
 Поддерживаемые типы значений: *Целое число*, *Число с плавающей точкой*, *Символ*, *Текст*, *Журнал*.
 Возвращает: 1 — если данные не поступали в течение заданного периода времени; 0 — в противном случае.

Параметры:

- См. общие параметры
- **sec** — период должен быть не менее 30 секунд, поскольку процесс синхронизации истории вычисляет эту функцию только каждые 30 секунд; `nodata(/host/key,0)` не допускается
- **mode** — если установлено значение *strict* (в двойных кавычках), эта функция не будет зависеть от доступности прокси (подробности см. в комментариях)

Комментарии:

- триггеры 'nodata', отслеживаемые прокси, по умолчанию 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, но 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, как только пройдет период оценки (пять минут) без данных.
- Эта функция отобразит ошибку, если в течение периода, заданного 1-м параметром:
- данные отсутствовали, и сервер Zabbix был перезапущен
- данные отсутствовали, и обслуживание было завершено
- данные отсутствовали, и элемент данных был добавлен или повторно включен.
- Ошибки отображаются в столбце *Info* в настройках триггера.

- Эта функция может работать некорректно, если есть различия во времени между сервером Zabbix, прокси и агентом. См. также: Требование синхронизации времени.

Пример:

```
nodata(/host/agent.ping,5m)=1 #      ,           Zabbix                    5
```

percentile(/host/key,(sec|#num)<:time shift>,percentage)

P-й процентиль за период, где P (percentage) задается третьим параметром.
 Поддерживаемые типы значений: *Float*, *Integer*.

Параметры:

- См. общие параметры
- **percentage** — число с плавающей точкой от 0 до 100 (включительно) с не более чем 4 знаками после десятичной точки

Примеры:

```
percentile(/host/net.if.in[eth0,bytes],1h,95)>1000000 #95-                          (   /  )    1
percentile(/host/system.cpu.util,5m,95)>80 #95-                       CPU                        80
percentile(/host/icmppingsec[192.168.0.2],15m,95)>0.15 #                        150   ,                    (
percentile(/host/net.if.in[eth0,bytes],1h,50) #      50-         (           )                      ;
(percentile(/host/net.if.in[eth0,bytes],1h,50)+percentile(/host/net.if.in[eth0,bytes],1h,51))/2 #
```

rate(/host/key,sec<:time shift>)

Среднее увеличение в секунду монотонно возрастающего счётчика в пределах заданного периода 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 KB/s          5
rate(/host/app.requests.count,1m)>100 #                        >100
```

См. все поддерживаемые функции.

5 Функции трендов

Функции трендов, в отличие от функций истории, используют данные трендов для вычислений.

Тренды хранят агрегированные почасовые значения. Функции трендов используют эти почасовые средние значения, поэтому они полезны для долгосрочного анализа.

Результаты функций трендов кэшируются, поэтому при нескольких вызовах одной и той же функции с одинаковыми параметрами информация из базы данных извлекается только один раз. Кэш функций трендов управляется параметром сервера TrendFunctionCacheSize.

Триггеры, которые ссылаются **только** на функции трендов, вычисляются один раз за наименьший период времени 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 функции истории (или функции 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 количество отклонений (по алгоритму stddevpop) между последним периодом данных и теми же периодами данных в предыдущих сезонах.
baselinewma	Вычисляет базовую линию, усредняя данные из того же временного интервала в нескольких равных периодах времени (”сезонах”) с использованием алгоритма взвешенного скользящего среднего.
trendavg	Среднее значение трендов в пределах заданного периода времени.
trendcount	Количество успешно полученных значений истории, использованных для вычисления значения тренда в пределах заданного периода времени.
trendmax	Максимум значений трендов в пределах заданного периода времени.
trendmin	Минимум значений трендов в пределах заданного периода времени.
trendstl	Возвращает долю аномалий за период обнаружения — десятичное значение от 0 до 1, которое равно (( )/( )).
trendsum	Сумма значений трендов в пределах заданного периода времени.

Общие параметры

- /host/key — это общий обязательный первый параметр
- time period:time shift — это общий второй параметр, где:
  - **time period** — период времени (минимум ’1h’), задаваемый как <N><time unit>, где N — количество единиц времени, time unit — h (час), d (день), w (неделя), M (месяц) или y (год).
  - **time shift** — смещение периода времени (см. примеры функций)

Подробности о функциях

Некоторые общие примечания о параметрах функций:

- Параметры функций разделяются запятой
- Необязательные параметры функций (или части параметров) обозначаются с помощью < >
- Специфические для функции параметры описаны для каждой функции отдельно
- Параметры /host/key и time period:time shift никогда не должны заключаться в кавычки

baselinedev(/host/key,data period:time shift,season unit,num seasons)

Возвращает количество отклонений (по алгоритму stddevpop) между последним периодом данных и такими же периодами данных в предыдущих сезонах.

Параметры:

- См. общие параметры
- **data period** — период сбора данных в пределах сезона, задаётся как <N><time unit>, где:
N — количество единиц времени
time unit — h (час), d (день), w (неделя), M (месяц) или y (год), должно быть равно season или меньше него
- **season unit** — календарная единица, определяющая один сезон (h, d, w, M, y); не может быть меньше data period
- **num seasons** — количество сезонов для оценки

Примеры:

```
baselinedev(/host/key,1d:now/d,"M",6) #                    (          )
baselinedev(/host/key,1h:now/h,"d",10)  #
```

baselinewma(/host/key,data period:time shift,season unit,num seasons)

Вычисляет базовый уровень путём усреднения данных из одного и того же временного интервала в нескольких равных периодах времени (”сезонах”) с использованием алгоритма взвешенного скользящего среднего.

Параметры:

- См. общие параметры
- **data period** — период сбора данных в пределах сезона, задаётся как <N><time unit>, где:
N — количество единиц времени
time unit — h (час), d (день), w (неделя), M (месяц) или y (год), должно быть равно сезону или меньше него
Time shift — смещение периода времени, определяет конец временного интервала сбора данных в сезонах (см. примеры)
- **season unit** — календарная единица, определяющая один сезон (h, d, w, M, y); не может быть меньше, чем data period
- **num seasons** — количество сезонов для оценки

Примеры:

```
baselinewma(/host/key,1h:now/h,"d",3) #                                    3       ,
baselinewma(/host/key,2h:now/h,"d",3) #                                          3       ,
baselinewma(/host/key,1d:now/d,"M",4) #                          ,     "   ",   4     ,
```

trendavg(/host/key,time period:time shift)

Среднее значений трендов в пределах заданного периода времени.

Параметры:

- См. общие параметры

Примеры:

```
trendavg(/host/key,1h:now/h) #               (     , 12:00-13:00)
trendavg(/host/key,1h:now/h-1h) #               (11:00-12:00)
trendavg(/host/key,1h:now/h-2h) #               (10:00-11:00)
trendavg(/host/key,1M:now/M-1y) #
```

trendcount(/host/key,time period:time shift)

Количество успешно полученных значений истории, использованных для вычисления значения тренда в пределах заданного периода времени.

Параметры:

- См. общие параметры

Примеры:

```
trendcount(/host/key,1h:now/h) #                      (     , 12:00-13:00)
trendcount(/host/key,1h:now/h-1h) #                      (11:00-12:00)
trendcount(/host/key,1h:now/h-2h) #                      (10:00-11:00)
trendcount(/host/key,1M:now/M-1y) #
```

trendmax(/host/key,time period:time shift)

Максимум в значениях трендов в пределах заданного периода времени.

Параметры:

- См. общие параметры

Примеры:

```
trendmax(/host/key,1h:now/h) #                (     , 12:00-13:00)
trendmax(/host/key,1h:now/h) - trendmin(/host/key,1h:now/h) →                                (
trendmax(/host/key,1h:now/h-1h) #                     (11:00-12:00)
trendmax(/host/key,1h:now/h-2h) #                     (10:00-11:00)
trendmax(/host/key,1M:now/M-1y) #
```

trendmin(/host/key,time period:time shift)

Минимум в значениях трендов в пределах заданного периода времени.

Параметры:

- См. общие параметры

Примеры:

```
trendmin(/host/key,1h:now/h) #                (     , 12:00-13:00)
trendmax(/host/key,1h:now/h) - trendmin(/host/key,1h:now/h) →                                (
trendmin(/host/key,1h:now/h-1h) #                     (11:00-12:00)
trendmin(/host/key,1h:now/h-2h) #                     (10:00-11:00)
trendmin(/host/key,1M:now/M-1y) #
```

trendstl(/host/key,eval period:time shift,detection period,season,<deviations>,<devalg>,<s window>)

Возвращает долю аномалий за период обнаружения — десятичное значение от 0 до 1, которое вычисляется как ((
)/( )).

Параметры:

- См. общие параметры

- **eval period** — период времени, который должен быть декомпозирован (минимум '1h'), задаётся в формате <N><time unit>, где
N — количество единиц времени
time unit — h (час), d (день), w (неделя), M (месяц) или y (год)
- **detection period** — период времени перед концом eval period, для которого вычисляются аномалии (минимум '1h', не может быть длиннее eval period), задаётся в формате <N><time unit>, где
N — количество единиц времени
time unit — h (час), d (день), w (неделя)
- **season** — кратчайший период времени, в котором ожидается повторяющийся шаблон ("сезон") (минимум '2h', не может быть длиннее eval period, количество записей в eval period должно быть больше удвоенного значения результирующей частоты (season/h)), задаётся в формате <N><time unit>, где
N — количество единиц времени
time unit — h (час), d (день), w (неделя)
- **deviations** — количество отклонений (вычисляемых с помощью devalg), при котором значение считается аномалией (может быть дробным), (должно быть больше или равно 1, по умолчанию 3)
- **devalg** (должен быть заключён в двойные кавычки) — алгоритм отклонения; может быть *stddevpop*, *stddevsamp* или *mad* (по умолчанию)
- **s window** — ширина (в лагах) окна loess для выделения сезонной компоненты (по умолчанию 10 * количество записей в eval period + 1)

Примеры:

```
trendstl(/host/key,100h:now/h,10h,2h) #              100             ,                    10
trendstl(/host/key,100h:now/h-10h,100h,2h,2.1,"mad") #             100                     10       ,
trendstl(/host/key,100d:now/d-1d,10d,1d,4,,10) #        100                    ,                    10
trendstl(/host/key,1M:now/M-1y,1d,2h,,"stddevsamp") #                        ,
```

trendsum(/host/key,time period:time shift)

Сумма значений трендов в пределах заданного периода времени.

Параметры:

- См. общие параметры

Примеры:

```
trendsum(/host/key,1h:now/h) #               (     , 12:00-13:00)
trendsum(/host/key,1h:now/h-1h) #               (11:00-12:00)
trendsum(/host/key,1h:now/h-2h) #               (10:00-11:00)
trendsum(/host/key,1M:now/M-1y) #
```

См. все поддерживаемые функции.

6 Математические функции

Все перечисленные здесь функции поддерживаются в:

- Выражениях триггеров
- Вычисляемых элементах данных
- Макросах выражений

Математические функции поддерживаются для типов значений float и integer, если не указано иное.

Функции перечислены без дополнительной информации. Нажмите на функцию, чтобы увидеть полные сведения.

Function	Description
abs	Абсолютное значение величины.
acos	Арккосинус величины как угла, выраженного в радианах.
asin	Арксинус величины как угла, выраженного в радианах.
atan	Арктангенс величины как угла, выраженного в радианах.
atan2	Арктангенс ординаты (значения) и координат абсциссы, заданный как угол, выраженный в радианах.
avg	Среднее значение значений указанного элемента данных.
cbrt	Кубический корень из величины.
ceil	Округляет величину вверх до ближайшего большего или равного целого числа.
cos	Косинус величины, где величина является углом, выраженным в радианах.
cosh	Гиперболический косинус величины.
cot	Котангенс величины, где величина является углом, выраженным в радианах.
degrees	Преобразует величину из радиан в градусы.

Function	Description
e	Число Эйлера (2.718281828459045).
exp	Число Эйлера в степени величины.
expm1	Число Эйлера в степени величины минус 1.
floor	Округляет величину вниз до ближайшего меньшего или равного целого числа.
log	Натуральный логарифм.
log10	Десятичный логарифм.
max	Наибольшее значение среди значений указанного элемента данных.
min	Наименьшее значение среди значений указанного элемента данных.
mod	Остаток от деления.
pi	Константа Пи (3.14159265358979).
power	Возведение величины в степень.
radians	Преобразует величину из градусов в радианы.
rand	Возвращает случайное целое значение.
round	Округляет величину до указанного числа десятичных знаков.
signum	Возвращает '-1', если величина отрицательная, '0', если величина равна нулю, '1', если величина положительная.
sin	Синус величины, где величина является углом, выраженным в радианах.
sinh	Гиперболический синус величины, где величина является углом, выраженным в радианах.
sqrt	Квадратный корень из величины.
sum	Сумма значений указанного элемента данных.
tan	Тангенс величины.
truncate	Усекает величину до указанного числа десятичных знаков.

Подробности о функциях

Некоторые общие замечания о параметрах функций:

- Параметры функций разделяются запятой
- В качестве параметров допускаются выражения
- Необязательные параметры функций (или части параметров) обозначаются < >

abs(value)

Абсолютное значение (от 0) величины.

Параметр:

- **value** — значение для проверки

Например, абсолютное значение как '3', так и '-3' будет равно '3'.

Пример:

```
abs(last(/host/key))>10
```

acos(value)

Арккосинус значения как угол, выраженный в радианах.

Параметр:

- **value** — проверяемое значение

Значение должно быть в диапазоне от -1 до 1. Например, арккосинус значения '0.5' будет равен '2.0943951'.

Пример:

```
acos(last(/host/key))
```

asin(value)

Арксинус значения как угол, выраженный в радианах.

Параметр:

- **value** — проверяемое значение

Значение должно быть в диапазоне от -1 до 1. Например, арксинус значения '0.5' будет равен '-0.523598776'.

Пример:

```
asin(last(/host/key))
```

atan(value)

Арктангенс значения как угол, выраженный в радианах.

Параметр:

- **value** — значение для проверки

Например, арктангенс значения ’1’ будет равен ’0.785398163’.

Пример:

```
atan(last(/host/key))
```

atan2(value,abscissa)

Арктангенс ординаты (value) и координаты абсциссы, заданный как угол, выраженный в радианах.

Параметры:

- **value** - проверяемое значение;
- **abscissa** - значение абсциссы.

Например, арктангенс координат ординаты и абсциссы для значения ’1’ будет равен ’2.21429744’.

Пример:

```
atan2(last(/host/key),2)
```

avg(<value1>,<value2>,...)

Среднее значение значений элементов данных, на которые есть ссылка.

Параметр:

- **valueX** — значение, возвращаемое другой функцией, работающей с историей элемента данных.

Пример:

```
avg(avg(/host/key,1h),avg(/host2/key2,1h))
```

cbrt(value)

Кубический корень значения.

Параметр:

- **value** — значение для проверки

Например, кубический корень из ’64’ будет равен ’4’, из ’63’ — ’3.97905721’.

Пример:

```
cbrt(last(/host/key))
```

ceil(value)

Округляет значение вверх до ближайшего большего или равного целого числа.

Параметр:

- **value** — значение для проверки

Например, ’2.4’ будет округлено вверх до ’3’. См. также floor().

Пример:

```
ceil(last(/host/key))
```

cos(value)

Косинус значения, где значение — это угол, выраженный в радианах.

Параметр:

- **value** — значение для проверки

Например, косинус значения ’1’ будет равен ’0.54030230586’.

Пример:

```
cos(last(/host/key))
```

cosh(value)

Гиперболический косинус значения. Возвращает значение как вещественное число, а не в экспоненциальной форме.

Параметр:

- **value** — значение для проверки

Например, гиперболический косинус значения ’1’ будет равен ’1.54308063482’.

Пример:

```
cosh(last(/host/key))
```

cot(value)

Котангенс значения, где значение — это угол, выраженный в радианах.

Параметр:

- **value** — проверяемое значение

Например, котангенс значения ’1’ будет равен ’0.54030230586’.

Пример:

```
cot(last(/host/key))
```

degrees(value)

Преобразует значение из радиан в градусы.

Параметр:

- **value** — значение для проверки

Например, значение ’1’, преобразованное в градусы, будет равно ’57.2957795’.

Пример:

```
degrees(last(/host/key))
```

e

Число Эйлера (2.718281828459045).

Пример:

```
e()
```

exp(value)

Число Эйлера в степени значения.

Параметр:

- **value** — значение для проверки

Например, число Эйлера в степени значения ’2’ будет равно ’7.38905609893065’.

Пример:

```
exp(last(/host/key))
```

expm1(value)

Число Эйлера в степени значения минус 1.

Параметр:

- **value** — значение для проверки

Например, число Эйлера в степени значения ’2’ минус 1 будет равно ’6.38905609893065’.

Пример:

```
expm1(last(/host/key))
```

floor(value)

Округляет значение вниз до ближайшего меньшего или равного целого числа.

Параметр:

- **value** — значение для проверки

Например, ’2.6’ будет округлено вниз до ’2’. См. также ceil().

Пример:

```
floor(last(/host/key))
```

log(value)

Натуральный логарифм.

Параметр:

- **value** — значение для проверки

Например, натуральный логарифм значения ’2’ будет равен ’0.69314718055994529’.

Пример:

```
log(last(/host/key))
```

log10(value)

Десятичный логарифм.

Параметр:

- **value** — значение для проверки

Например, десятичный логарифм значения ’5’ будет равен ’0.69897000433’.

Пример:

```
log10(last(/host/key))
```

max(<value1>,<value2>,...)

Наибольшее значение из значений, на которые ссылаются.

Параметр:

- **valueX** — значение, возвращаемое другой функцией, работающей с историей элемента данных.

Пример:

```
max(avg(/host/key,1h),avg(/host2/key2,1h))
```

min(<value1>,<value2>,...)

Наименьшее значение значений ссылочных элементов данных.

Параметр:

- **valueX** — значение, возвращаемое другой функцией, которая работает с историей элемента данных.

Пример:

```
min(avg(/host/key,1h),avg(/host2/key2,1h))
```

mod(value,denominator)

Остаток от деления.

Параметры:

- **value** — проверяемое значение;
- **denominator** — делитель.

Например, остаток от деления значения ’5’ на делитель ’2’ будет равен ’1’.

Пример:

```
mod(last(/host/key),2)
```

pi

Константа Pi (3.14159265358979).

Пример:

```
pi()
```

power(value,power value)

Степень значения.

Параметры:

- **value** — значение для проверки;
- **power value** — используемая степень N.

Например, 3-я степень значения ’2’ будет равна ’8’.

Пример:

```
power(last(/host/key),3)
```

radians(value)

Преобразует значение из градусов в радианы.

Параметр:

- **value** — проверяемое значение

Например, значение ’1’, преобразованное в радианы, будет равно ’0.0174532925’.

Пример:

```
radians(last(/host/key))
```

rand

Возвращает случайное целочисленное значение. Псевдослучайное число, сгенерированное с использованием времени в качестве начального значения (достаточно для математических целей, но не для криптографии).

Пример:

```
rand()
```

round(value,decimal places)

Округляет значение до указанного количества десятичных знаков.

Параметры:

- **value** — значение для проверки;
- **decimal places** — укажите количество десятичных знаков для округления (также возможно 0).

Например, значение ’2.5482’, округленное до 2 десятичных знаков, будет равно ’2.55’.

Пример:

```
round(last(/host/key),2)
```

signum(value)

Возвращает ’-1’, если значение отрицательное, ’0’, если значение равно нулю, и ’1’, если значение положительное.

Параметр:

- **value** — значение для проверки.

Пример:

```
signum(last(/host/key))
```

sin(value)

Синус значения, где значение — это угол, выраженный в радианах.

Параметр:

- **value** — значение для проверки

Например, синус значения ’1’ будет равен ’0.8414709848’.

Пример:

```
sin(last(/host/key))
```

sinh(value)

Гиперболический синус значения.

Параметр:

- **value** — значение для проверки

Например, гиперболический синус значения ’1’ будет равен ’1.17520119364’.

Пример:

```
sinh(last(/host/key))
```

sqrt(value)

Квадратный корень значения.
 Эта функция завершится с ошибкой при отрицательном значении.

Параметр:

- **value** — значение для проверки

Например, квадратный корень значения ’3.5’ будет равен ’1.87082869339’.

Пример:

```
sqrt(last(/host/key))
```

sum(<value1>,<value2>,...)

Сумма значений указанных элементов данных.

Параметр:

- **valueX** — значение, возвращаемое другой функцией, которая работает с историей элемента данных.

Пример:

```
sum(avg(/host/key,1h),avg(/host2/key2,1h))
```

tan(value)

Тангенс значения.

Параметр:

- **value** — значение для проверки

Например, тангенс значения ’1’ будет равен ’1.55740772465’.

Пример:

```
tan(last(/host/key))
```

truncate(value,decimal places)

Усечь значение до указанного количества десятичных знаков.

Параметры:

- **value** — значение для проверки;
- **decimal places** — укажите количество десятичных знаков для усечения (также возможно 0).

Например, значение ’2.5482’, усечённое до 2 десятичных знаков, будет равно ’2.54’.

Пример:

```
truncate(last(/host/key),2)
```

См. все поддерживаемые функции.

7 Функции операторов

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
between	Проверка, принадлежит ли значение заданному диапазону.
in	Проверка, равно ли значение хотя бы одному из перечисленных значений.

Подробности о функциях

Некоторые общие замечания о параметрах функций:

- Параметры функций разделяются запятой
- В качестве параметров допускаются выражения

between(value,min,max)

Проверяет, принадлежит ли значение заданному диапазону.
 Поддерживаемые типы значений: *Integer*, *Float*.
 Возвращает: 1 — в диапазоне; 0 — в противном случае.

Параметры:

- **value** — значение для проверки;
- **min** — минимальное значение;
- **max** — максимальное значение.

Пример:

```
between(last(/host/key),1,10)=1 #     ,                    1   10
```

in(value,value1,value2,...valueN)

Проверяет, равно ли значение хотя бы одному из перечисленных значений.
 Поддерживаемые типы значений: *Integer*, *Float*, *Character*, *Text*, *Log*.
 Возвращает: 1 — если равно; 0 — в противном случае.

Параметры:

- **value** — проверяемое значение;
- **valueX** — перечисленные значения (строковые значения должны быть заключены в двойные кавычки).

Значение сравнивается с перечисленными значениями как число, если все эти значения могут быть преобразованы в числовой тип; в противном случае сравнение выполняется как строк.

Пример:

```
in(last(/host/key),5,10)=1 #     ,                   5    10
in("text",last(/host/key),last(/host/key,#2))=1 #     ,     "text"                 2
```

См. все поддерживаемые функции.

8 Прогнозирующие функции

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.

Функция	Описание
forecast	Будущее значение, максимум, минимум, дельта или среднее элемента данных.
timeleft	Время в секундах, необходимое элементу данных для достижения указанного порога.

Общие параметры

- `/host/key` — это общий обязательный первый параметр для функций, ссылающихся на историю элемента данных узла сети
- `(sec|#num)<:time shift>` — это 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** (необязательно) позволяет сдвинуть точку вычисления назад во времени. Подробнее о задании time shift см. в разделе more details.

Подробности функций

Некоторые общие замечания о параметрах функций:

- Параметры функций разделяются запятой
- Необязательные параметры функций (или части параметров) обозначаются < >
- Специфические для функции параметры описаны для каждой функции
- 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, максимум, минимум, дельта или среднее элемента данных.
 Поддерживаемые типы значений: *Float*, *Integer*.

Параметры:

- См. общие параметры;
- **time** — горизонт прогнозирования в секундах (можно использовать суффиксы времени); поддерживаются отрицательные значения;
- **fit** (необязательно; должно быть заключено в двойные кавычки) — функция, используемая для аппроксимации исторических данных. Поддерживаемые 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** (необязательно; должно быть заключено в двойные кавычки) — требуемый результат. Поддерживаемые режимы:
value — значение (по умолчанию)
max — максимум
min — минимум
delta — *max-min*
avg — среднее
Обратите внимание, что *value* оценивает значение элемента данных в момент `now + time`; *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, возвращаемое значение обрезается до 1.7976931348623158E+308 или -1.7976931348623158E+308 соответственно;
- Становится неподдерживаемой только при неправильном использовании в выражении (неверный тип элемента данных, недопустимые параметры), в противном случае в случае ошибок возвращает -1;
- См. также дополнительную информацию о предиктивных функциях 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)<:сдвиг времени>,threshold,<fit>)

Время в секундах, необходимое элементу данных для достижения указанного порогового значения.
 Поддерживаемые типы значений: *Float*, *Integer*.

Параметры:

- См. общие параметры;
- **threshold** — значение, которое необходимо достичь (можно использовать суффиксы единиц);
- **fit** (необязательно; должно быть заключено в двойные кавычки) — см. forecast().

Комментарии:

- Если возвращаемое значение больше 1.7976931348623158E+308, оно усекается до 1.7976931348623158E+308;
- Возвращает 1.7976931348623158E+308, если пороговое значение не может быть достигнуто;
- Становится неподдерживаемой только при неправильном использовании в выражении (неверный тип элемента данных, недопустимые параметры), в противном случае при ошибках возвращает -1;
- См. также дополнительную информацию о функциях прогнозирующих 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` задаёт, насколько далеко в будущее Zabbix должен экстраполировать зависимости, обнаруженные в исторических данных. Независимо от того, используете ли 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 расположен 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 это пересечение уже произошло в прошлом, и поэтому переключается на следующее пересечение с уровнем `threshold`, если оно есть. 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	$x = a + b*\log(t)$

`fit`	x = f(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 к данным указанную функцию. 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 между текущим моментом и этим моментом в будущем. Для некоторых вариантов `fit` (например, *polynomial*) простое значение из будущего может вводить в заблуждение.

`mode`	Результат **forecast**
value	f(now + `time`)
max	$\max_{\text{now <= t <= now + time}}$ f(t)
min	$\min_{\text{now <= t <= now + time}}$ 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)`

9 Строковые функции

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.

Функция	Описание
ascii	ASCII-код крайнего левого символа значения.
bitlength	Длина значения в битах.
bytelength	Длина значения в байтах.
char	Возвращает символ, интерпретируя значение как ASCII-код.
concat	Строка, полученная в результате объединения значений ссылочных элементов данных или константных значений.
insert	Вставляет указанные символы или пробелы в строку символов, начиная с указанной позиции в строке.
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 ‘start’.
repeat	Повторяет строку.
replace	Находит шаблон в значении и заменяет его на replacement.
right	Возвращает крайние правые символы значения.
rtrim	Удаляет указанные символы из конца строки.
trim	Удаляет указанные символы из начала и конца строки.
xmlxpath	Возвращает результат XML XPath.

Подробности о функциях

Некоторые общие замечания о параметрах функций:

- Параметры функций разделяются запятой
- В качестве параметров допускаются выражения
- Строковые параметры должны быть заключены в двойные кавычки; в противном случае они могут быть неверно интерпретированы

- Необязательные параметры функций (или части параметров) обозначаются < >

ascii(value)

ASCII-код крайнего слева символа значения.
 Поддерживаемые типы значений: *String*, *Text*, *Log*.

Параметр:

- **value** — значение для проверки

Например, значение вида 'Abc' вернет '65' (ASCII-код для 'A').

Пример:

```
ascii(last(/host/key))
```

bitlength(value)

Длина value в битах.
 Поддерживаемые типы значений: *String*, *Text*, *Log*, *Integer*.

Параметр:

- **value** — проверяемое значение

Пример:

```
bitlength(last(/host/key))
```

bytelength(value)

Длина value в байтах.
 Поддерживаемые типы значений: *String*, *Text*, *Log*, *Integer*.

Параметр:

- **value** — проверяемое значение

Пример:

```
bytelength(last(/host/key))
```

char(value)

Возвращает символ, интерпретируя значение как ASCII-код.
 Поддерживаемые типы значений: *Integer*.

Параметр:

- **value** — значение для проверки

Значение должно находиться в диапазоне 0-255. Например, значение '65' (интерпретируемое как ASCII-код) вернёт 'A'.

Пример:

```
char(last(/host/key))
```

concat(<value1>,<value2>,...)

Строка, полученная в результате объединения значений ссылочных элементов данных или константных значений.
 Поддерживаемые типы значений: *String*, *Text*, *Log*, *Float*, *Integer*.

Параметр:

- **valueX** — значение, возвращаемое одной из функций истории, или константное значение (строка, целое число или число с плавающей точкой). Должно содержать как минимум два параметра.

Например, значение 'Zab', объединённое с 'bix' (константной строкой), вернёт 'Zabbix'.

Примеры:

```
concat(last(/host/key),"bix")
concat("1 min: ",last(/host/system.cpu.load[all,avg1]),", 15 min: ",last(/host/system.cpu.load[all,avg15]))
```

insert(value,start,length,replacement)

Вставляет указанные символы или пробелы в строку символов, начиная с указанной позиции в строке.
 Поддерживаемые типы значений: *String*, *Text*, *Log*.

Параметры:

- **value** — значение для проверки;
- **start** — начальная позиция;
- **length** — позиции для замены;

- **replacement** — строка замены.

Например, значение вида ’Zabbbix’ будет заменено на ’Zabbix’, если ’bb’ (начальная позиция 3, количество заменяемых позиций 2) заменить на ’b’.

Пример:

```
insert(last(/host/key),3,2,"b")
```

jsonpath(value,path,<default>)

Возвращает результат JSONPath.
 Поддерживаемые типы значений: *String*, *Text*, *Log*.

Параметры:

- **value** — проверяемое значение;
- **path** — путь (должен быть заключён в кавычки);
- **default** — необязательное резервное значение, если запрос JSONPath не возвращает данных. Обратите внимание, что при других ошибках возвращается сбой (например, ”unsupported construct”).

Пример:

```
jsonpath(last(/host/proc.get[zabbix_agentd,,,summary]),"$..size")
```

left(value,count)

Возвращает крайние слева символы значения.
 Поддерживаемые типы значений: *String*, *Text*, *Log*.

Параметры:

- **value** — значение для проверки;
- **count** — количество возвращаемых символов.

Например, можно вернуть ’Zab’ из ’Zabbix’, указав 3 крайних слева символа для возврата. См. также right().

Пример:

```
left(last(/host/key),3) #
```

length(value)

Длина value в символах.
 Поддерживаемые типы значений: *String*, *Text*, *Log*.

Параметр:

- **value** — значение для проверки.

Примеры:

```
length(last(/host/key)) #
length(last(/host/key,#3)) #
length(last(/host/key,#1:now-1d)) #
```

ltrim(value,<chars>)

Удаляет указанные символы из начала строки.
 Поддерживаемые типы значений: *String*, *Text*, *Log*.

Параметры:

- **value** — значение для проверки;
- **chars** (необязательно) — указывает символы для удаления.

По умолчанию из начала строки удаляются пробельные символы (если необязательные символы не указаны). См. также: rtrim(), trim().

Примеры:

```
ltrim(last(/host/key)) #
ltrim(last(/host/key),"Z") #            'Z'
ltrim(last(/host/key)," Z") #                   'Z'
```

mid(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, можно вернуть 'abbi' из значения 'Zabbix', если начальная позиция равна 2, а количество возвращаемых позиций — 4.

Пример:

```
mid(last(/host/key),2,4)="abbi"
```

repeat(value,count)

Повторяет строку.
 Поддерживаемые типы значений: *String*, *Text*, *Log*.

Параметры:

- **value** — значение для проверки;
- **count** — количество повторений.

Пример:

```
repeat(last(/host/key),2) #
```

replace(value,pattern,replacement)

Найдите шаблон в значении и замените его на replacement. Все вхождения шаблона будут заменены.
 Поддерживаемые типы значений: *String*, *Text*, *Log*.

Параметры:

- **value** - значение для проверки;
- **pattern** - шаблон для поиска;
- **replacement** - строка, на которую заменяется шаблон.

Пример:

```
replace(last(/host/key),"ibb","abb") #        'ibb'    'abb'
```

right(value,count)

Возвращает крайние справа символы значения.
 Поддерживаемые типы значений: *String*, *Text*, *Log*.

Параметры:

- **value** — значение для проверки;
- **count** — количество символов для возврата.

Например, можно вернуть 'bix' из 'Zabbix', указав 3 крайних справа символа для возврата. См. также left().

Пример:

```
right(last(/host/key),3) #
```

rtrim(value,<chars>)

Удаляет указанные символы с конца строки.
 Поддерживаемые типы значений: *String*, *Text*, *Log*.

Параметры:

- **value** — значение для проверки;
- **chars** (необязательно) — указывает символы для удаления.

По умолчанию справа удаляются пробельные символы (если необязательные символы не указаны). См. также: ltrim(), trim().

Примеры:

```
rtrim(last(/host/key)) #
rtrim(last(/host/key),"x") #            'x'
rtrim(last(/host/key),"x ") #            'x'
```

trim(value,<chars>)

Удаляет указанные символы из начала и конца строки.
 Поддерживаемые типы значений: *String*, *Text*, *Log*.

Параметры:

- **value** — значение для проверки;
- **chars** (необязательно) — указывает символы для удаления.

По умолчанию пробельные символы удаляются с обеих сторон (если необязательные символы не указаны). См. также: ltrim(), rtrim().

Примеры:

```
trim(last(/host/key)) #
trim(last(/host/key),"_") #     '_'
```

xmlxpath(value,path,<default>)

Возвращает результат XML XPath.
 Поддерживаемые типы значений: *String*, *Text*, *Log*.

Параметры:

- **value** — значение для проверки;
- **path** — путь (должен быть заключён в кавычки);
- **default** — необязательное резервное значение, если запрос XML XPath возвращает пустой набор узлов. Оно не будет возвращено, если пустой результат не является набором узлов (то есть пустой строкой). При других ошибках возвращается ошибка (например, ”invalid expression”).

Пример:

```
xmlxpath(last(/host/xml_result),"/response/error/status")
```

См. все поддерживаемые функции.

4 Рекомендации по использованию 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 более длительный период, используя такие функции, как **avg**, **min** и **max**.
2. Рассмотрите возможность использования функции **percentile** (установите её на 95% или 5%) в триггерах, если хотите избежать оповещений из-за случайных всплесков и падений.
3. Используйте гистерезис, чтобы избежать дребезга триггера — частых изменений состояния триггера (OK ↔ Problem). Непрерывность проблемного состояния можно определить, добавив выражение восстановления (отдельное условие для устранения проблемы).
4. Используйте зависимости триггеров, чтобы избежать оповещений, не связанных с первопричиной.
5. Используйте важность триггера, чтобы оповещать только о более серьёзных проблемах.
6. Определите 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").

Важность триггера

Важность триггера можно использовать, чтобы избежать лишнего шума оповещений по менее серьезным проблемам.

См. все рекомендации.

Обзор

Важность представляет относительную значимость триггера.



Назначение важности полезно не только для визуального представления, но и для предотвращения уведомлений о незначительных проблемах.

Важности триггеров используются для:

- Визуального представления (отдельный цвет для каждой созданной проблемы на основе исходной важности триггера)
- Ограничения уведомлений на основе важности (см. условия действий)
- Пользовательских способов оповещения — разные способы оповещения (каналы уведомлений) для разных уровней важности (например, SMS для триггеров с важностью *Высокая* и *Катастрофа*, а Email — для триггеров других уровней важности)
- Звука в глобальных оповещениях — разные звуковые сигналы для разных уровней важности

По умолчанию поддерживаются следующие уровни важности триггеров:

Важность	Цвет	Описание
Не классифицировано	Серый	Может использоваться, когда уровень важности события неизвестен, не определен, не входит в обычную область мониторинга и т. д., например, во время начальной настройки, как заполнитель для будущей оценки или как часть процесса интеграции.
Информация	Светло-синий	Может использоваться для информационных событий, которые не требуют немедленного внимания, но все же могут предоставлять ценную информацию.
Предупреждение	Желтый	Может использоваться для указания на потенциальную проблему, которая может потребовать расследования или действий, но не является критической.
Средняя	Оранжевый	Может использоваться для указания на значительную проблему, которую следует устранить достаточно скоро, чтобы предотвратить дальнейшие проблемы.
Высокая	Светло-красный	Может использоваться для указания на критические проблемы, требующие немедленного внимания во избежание значительных нарушений.
Катастрофа	Красный	Может использоваться для указания на серьезный инцидент, требующий немедленных действий для предотвращения, например, отказов системы или потери данных.

Названия уровней важности триггеров и их цвета можно **настроить** в разделе *Администрирование* > *Общие* > *Параметры отображения триггеров*.
Цвета являются общими 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.

Зависимости триггеров

Зависимости триггеров можно использовать, чтобы избежать оповещений, не связанных 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 на самом деле причиной был только маршрутизатор.

Именно в таких случаях может быть полезна зависимость между узлами сети. Если зависимость настроена, уведомления для зависимых объектов могут быть удержаны, и будет отправлено только уведомление о корневой проблеме.

Хотя Zabbix не поддерживает зависимости между узлами сети напрямую, их можно определить другим, более гибким способом — с помощью зависимостей триггеров. Триггер может зависеть от одного или нескольких других триггеров.

Итак, 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 будет изменять свое состояние, пока триггер, от которого он зависит, находится в состоянии 'PROBLEM' — и, следовательно, никакие зависимые действия выполняться не будут, а уведомления отправляться не будут.

Если и сервер, и маршрутизатор недоступны, и зависимость настроена, Zabbix не будет выполнять действия для зависимого триггера.

Пока родительский триггер находится в состоянии PROBLEM, его зависимые триггеры могут сообщать значения, которым нельзя доверять. Поэтому зависимые триггеры не будут переоцениваться до тех пор, пока родительский триггер (маршрутизатор в приведенном выше примере):

- не вернется из состояния 'PROBLEM' в состояние 'OK';
- не изменит свое состояние с 'PROBLEM' на 'UNKNOWN';
- не будет закрыт вручную, с помощью корреляции или с помощью функций date and time и/или nodata();
- не будет разрешен значением элемента данных, не участвующего в зависимом триггере;
- не будет отключен, либо пока не будет отключен его элемент данных или узел сети элемента данных

Во всех упомянутых выше случаях зависимый триггер (сервер) будет переоценен только при получении для него новой метрики. Это означает, что зависимый триггер может обновиться не сразу.

Также:

- Зависимость триггера может быть добавлена от любого триггера 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 может быть привязан к узлу сети (или другому шаблону) только вместе с шаблоном B, тогда как шаблон B может быть привязан к узлу сети (или другому шаблону) отдельно.
- Зависимость триггера может быть добавлена от триггера шаблона к триггеру узла сети. В этом случае привязка такого шаблона к узлу сети создаст триггер узла сети, который зависит от того же шаблона триггера, от которого зависел исходный триггер. Это позволяет, например, иметь шаблон, в котором некоторые триггеры зависят от

триггеров маршрутизатора (узла сети). Все узлы сети, привязанные к этому шаблону, будут зависеть от этого конкретного маршрутизатора.

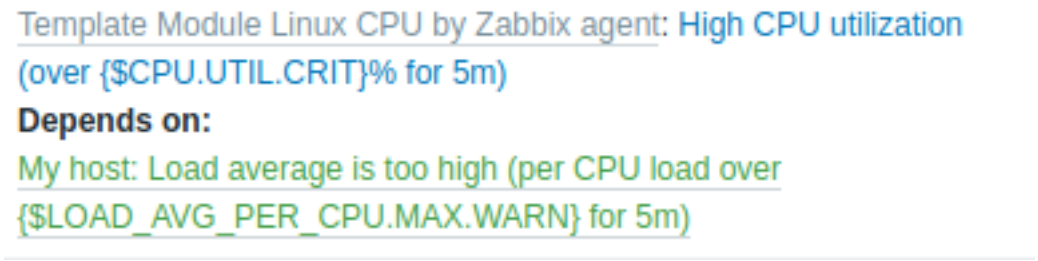
- Зависимость триггера не может быть добавлена от триггера узла сети к триггеру шаблона.
- Зависимость триггера может быть добавлена от прототипа триггера к другому прототипу триггера (в пределах одного и того же правила обнаружения низкого уровня) или к реальному триггеру. Прототип триггера не может 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.

Настройка

Чтобы определить зависимость, откройте вкладку «Зависимости» в форме настройки триггера. Нажмите *Добавить* в блоке «Зависимости» и выберите один или несколько триггеров, от которых будет зависеть данный триггер.



Нажмите *Обновить*. Теперь в списке у триггера будет отображаться индикатор его зависимости.



Пример нескольких зависимостей

Например, узел сети находится за Router2, а Router2 находится за Router1.

```
Zabbix - Router1 - Router2 - Host
```

Если Router1 недоступен, то, очевидно, узел сети и Router2 также недоступны, однако получение трёх уведомлений о том, что узел сети, 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 , 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 узла сети не будет обновлён.

4 События

Обзор

В Zabbix генерируется несколько типов событий:

- события триггеров — каждый раз, когда триггер изменяет свой статус (*OK→PROBLEM→OK*)
- события сервисов — каждый раз, когда сервис изменяет свой статус (*OK→PROBLEM→OK*)
- события обнаружения — когда обнаруживаются узлы сети или сервисы
- события авторегистрации — когда активные агенты автоматически регистрируются сервером
- внутренние события — когда элемент данных/правило низкоуровневого обнаружения становится неподдерживаемым или триггер переходит в неизвестное состояние

События имеют временные метки и могут служить основой для таких действий, как отправка уведомлений по электронной почте и т. д.

Чтобы просмотреть сведения о событиях в веб-интерфейсе, перейдите в *Мониторинг → Проблемы*. Там можно нажать на дату и время события, чтобы просмотреть подробности события.

Дополнительная информация доступна здесь:

- события триггеров
- другие источники событий

1 Генерация событий на триггеры

Обзор

Изменение состояния триггера является наиболее частым и наиболее важным источником событий. Каждый раз, когда триггер изменяет своё состояние, создаётся событие. Событие содержит сведения об изменении состояния триггера — когда это произошло и каково новое состояние.

Триггеры создают два типа событий — `Problem` и `OK`.

События проблемы

Событие проблемы создается:

- когда выражение триггера вычисляется как ИСТИНА, если триггер был в состоянии OK;
- каждый раз выражение триггера вычисляется как ИСТИНА, если у триггера активировано множественная генерация событий проблема.

События OK

Событие OK закрывает связанные событие(-я) о проблеме и может быть создано тремя компонентами:

- триггеры - на основании настроек ’Генерация OK событий’ и ’OK событие закрывает’;
- корреляция событий
- менеджер задач – при закрытии вручную события

Триггеры

У триггеров имеется настройка ’Формирование OK событий’, которая определяет, как генерируются OK события:

- *Выражение* - событие OK генерируется триггеру в состоянии проблема, когда выражение этого триггера вычисляется как ЛОЖЬ. Это самая простая настройка, активированная по умолчанию.
- *Выражение восстановления* - событие OK генерируется триггеру в состоянии проблема, когда выражение этого триггера вычисляется как ЛОЖЬ и выражение восстановления вычисляется как ИСТИНА. Эту опцию можно использовать, если критерий восстановления триггера отличается от критерия проблемы.
- *Нет* - событие OK никогда не генерируется. Эту опцию можно использовать в сочетании с множественной генерацией проблем для простой отправки оповещений, когда что-то происходит.

Кроме того, у триггеров имеется настройка ’OK событие закрывает’, которая определяет какие события о проблемах должны быть закрыты:

- *Все проблемы* - событие OK закроет все открытые проблемы, которые созданы этим триггером
- *Все проблемы, если значение тега совпадает* - событие OK закрывает все открытые проблемы, которые созданы этим триггером, и имеющие по крайней мере одно совпадающее значение тега. Тег задается настройкой триггера ’Теги для совпадения’. Если нет никаких событий о проблемах для закрытия, тогда событие OK не генерируется. Эту опцию часто называют корреляцией событий на уровне триггеров.

Корреляция событий

Корреляция событий (также называемая глобальной корреляцией событий) является способом настройки пользовательских правил закрытия событий (в результате чего генерируются OK события).

Правила определяют каким образом новые события о проблемах соотносятся с существующими событиями о проблемах и позволяют закрыть новое событие или совпадающие события при помощи генерации OK событий.

Однако, корреляцию событий необходимо настраивать очень осторожно, так как она может негативно повлиять на производительность обработки событий или, при неправильной настройке, закрыть больше событий, чем требуется (в худшем случае могут быть закрыты вообще все события о проблемах). Несколько советов по настройке:

1. всегда старайтесь уменьшить масштаб корреляции, указав уникальный тег для контрольного события (парное событие со старыми событиями) и использовав условие корреляции ’тег нового события’
2. не забывайте добавлять условие на основе старого события при использовании операции ’закрыть старые события’, в противном случае все существующие проблемы могут быть закрыты

3. избегайте использования распространенных имен тегов, которые используются различными настройками корреляции

Диспетчер задач

Если для триггера включена настройка «Разрешить ручное закрытие», то события проблем, сгенерированные этим триггером, можно закрывать вручную. Это выполняется в веб-интерфейсе при обновлении проблемы. Событие не закрывается напрямую — вместо этого создаётся задача «закрыть событие», которая вскоре обрабатывается диспетчером задач. Диспетчер задач создаст соответствующее событие OK, и событие проблемы будет закрыто.

2 Другие источники событий

События сервиса

События сервиса создаются, только 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 на обнаружения

Zabbix периодически сканирует диапазоны IP адресов заданные в правилах сетевого обнаружения. Частота этой проверки настраивается отдельно по каждому правилу. После того, как узел сети или сервис обнаружен, генерируется событие (или несколько событий) на обнаружение.

Zabbix генерирует следующие события:

Событие	Когда генерируется
Сервис доступен	Каждый раз когда Zabbix обнаруживает активный сервис.
Сервис недоступен	Каждый раз когда Zabbix не может обнаружить сервис.
Узел сети доступен	Если хотя бы один сервис доступен для IP.
Узел сети недоступен	Если все сервисы не доступны.
Сервис обнаружен	Если сервис стал доступен после его недоступности или обнаружен впервые.
Сервис потерян	Если сервис потерян после того как был доступен.
Узел сети обнаружен	Если узел сети стал доступен после его недоступности или обнаружен впервые.
Узел сети потерян	Если узел сети потерян после того как был доступен.

События авторегистрации активного агента

Авторегистрация активного агента создает события в Zabbix.

Если настроено, событие авторегистрации 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.

Дополнительную информацию см. на странице авторегистрация 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 низкоуровневого обнаружения меняет состояние с «normal» на «unsupported»
- правило низкоуровневого обнаружения меняет состояние с «unsupported» на «normal»
- триггер меняет состояние с «normal» на «unknown»
- триггер меняет состояние с «unknown» на «normal»

Цель введения внутренних событий — позволить пользователям получать уведомления, когда происходит любое внутреннее событие, например, если элемент данных становится неподдерживаемым и перестаёт собирать данные.

Внутренние события создаются только тогда, когда включены внутренние действия для этих событий. Чтобы остановить генерацию внутренних событий (например, для случаев, когда элементы данных становятся неподдерживаемыми), отключите все действия для внутренних событий в разделе Оповещения → Действия → Внутренние действия.

Note:
Если внутренние действия отключены, то, пока объект находится в состоянии «unsupported», событие восстановления для этого объекта всё равно будет создано.

Если внутренние действия включены, то, пока объект находится в состоянии «unsupported», событие восстановления для этого объекта будет создано, даже если для объекта не было создано «problem event».

См. также: Получение уведомлений о неподдерживаемых элементах данных

3 Закрытие проблем вручную

Обзор

Хотя обычно события проблем устраняются автоматически, когда состояние триггера меняется с `Problem` на `OK`, могут быть 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 необходимо настроить для оптимальной производительности. В этом случае о проблеме сообщают администраторам Linux, они устраняют её, а затем закрывают проблему вручную.

Проблемы можно закрывать вручную только для триггеров, у которых включена опция *Allow manual close*.

Когда проблема «закрывается вручную», Zabbix создаёт 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, разрешённых 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}`.

Проблема, закрытая вручную, не означает, что соответствующий триггер больше никогда не перейдёт в состояние *Problem*. Выражение триггера вычисляется повторно и может снова привести к проблеме:

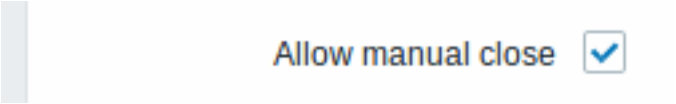
- Когда поступают новые данные для любого элемента данных, включённого в выражение триггера (обратите внимание, что значения, отброшенные на этапе предобработки с ограничением частоты, не считаются полученными и не вызовут повторное вычисление выражения триггера);
- Когда в выражении используются функции date and time и/или nodata().

Настройка

Для закрытия проблем вручную необходимо выполнить два шага .

Настройка триггера

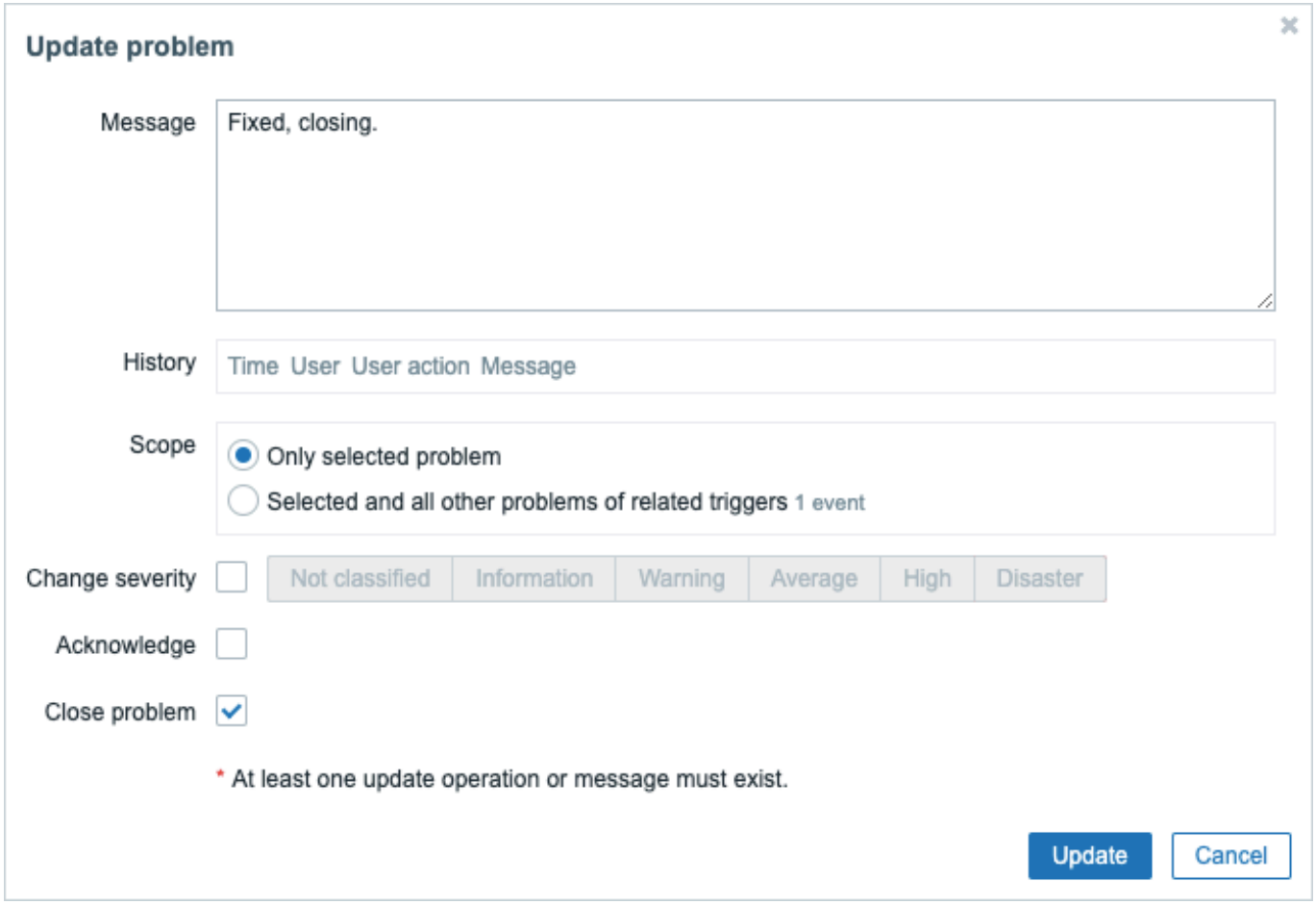
В настройке триггера включите опцию *Разрешить закрывать вручную*.



Окно обновления проблемы

Если для триггера с флагом *Manual close* возникает проблема, вы можете открыть всплывающее окно обновления проблемы этой проблемы и закрыть проблему вручную.

Чтобы закрыть проблему, установите флажок *Close problem* в форме и нажмите *Update*.



Все обязательные для заполнения поля отмечены красной звёздочкой.

Запрос обрабатывается сервером Zabbix. Обычно закрытие проблемы занимает несколько секунд. Во время этого процесса в разделе *Monitoring* > *Problems* в качестве статуса проблемы отображается *CLOSING*.

Подтверждение

Можно проверить, что проблема была закрыта вручную:

- в сведениях о событии, доступных через *Мониторинг* > *Проблемы*;
- с помощью макроса `{EVENT.UPDATE.HISTORY}` в сообщениях уведомлений, который предоставит эту информацию.

5 Корреляция событий

Обзор

Корреляция событий позволяет сопоставлять события о проблемах с их решениями очень точным и гибким способом.

Корреляцию событий можно задавать:

- на уровне триггера - один триггер можно использовать, чтобы связать отдельные проблемы с их решением
- глобально - проблемы можно соотнести с их решением с другого триггера/метода опроса с использованием глобальных правил корреляции

1 Корреляция событий на основе триггеров

Обзор

Корреляция событий на основе триггеров позволяет сопоставлять отдельные проблемы, сообщаемые одним триггером.

Хотя в общем случае событие OK может закрыть все события проблем, созданные одним триггером, бывают ситуации, когда требуется более детальный подход. Например, при мониторинге лог-файлов может потребоваться обнаруживать определённые проблемы в лог-файле 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 логов, обработки trap-сообщений и т. д.

В Zabbix можно связывать события проблем на основе тегов. Теги используются для извлечения значений и создания идентификаторов для событий проблем. Благодаря этому проблемы также можно закрывать по отдельности на основе совпадающего тега.

Иными словами, один и тот же триггер может создавать отдельные события, идентифицируемые тегом события. Поэтому события проблем можно определять по одному и закрывать отдельно на основе идентификации по тегу события.

Как это работает

При мониторинге журналов вы можете столкнуться со строками, подобными этим:

```
1: Service 1 stopped
2: Service 2 stopped
3: Service 1 was restarted
4: Service 2 was restarted
```

Идея корреляции событий заключается в том, чтобы сопоставить событие проблемы из Строки1 с устранением из Строки3, а событие проблемы из Строки2 с устранением из Строки4 и закрыть эти проблемы по одной:

```
1: Service 1 stopped
3: Service 1 was restarted #           1

2: Service 2 stopped
4: Service 2 was restarted #           2
```

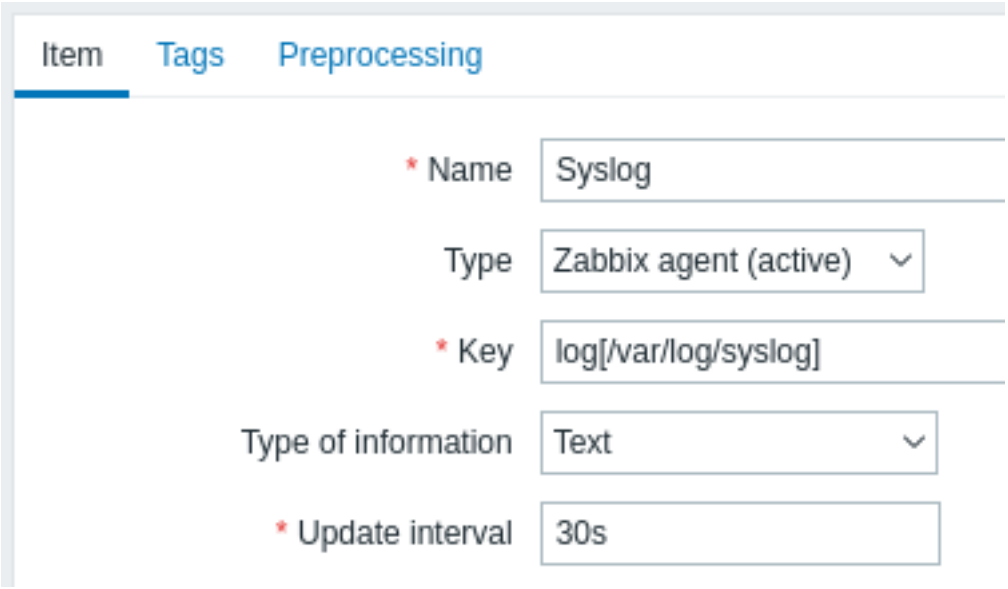
Для этого необходимо пометить эти связанные события, например, тегами ”Service 1” и ”Service 2”. Это можно сделать, применив регулярное выражение к строке журнала, чтобы извлечь значение тега. Затем, когда создаются события, им назначаются теги ”Service 1” и ”Service 2” соответственно, и проблему можно сопоставить с устранением.

Настройка

Элемент данных

Для начала вы можете настроить элемент данных, который мониторит файл журнала, например:

`log[/var/log/syslog]`



Когда элемент данных будет настроен, подождите минуту, пока изменения конфигурации вступят в силу и, затем перейдите в Последние данные, чтобы убедиться, что элемент данных начал собирать данные.

Триггер

Когда элемент данных работает, необходимо настроить триггер. Важно определить, какие записи в журнале заслуживают внимания. Например, следующее выражение триггера будет искать строку вида ’Stopping’, чтобы сигнализировать о потенциальных проблемах:

```
find(/My host/log[/var/log/syslog],,"regexp","Stopping")=1
```

Attention:
Чтобы каждая строка, содержащая ”Stopping”, рассматривалась как проблема, также установите для параметра *Режим генерации событий проблемы* в настройках триггера значение ’Multiple’.

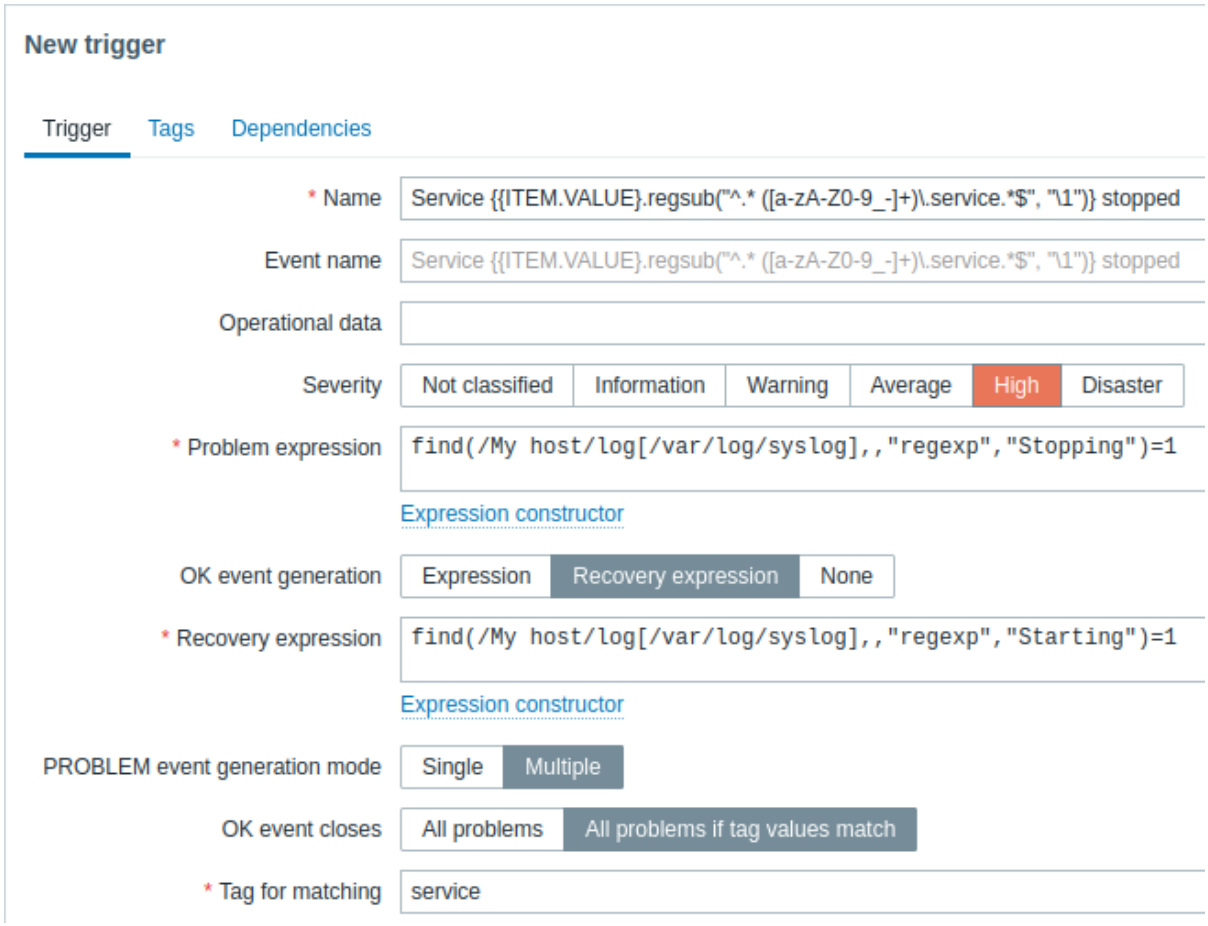
Затем задайте выражение восстановления. Следующее выражение восстановления закроет все проблемы, если будет найдена строка журнала, содержащая строку ”Starting”:

```
find(/My host/log[/var/log/syslog],,"regexp","Starting")=1
```

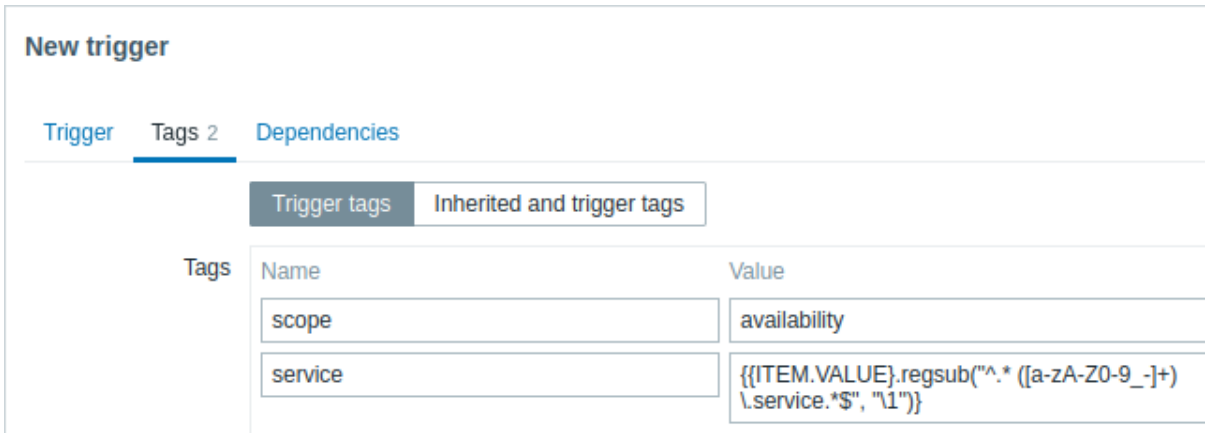
Поскольку это нежелательно, важно каким-либо образом убедиться, что закрываются соответствующие исходные проблемы, а не просто все проблемы. Здесь могут помочь теги.

Проблемы и их устранение можно сопоставлять, указав тег в настройках триггера. Необходимо выполнить следующие настройки:

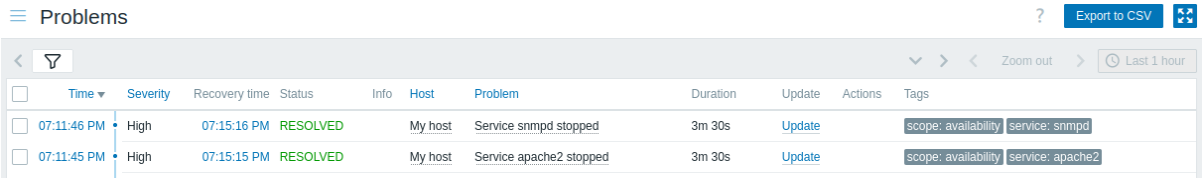
- *Режим генерации событий проблемы*: Multiple
- *OK-событие закрывает*: Все проблемы, если значения тегов совпадают
- Укажите имя тега для сопоставления событий



- настройте теги для извлечения значений тегов из строк журнала



Если настройка выполнена успешно, вы сможете увидеть события проблем, помеченные тегами по приложению и сопоставленные с их устранением, в разделе *Мониторинг → Проблемы*.



Warning:
Поскольку возможна неправильная настройка, при которой для **несвязанных** проблем могут создаваться похожие теги событий, пожалуйста, ознакомьтесь с описанными ниже случаями!

- Индексированные макросы всегда относятся к полю *Выражение* в настройках триггера, а не к *Выражению восстановления*. Например, в событии восстановления **{ITEM.VALUE1}** будет преобразован 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 и том же триггере с разными значениями тегов, применяя отдельные регулярные выражения в значениях тегов для извлечения имен, например, service A и service B, из макроса {ITEM.VALUE} (например, когда форматы сообщений различаются). Однако это может не сработать так, как задумано, если не будет совпадения с регулярными выражениями. Регулярные выражения без совпадений дадут пустые значения тегов, и одного пустого значения тега как в событиях проблемы, так и в OK-событиях 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 некорректно, оно молча заменяется строкой *UNKNOWN*. Если исходное событие проблемы со значением тега *UNKNOWN* было пропущено, впоследствии могут появиться OK-события с тем же значением тега *UNKNOWN*, которые могут закрыть события проблем, 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 являются специфичными, это также может привести к появлению похожих тегов событий для несвязанных проблем.

2 Глобальная корреляция событий

Обзор

Глобальная корреляция событий позволяет охватить все метрики, которые отслеживаются Zabbix, и создавать корреляции.

Можно коррелировать события, созданные совершенно разными триггерами, и применять к ним всем одинаковые операции. Создавая интеллектуальные правила корреляции, вы действительно можете избавить себя от тысяч повторяющихся уведомлений и сосредоточиться на первопричинах проблемы!

Глобальная корреляция событий — это мощный механизм, который позволяет уйти от логики проблем и их устранения, основанной на одном триггере. До сих пор одно событие проблемы создавалось одним триггером, и для устранения проблемы мы зависели от этого же триггера. Мы не могли устранить проблему, созданную одним триггером, с помощью другого триггера. Но с корреляцией событий на основе тегов событий это возможно.

Например, триггер журнала может сообщать о проблемах приложения, тогда как триггер опроса может сообщать, что приложение работает и доступно. Используя теги событий, вы можете пометить триггер журнала тегом *status:down*, а триггер опроса — тегом *status:up*. Затем в глобальном правиле корреляции вы можете 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 одинаковую операцию. Что, если бы мы могли получать только одно сообщение о проблеме на каждую проблему сетевого порта? Не нужно сообщать обо всех. Это также возможно с глобальной корреляцией событий.

Глобальная корреляция событий настраивается в **правилах корреляции**. Правило корреляции определяет, как новые события проблемы сопоставляются с существующими событиями проблемы и что делать в случае совпадения (закрыть

новое событие, закрыть совпавшие старые события, сгенерировав соответствующие события OK). Если проблема закрыта глобальной корреляцией, это отображается в столбце *Info* в разделе *Monitoring* > *Problems*.

Настройка глобальных правил корреляции доступна только пользователям уровня Super Admin.

Attention:
Корреляцию событий необходимо настраивать очень внимательно, так как она может отрицательно повлиять на производительность обработки событий или, при неправильной настройке, закрыть больше событий, чем предполагалось (в худшем случае могут быть закрыты даже все события проблемы).

Чтобы настроить глобальную корреляцию **безопасно**, соблюдайте следующие важные рекомендации:

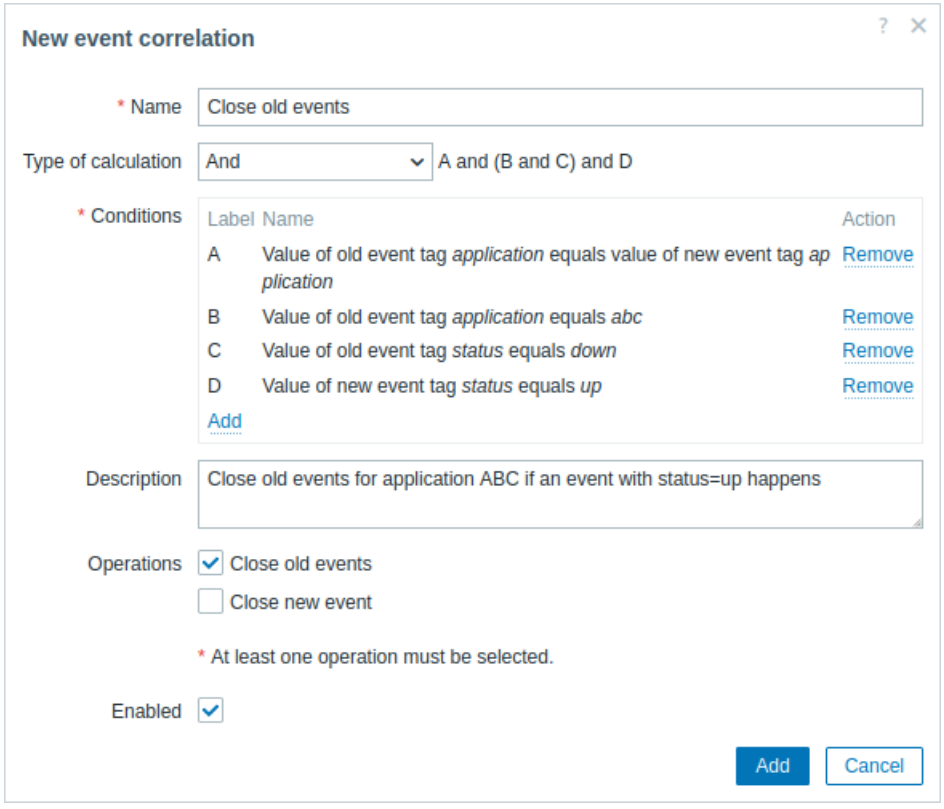
- Сужайте область корреляции. Всегда задавайте уникальный тег для нового события, которое сопоставляется со старыми событиями, и используйте условие корреляции *New event tag name*.
- Добавляйте явное условие для старого события при использовании **Close old events**. Всегда добавляйте как минимум одно условие *Old event* (например, *Old event tag name*, *Old event tag value* или *Event tag pair*) при выборе **Close old events** — в противном случае правило может сопоставить и закрыть несвязанные существующие события проблемы (в худшем случае — все проблемы). Предпочтительно использовать **Event tag pair** для сопоставления значений времени выполнения (host:port, session id и т. п.), а также по возможности дополнительно сужать совпадение по узлу сети или группе узлов сети.
- Избегайте использования общих имен тегов, которые могут в итоге использоваться разными конфигурациями корреляции.
- Ограничивайте количество правил корреляции только действительно необходимыми.

См. также: известные проблемы.

Конфигурация

Чтобы настроить правила корреляции событий глобально:

- Перейдите в *Сбор данных* > *Корреляция событий*
- Нажмите *Создать корреляцию событий* справа (или на имя корреляции, чтобы редактировать существующее правило)
- Введите параметры правила корреляции в форме

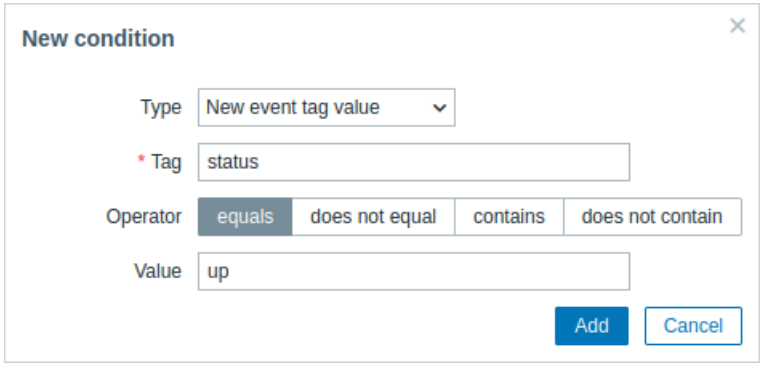


Все обязательные поля ввода отмечены красной звездочкой.

Параметр	Описание
Имя	Уникальное имя правила корреляции.

Параметр	Описание
Тип вычисления	Доступны следующие варианты вычисления условий: **И** - должны выполняться все условия **Или** - достаточно выполнения одного условия **И/Или** - И для разных типов условий и Или для одинаковых типов условий **Пользовательское выражение** - заданная пользователем 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).
Условия	Список условий. Подробнее о настройке условия см. ниже.
Описание	Описание правила корреляции.
Операции	Установите флажок у операции, которую нужно выполнить при корреляции события. Доступны следующие операции: **Закрыть старые события** - закрывать старые события при возникновении нового события. При использовании операции *Закрыть старые события* всегда добавляйте условие на основе старого события, иначе могут быть закрыты все существующие проблемы. **Закрыть новое событие** - закрывать новое событие при его возникновении. **Предупреждение!** Не оставляйте условия старого/нового события пустыми при использовании *Закрыть старые события/Закрыть новое событие*. Если выбрать операцию *Закрыть старые события* без добавления условия, соответствующего старому событию, Zabbix может сопоставить все существующие старые события и закрыть их. Всегда добавляйте явное условие старого события (например, *Имя тега старого события* или *Пара тегов события*) при использовании *Закрыть старые события*. Например, правило, использующее только *Условие нового события* и операцию *Закрыть старые события*, будет сопоставлять все старые события, которые соответствуют (отсутствующим) критериям старого события, фактически закрывая старые проблемы.
Включено	Если установить этот флажок, правило корреляции будет включено.

Чтобы настроить параметры нового условия, нажмите Add в блоке Условия. Откроется всплывающее окно, где можно редактировать параметры условия.



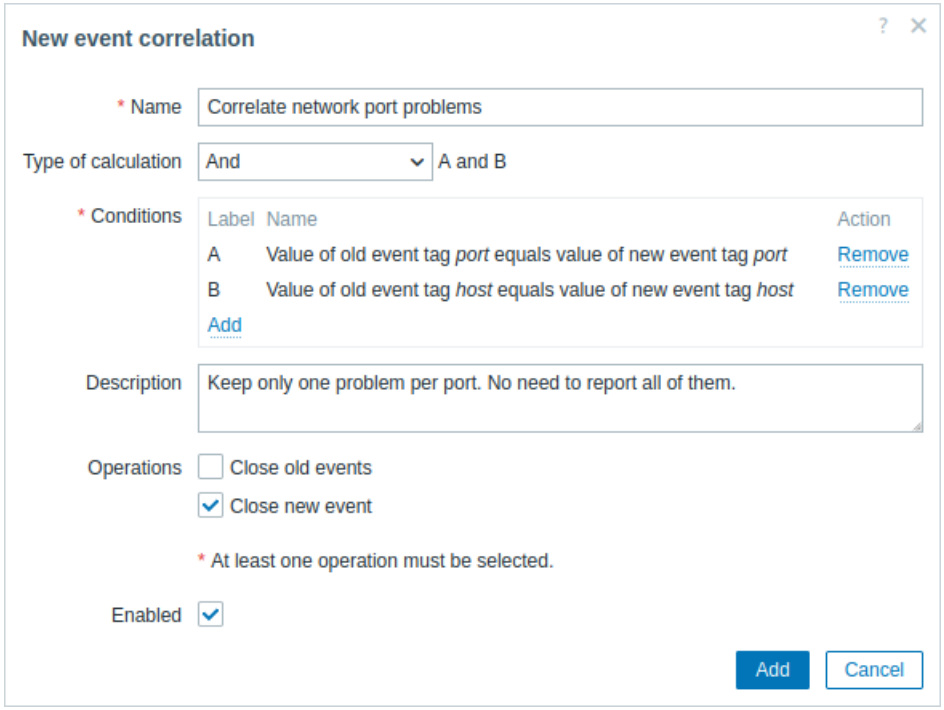
Параметр	Описание
Новое условие	Выберите условие для корреляции событий. *Обратите внимание*, что если не указано условие старого события, могут быть сопоставлены и закрыты все старые события. Аналогично, если не указано условие нового события, могут быть сопоставлены и закрыты все новые события. Доступны следующие условия: **Имя тега старого события** - укажите имя тега старого события для сопоставления. **Имя тега нового события** - укажите имя тега нового события для сопоставления. **Новая группа узлов сети события** - укажите группу узлов сети нового события для сопоставления. **Пара тегов события** - 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 совпадать не обязаны. Этот вариант полезен для сопоставления значений времени выполнения, которые могут быть неизвестны на момент настройки (см. также Пример). **Значение тега старого события** - укажите имя и значение тега старого события для сопоставления, используя следующие операторы: *равно* - имеет значение тега старого события *не равно* - не имеет значения тега старого события *содержит* - содержит строку в значении тега старого события *не содержит* - не содержит строку в значении тега старого события **Значение тега нового события** - укажите имя и значение тега нового события для сопоставления, используя следующие операторы: *равно* - имеет значение тега нового события *не равно* - не имеет значения тега нового события *содержит* - содержит строку в значении тега нового события *не содержит* - не содержит строку в значении тега нового события

Warning:
Поскольку возможна неправильная настройка, если похожие теги событий могут создаваться для **несвязанных** проблем, пожалуйста, ознакомьтесь со случаями, описанными ниже!

- Фактические теги и значения тегов становятся видимыми только при срабатывании триггера. Если используемое регулярное выражение некорректно, оно без уведомления заменяется строкой *UNKNOWN*. Если исходное событие проблемы со значением тега *UNKNOWN* было пропущено, впоследствии могут появиться события OK с тем же значением тега *UNKNOWN*, которые могут закрыть события проблем, которые не должны были быть закрыты.
- Если пользователь использует макрос {ITEM.VALUE} без функций макросов в качестве значения тега, применяется ограничение в 255 символов. Если сообщения журнала длинные, а первые 255 символов не являются специфичными, это также может привести к появлению похожих тегов событий для несвязанных проблем.

Пример

Остановите повторяющиеся события проблем с одного и того же сетевого порта.



Это глобальное правило корреляции будет выполнять корреляцию проблем, если значения тегов *узел сети* и *port* существуют у триггера и совпадают в исходном и новом событии.

Операция будет закрывать новые события проблем на одном и том же сетевом порту, оставляя открытой только исходную проблему.

6 Тегирование

Обзор

Теги состоят из имени тега и значения тега. При назначении тегов сущностям можно использовать только имя или пару имени со значением (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 прототипов столбец *Теги* отображает как обычные, так и унаследованные теги. При наведении указателя мыши на *унаследованный* тег или при щелчке по нему появляется всплывающая JavaScript-подсказка 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 всплывающей подсказки.

Note:
Обратитесь к официальным рекомендациям Zabbix с общими рекомендациями по определению тегов, а также со специальными рекомендациями для шаблонов, элементов данных, триггеров и правил низкоуровневого обнаружения.

Теги имеют несколько назначений, наиболее важное из которых — маркировка событий. Если сущностям назначены теги, любое новое событие, связанное с помеченной сущностью, унаследует её теги. Например:

- при наличии тегов у шаблонов — любая проблема узла сети (созданная триггерами из шаблона) унаследует теги шаблона.
- при наличии тегов у узлов сети — любая проблема узла сети унаследует теги узла сети.
- при наличии тегов у элементов данных/веб-сценариев — любая проблема элемента данных/веб-сценария унаследует теги элемента данных/веб-сценария.
- при наличии тегов у триггеров — любая проблема, созданная триггером, унаследует теги триггера.

Событие проблемы наследует все теги из всей цепочки сущностей — шаблонов, узлов сети, элементов данных/веб-сценариев, триггеров. Идентичные комбинации `tag:value` (после раскрытия макросов) объединяются в одну, что позволяет избежать дублирования.

События восстановления, созданные при ручном закрытии, также включают раскрытые теги событий, 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 быть сгруппированы на основе тегов событий.
- теги проблем могут использоваться для сопоставления проблем с сервисами.

Сущности могут быть помечены одинаковым именем тега, но разными значениями тега (например, `component:memory` и `component:storage`). Аналогично, сущность может иметь тег без значения и тот же тег со значением (например, `database` и `database:postgresql`). Такие теги не считаются дубликатами.

Варианты использования

Некоторые распространенные варианты использования тегов:

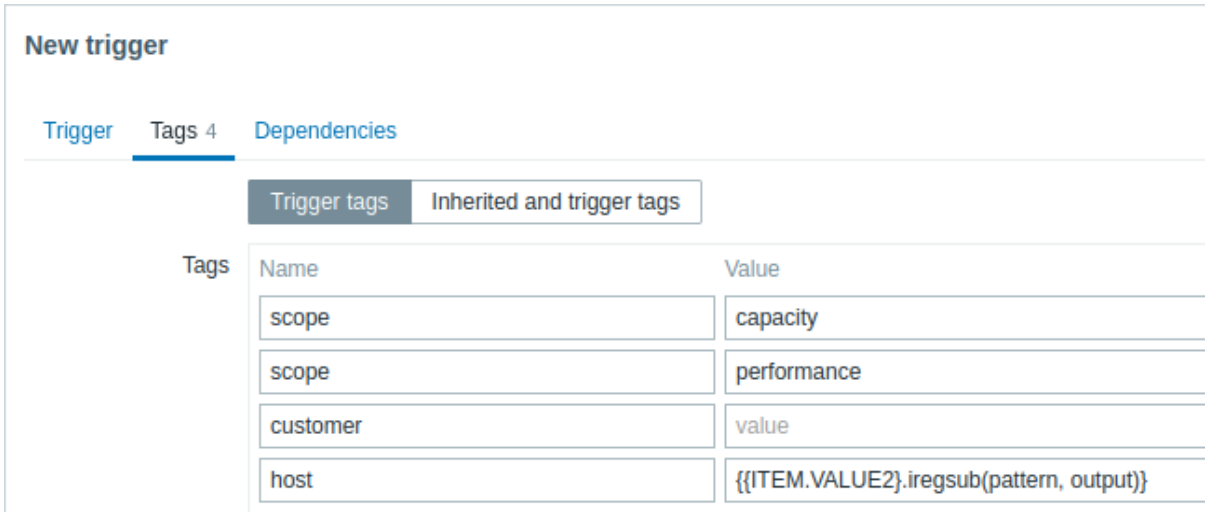
1. Пометка событий триггеров:
   - Определите тег триггера (например, `scope:performance`).
   - Проблемы, созданные этим триггером, будут иметь тег триггера.
2. Пометка проблем, унаследованных от шаблона:
   - Определите тег шаблона (например, `target:mysql`).
   - Проблемы, созданные триггерами из 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* значение тега у проблем будет преобразовано в данные, извлеченные из значения элемента данных.
6. Выявление проблем в файле журнала и их раздельное 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 настройке 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 триггеров (например, `scope:security` для trigger1 и `scope:availability` для trigger2).
   - Используйте фильтрацию по тегам в условиях действия, чтобы получать уведомления только о событиях, соответствующих данным тегов.
8. Идентификация проблем в уведомлениях:
   - Определите теги триггеров.
   - Используйте макрос {EVENT.TAGS} в уведомлении о проблеме.
   - Уведомление о проблеме будет содержать теги триггера, что упростит определение того, к какому приложению/сервису 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 low-level discovery (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.
    - Дополнительно используйте теги сервиса, чтобы связать сервис с SLA для расчета SLA.
12. Связывание сервисов с проблемами с помощью тегов проблем сервиса:
    - Определите тег проблемы в настройке сервиса (например, `target:mysql`).
    - Проблемы с совпадающим тегом будут автоматически коррелироваться с сервисом, а статус сервиса будет изменяться в соответствии с настроенными правилами расчета статуса сервиса.
13. Подавление проблем, когда узел сети находится в режиме обслуживания:
    - Определите теги в настройке периода обслуживания.
    - Проблемы с заданными тегами будут подавляться.
14. Предоставление доступа группам пользователей:
    - Определите теги в настройке группы пользователей.
    - Пользователи в группе пользователей смогут просматривать только проблемы с заданными тегами.

Конфигурация

Теги можно задать на отдельной вкладке, например, в настройках триггера:



Поддержка макросов

Встроенные и пользовательские макросы в тегах разрешаются в момент события. До наступления события эти макросы будут отображаться в веб-интерфейсе Zabbix неразрешёнными.

Макросы низкоуровневого обнаружения разрешаются в процессе обнаружения.

Следующие макросы могут использоваться в именах и значениях тегов триггера:

- встроенные макросы {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.*} (для ссылки на значения инвентарных данных узла сети из 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:

- встроенные макросы {HOST.HOST}, {HOST.NAME}, {HOST.CONN}, {HOST.DNS}, {HOST.IP}, {HOST.PORT} и {HOST.ID}
- встроенные макросы {INVENTORY.*}
- пользовательские макросы
- макросы низкоуровневого обнаружения (только в тегах прототипа узла сети и прототипа элемента данных)

Следующие макросы могут использоваться в уведомлениях на основе триггеров:

- встроенные макросы {EVENT.TAGS} и {EVENT.RECOVERY.TAGS} (эти макросы будут разрешены в список тегов события или тегов события восстановления, разделённых запятыми)
- встроенные макросы {EVENT.TAGSJSON} и {EVENT.RECOVERY.TAGSJSON} (эти макросы будут разрешены в JSON-массив, содержащий объекты тегов события или объекты тегов события восстановления)

Извлечение подстроки в тегах триггера

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 его значение после раскрытия макроса превышает 255 символов, оно будет усечено до 255 символов.

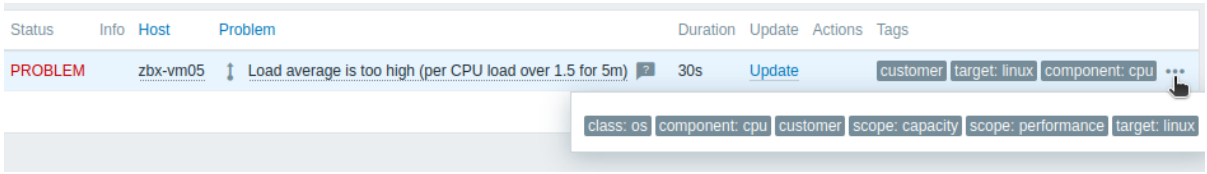
См. также: Использование функций макросов в макросах низкоуровневого обнаружения для тегирования событий.

Просмотр тегов событий

Теги, если они определены, можно просматривать у новых событий в следующих разделах:

- *Мониторинг* > *Проблемы*
- *Мониторинг* > *Проблемы* > *Сведения о событии*
- *Панели* > *Виджет проблем*

Порядок и количество отображаемых тегов определяются параметрами фильтрации *Приоритет отображения тегов* и *Показывать теги* в разделе *Мониторинг* > *Проблемы* или в виджете панели *Проблемы*. Обратите внимание, что может отображаться не более трёх тегов; если тегов больше, при наведении курсора на троеточие все теги будут показаны во всплывающем окне.



7 Визуализация

Пожалуйста, используйте боковую панель для доступа к содержимому раздела «Визуализация».

1 Графики

Обзор

Когда в Zabbix поступает большое количество данных, пользователям становится намного проще, если они могут видеть наглядное представление происходящего, а не только числа.

Именно здесь помогают графики. Графики позволяют с первого взгляда понять поток данных, сопоставить проблемы, определить, когда что-то началось, или наглядно показать, когда что-то может превратиться в проблему.

Zabbix предоставляет пользователям:

- встроенные простые графики для данных одного элемента данных
- возможность создавать более сложные настраиваемые графики
- быстрый доступ к сравнению нескольких элементов данных в ad-hoc графиках
- современные настраиваемые векторные графики и круговые диаграммы

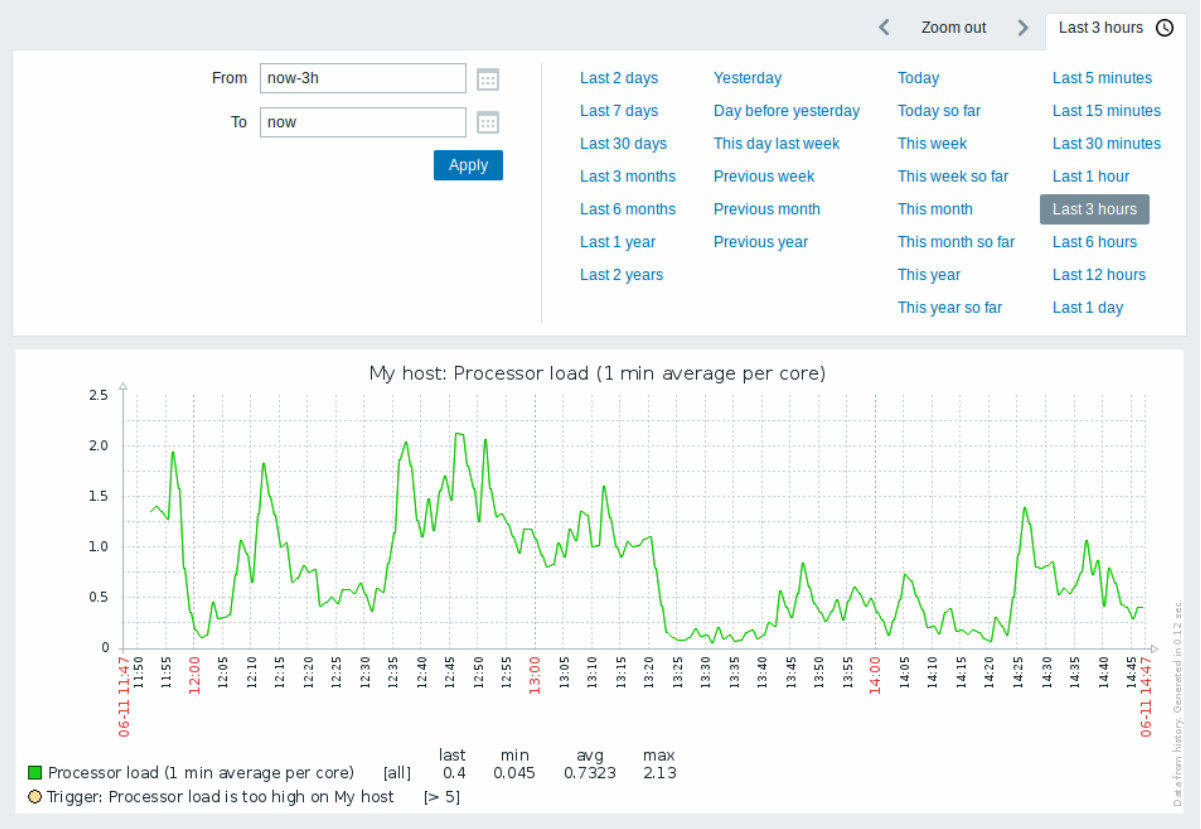
1 Простые графики

Обзор

Простые графики предназначены для визуализации данных, собранных элементами данных.

Для просмотра простых графиков от пользователя не требуется никакой настройки. Они автоматически доступны в Zabbix.

Просто перейдите в *Мониторинг* > *Последние данные* и нажмите на ссылку Graph для соответствующего элемента данных — отобразится график.



Note:
Простые графики доступны для всех числовых элементов данных. Для текстовых элементов данных в *Мониторинг* > *Последние данные* доступна ссылка на History.

Селектор периода времени

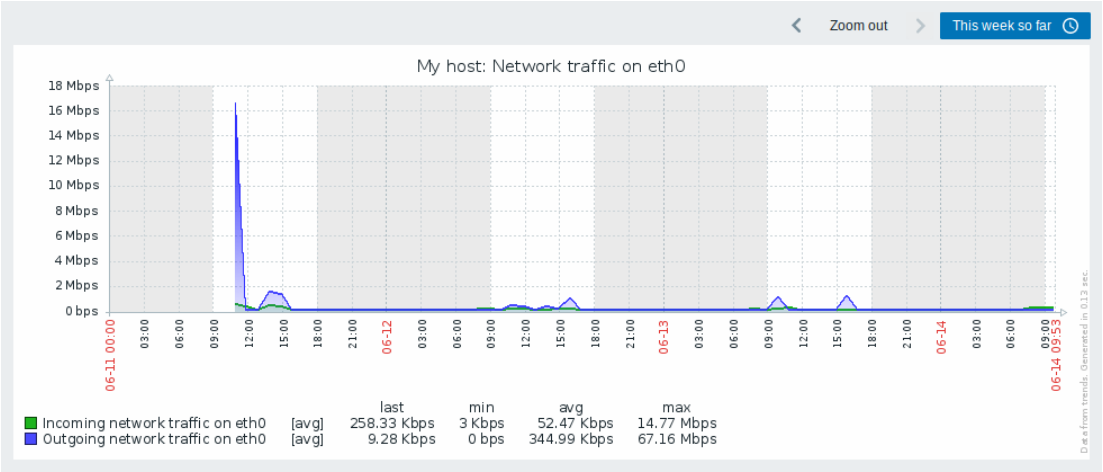
Селектор периода времени над графиком позволяет выбрать часто используемые периоды одним щелчком мыши. Для получения дополнительной информации см. Селектор периода времени и узла сети.

Последние данные и более длительные периоды

Для самых последних данных рисуется **одна** линия, соединяющая каждое полученное значение. Одна линия рисуется до тех пор, пока для одного значения доступен хотя бы один горизонтальный пиксель.

Для данных, отображающих более длительный период, рисуются **три линии** — тёмно-зелёная показывает среднее значение, а светло-розовая и светло-зелёная линии показывают максимальные и минимальные значения в данный момент времени. Пространство между верхними и нижними значениями заполняется жёлтым фоном.

Рабочее время (рабочие дни) отображается на графиках белым фоном, а нерабочее время — серым (в стандартной теме веб-интерфейса *Original blue*).

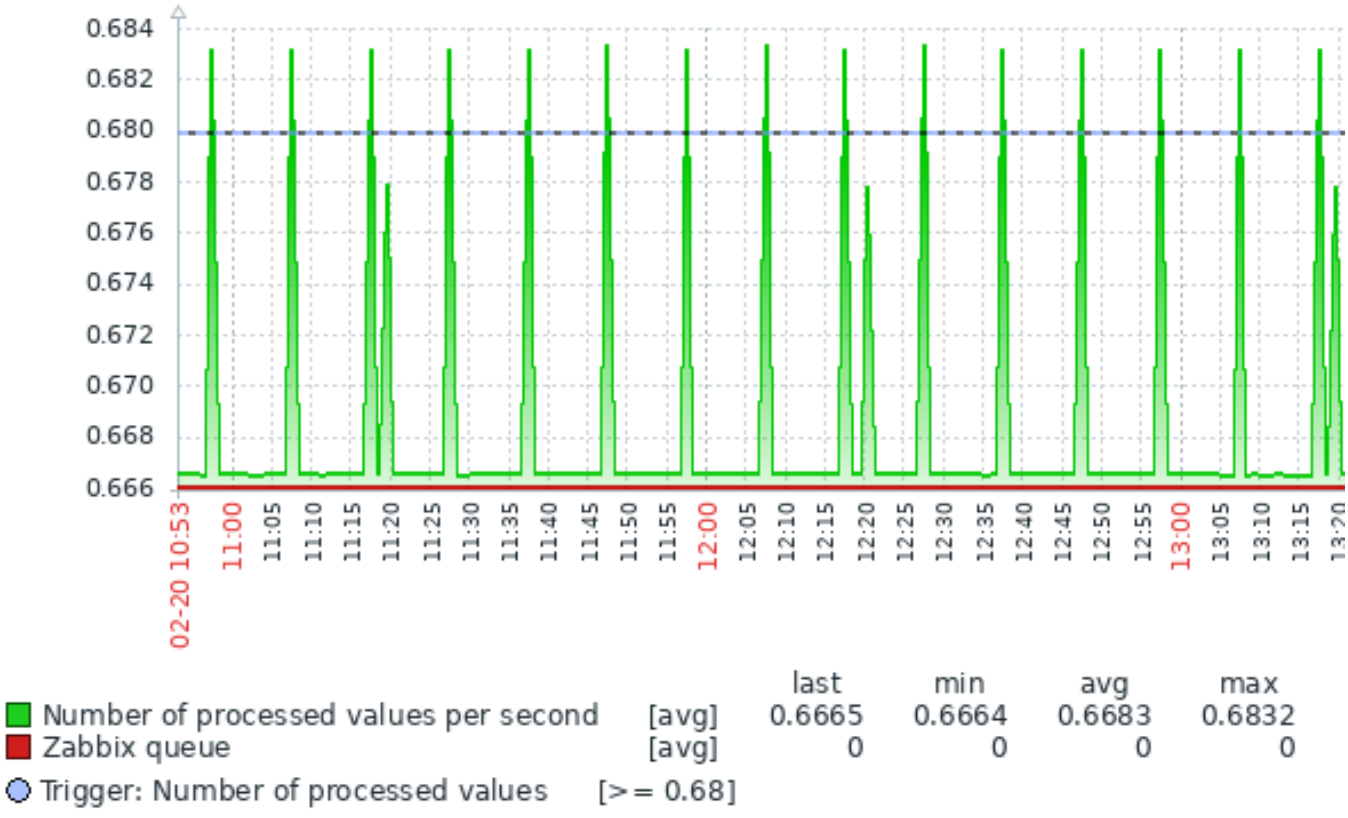


Рабочее время всегда отображается на простых графиках, тогда как его отображение в пользовательских графиках зависит от предпочтений пользователя.

Рабочее время не отображается, если график показывает период более 3 месяцев.

Линии триггеров

Простые триггеры отображаются в виде линий с черными штрихами поверх цвета важности триггера — обратите внимание на синюю линию на графике и информацию о триггере, отображаемую в легенде. На графике может отображаться до 3 линий триггеров; если триггеров больше, то приоритет отдается триггерам с более низкой важностью. Триггеры всегда отображаются на простых графиках, тогда как их отображение на пользовательских графиках зависит от предпочтений пользователя.



Создание 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, откуда поступают данные.

На то, будет ли использоваться история или тренды, влияют несколько факторов:

- срок хранения истории элемента данных. Например, история элемента данных может храниться 14 дней. В таком случае любые данные старше четырнадцати дней будут браться 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 всё ещё доступна).
- если тренды отключены, для построения графика используется история элемента данных — если она доступна за этот период.

Отсутствие данных

Для элементов данных с регулярным интервалом обновления на графике ничего не отображается, если данные элемента данных не собираются.

Однако для элементов данных типа trapper и элементов данных с интервалом обновления по расписанию (и регулярным интервалом обновления, установленным в 0) рисуется прямая линия до первого собранного значения и от последнего собранного значения до конца графика; линия находится соответственно на уровне первого/последнего значения.

Переключение на необработанные значения

Выпадающий список в правом верхнем углу позволяет переключиться с простого графика на списки *Значения/500 последних значений*. Это может быть полезно для просмотра числовых значений, из которых состоит график.

Представленные здесь значения являются необработанными, то есть единицы измерения или постобработка значений не используются. Однако преобразование значений применяется.

Известные проблемы

Смотрите известные проблемы по графикам.

2 Пользовательские графики

Обзор

Пользовательские графики, как следует из названия, предоставляют возможности настройки.

Хотя простые графики хорошо подходят для просмотра данных одного элемента данных, они не предоставляют возможностей конфигурирования.

Таким образом, если вы хотите изменить стиль графика или способ отображения линий, либо сравнить несколько элементов данных, например входящий и исходящий трафик на одном графике, вам нужен пользовательский график.

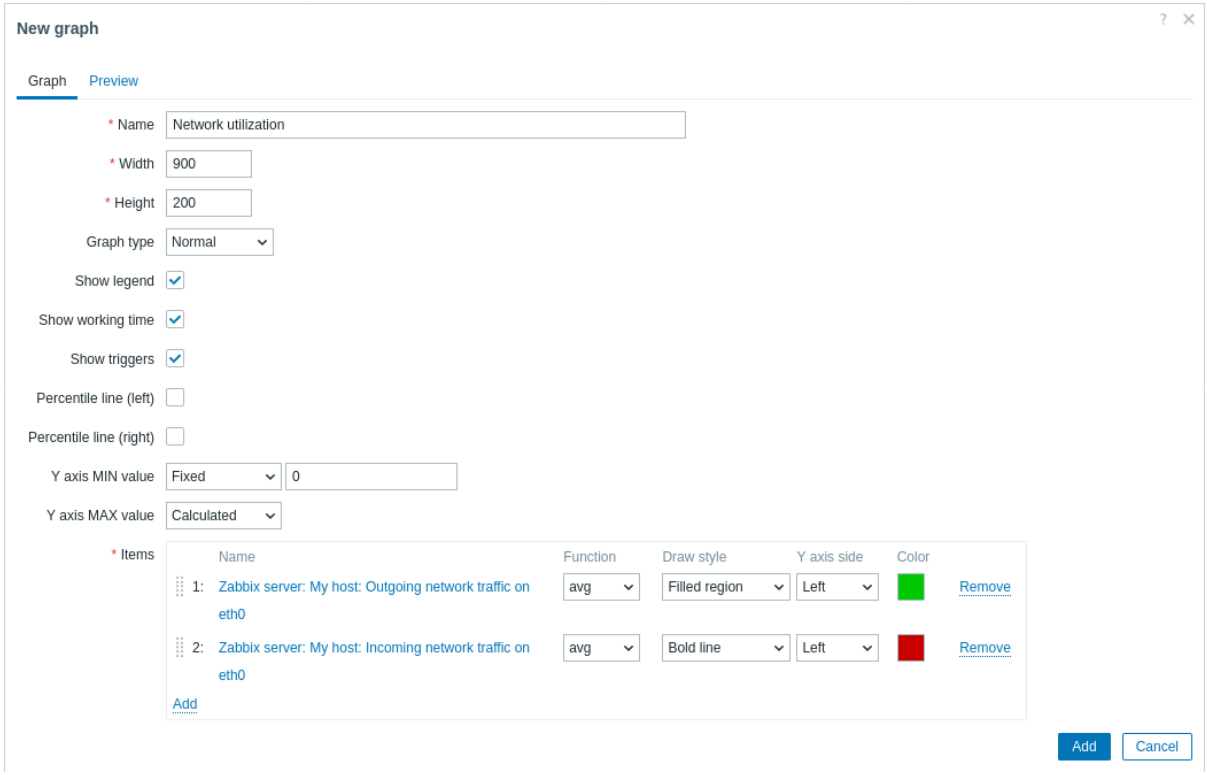
Пользовательские графики настраиваются вручную.

Их можно создавать для узла сети, нескольких узлов сети или одного шаблона.

Настройка пользовательских графиков

Чтобы создать пользовательский график, выполните следующие действия:

- Перейдите в *Сбор данных > Узлы сети (или Шаблоны)*
- Нажмите *Графики* в строке рядом с нужным узлом сети или шаблоном
- На экране графиков нажмите *Создать график*
- Измените атрибуты графика



Все обязательные поля ввода отмечены красной звездочкой.

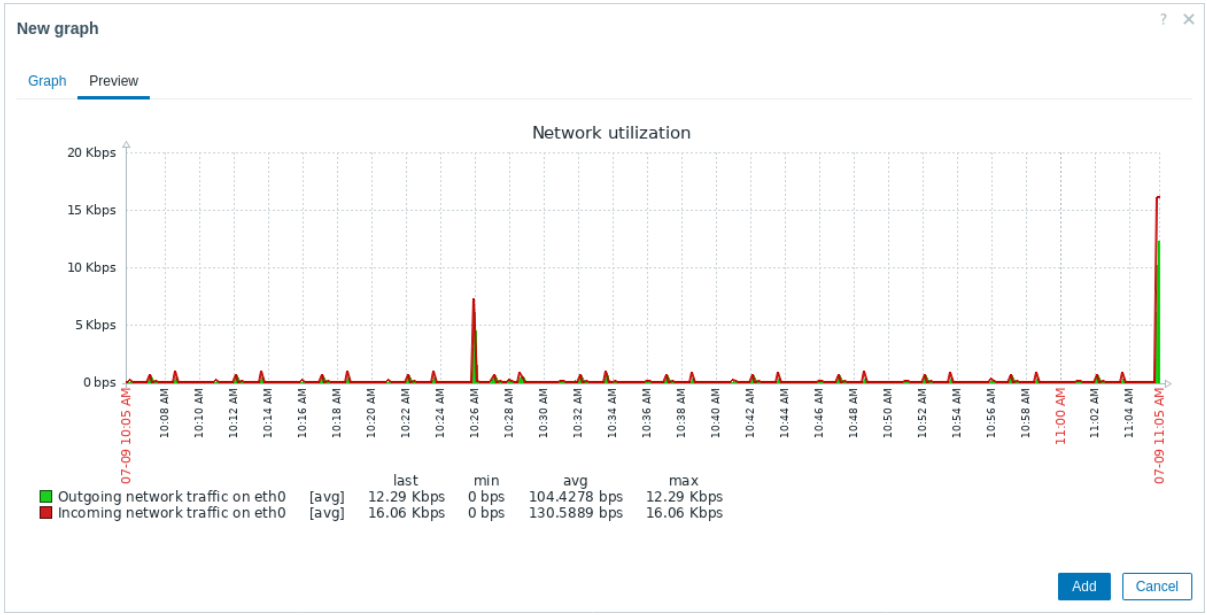
Атрибуты графика:

Параметр	Описание
Имя	Уникальное имя графика. В этом поле поддерживаются макросы выражений, но только с функциями `avg`, `last`, `min` и `max`, где время указывается как параметр (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 ссылаться неявно, например `{?avg(//key,1h)}`.
Ширина	Ширина графика в пикселях (только для предварительного просмотра и круговых/разделённых круговых графиков).
Высота	Высота графика в пикселях.
Тип графика	Тип графика: **Обычный** - обычный график, значения отображаются линиями **Накопительный** - накопительный график, отображаются заполненные области **Круговой** - круговой график **Разделённый** - "разделённый" круговой график, сектора отображаются как "вынутые" из круга
Показывать легенду	Если установить этот флажок, будет отображаться легенда графика.
Показывать рабочее время	Если выбрано, нерабочие часы будут показаны серым фоном. Этот параметр недоступен для круговых и разделённых круговых графиков.
Показывать триггеры	Если выбрано, простые триггеры будут отображаться в виде линий с чёрным пунктиром поверх цвета важности триггера. Этот параметр недоступен для круговых и разделённых круговых графиков.
Линия процентиля (слева)	Отображать процентиль для левой оси Y. Если, например, установлен 95-й процентиль, то линия процентиля будет находиться на уровне, ниже которого лежат 95 процентов значений. Отображается ярко-зелёной линией. Доступно только для обычных графиков.
Линия процентиля (справа)	Отображать процентиль для правой оси Y. Если, например, установлен 95-й процентиль, то линия процентиля будет находиться на уровне, ниже которого лежат 95 процентов значений. Отображается ярко-красной линией. Доступно только для обычных графиков.

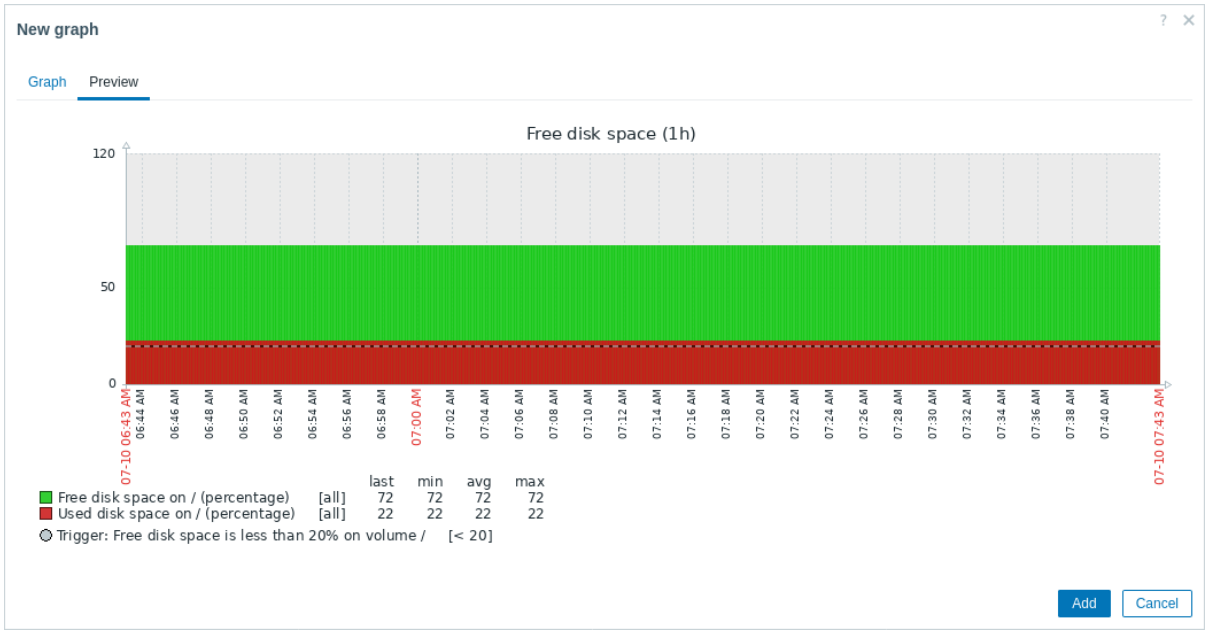
Параметр	Описание
Минимальное значение оси Y	Минимальное значение оси Y: **Вычисляемое** - минимальное значение оси Y будет вычислено автоматически. **Фиксированное** - фиксированное минимальное значение оси Y. **Элемент данных** - последнее значение выбранного элемента данных будет минимальным значением. Этот параметр недоступен для круговых и разделённых круговых графиков.
Максимальное значение оси Y	Максимальное значение оси Y: **Вычисляемое** - максимальное значение оси Y будет вычислено автоматически. **Фиксированное** - фиксированное максимальное значение оси Y. **Элемент данных** - последнее значение выбранного элемента данных будет максимальным значением Этот параметр недоступен для круговых и разделённых круговых графиков.
3D-вид	Включить стиль 3D. Только для круговых и разделённых круговых графиков.
Элементы данных	Элементы данных, данные которых должны отображаться на этом графике. Нажмите *Добавить*, чтобы выбрать элементы данных. Вы также можете выбрать различные параметры отображения (функция, стиль рисования, отображение по левой/правой оси, цвет).
Порядок сортировки (0→100)	Порядок отрисовки. 0 будет обработан первым. Можно использовать для отрисовки линий или областей позади (или перед) другими. Вы можете перетаскивать элементы данных с помощью значка в начале строки, чтобы задать порядок сортировки или определить, какой элемент данных будет отображаться перед другим.
Имя	Имя выбранного элемента данных отображается как ссылка. Нажатие на ссылку открывает список других доступных элементов данных.
Тип	Тип (доступно только для круговых и разделённых круговых графиков): **Простой** - значение элемента данных пропорционально представлено на круговой диаграмме **Сумма графика** - значение элемента данных представляет всю круговую диаграмму Обратите внимание, что цвет элемента данных "сумма графика" будет виден только в той части, которая не занята "пропорциональными" элементами данных.
Функция	Выберите, какие значения будут отображаться, если для элемента данных существует более одного значения на один вертикальный пиксель графика: **all** - отображать все возможные значения (минимум, максимум, среднее) на графике. Обратите внимание, что для более коротких периодов эта настройка не влияет на отображение; только для более длинных периодов, когда увеличивается плотность данных в одном вертикальном пикселе графика, `all` начинает отображать минимальные, максимальные и средние значения. Эта функция доступна только для типа графика *Обычный*. См. также: Создание графиков из history/trends. **avg** - отображать средние значения **last** - отображать последние значения. Эта функция доступна только если в качестве типа графика выбран *Круговой/Разделённый круговой*. **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), который будет применён к данным элемента данных - *Линия*, *Жирная линия*, *Заполненная область*, *Точка*, *Пунктирная линия*, *Градиентная линия*.
Сторона оси Y	Выберите сторону оси Y для отображения данных элемента данных - *Слева*, *Справа*.
Цвет	Выберите цвет, который будет применён к данным элемента данных.

Предпросмотр графика

На вкладке *Предпросмотр* отображается предварительный просмотр графика, чтобы вы могли сразу увидеть, что создаёте.



Обратите внимание, что в предварительном просмотре не будут показаны какие-либо данные для элементов данных шаблона.



В этом примере обратите внимание на жирную пунктирную линию, отображающую уровень триггера, а также на информацию о триггере, отображаемую в легенде.

Note:
Может быть отображено не более 3 линий триггеров. Если триггеров больше, то для отображения в первую очередь выбираются триггеры с более низкой важностью.

 Если высота графика задана менее 120 пикселей, в легенде не будет отображаться ни один триггер.

3 Ситуационные графики

Обзор

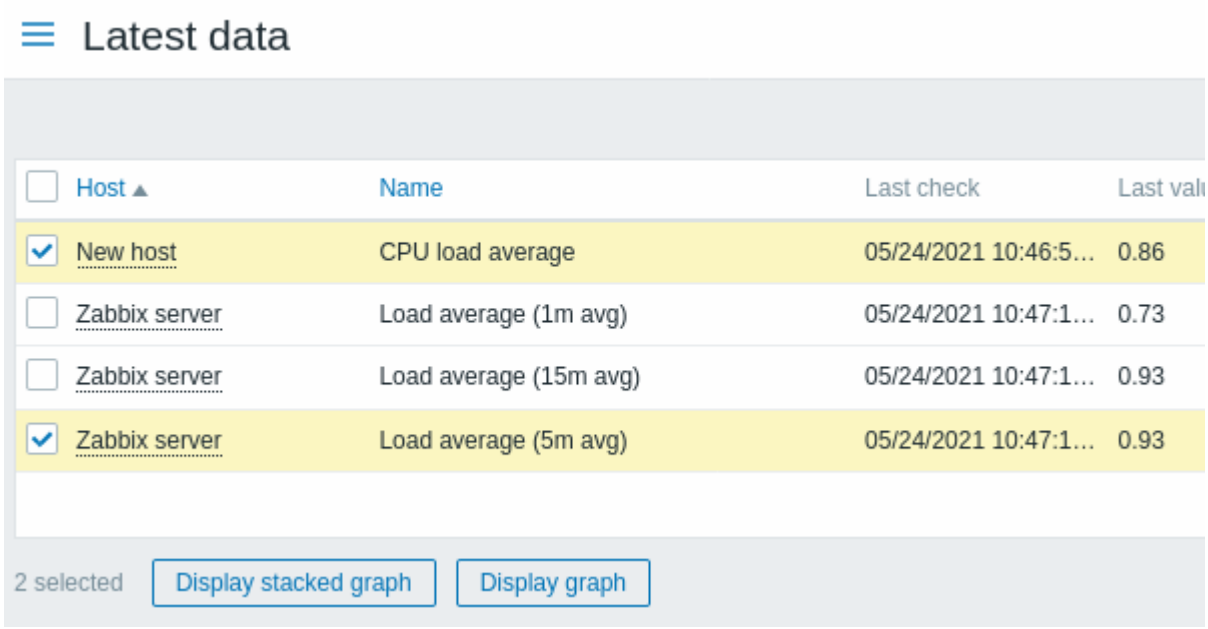
Хотя простой график отлично подходит для доступа к данным одного элемента данных, а пользовательские графики предлагают возможности настройки, ни один из этих двух вариантов не позволяет быстро создать график сравнения для нескольких элементов данных с минимальными усилиями и без сопровождения.

Чтобы решить эту проблему, можно очень быстро создавать специальные графики для нескольких элементов данных.

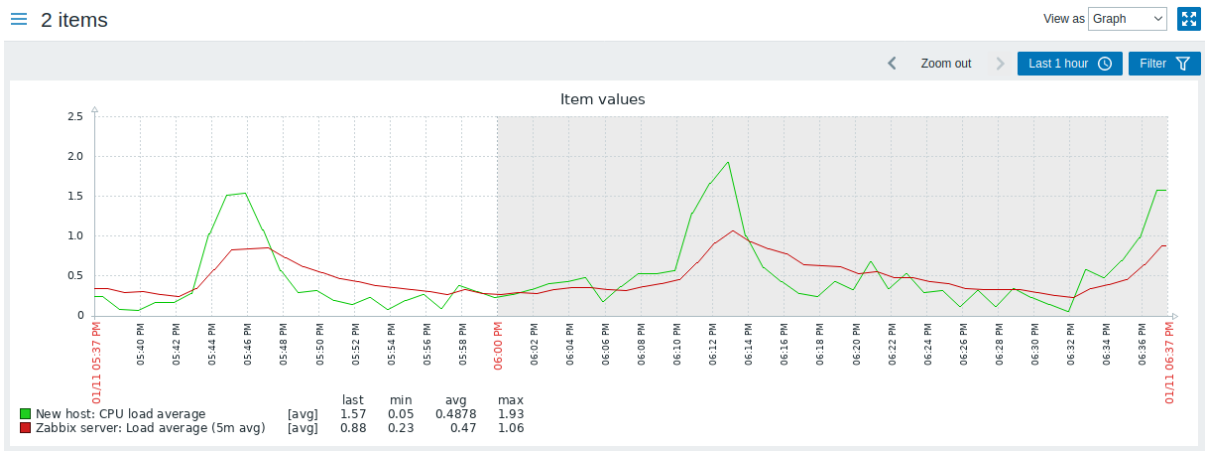
Настройка

Чтобы создать ad-hoc график, выполните следующие действия:

- Перейдите в *Мониторинг* → *Последние данные*
- Используйте фильтр, чтобы отобразить нужные элементы данных
- Отметьте флажками элементы данных, которые хотите вывести на график
- Нажмите кнопку *Display stacked graph* или *Display graph*

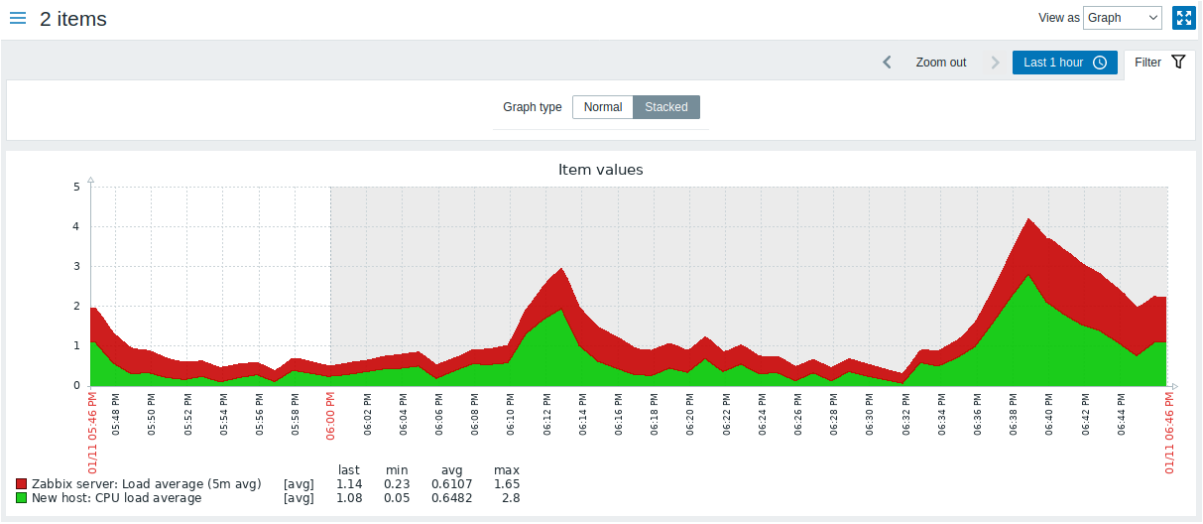


Ваш график будет создан мгновенно:



Обратите внимание, что во избежание отображения слишком большого количества линий на графике для каждого элемента данных отображается только среднее значение (линии минимального/максимального значений не отображаются). Триггеры и информация о триггерах на графике не отображаются.

В окне созданного графика вам доступен селектор периода времени, а также возможность переключаться с «обычного» линейного графика на составной и обратно.



#####4 Агрегирование на графиках {#manual-config-visualization-graphs-aggregate}

Обзор

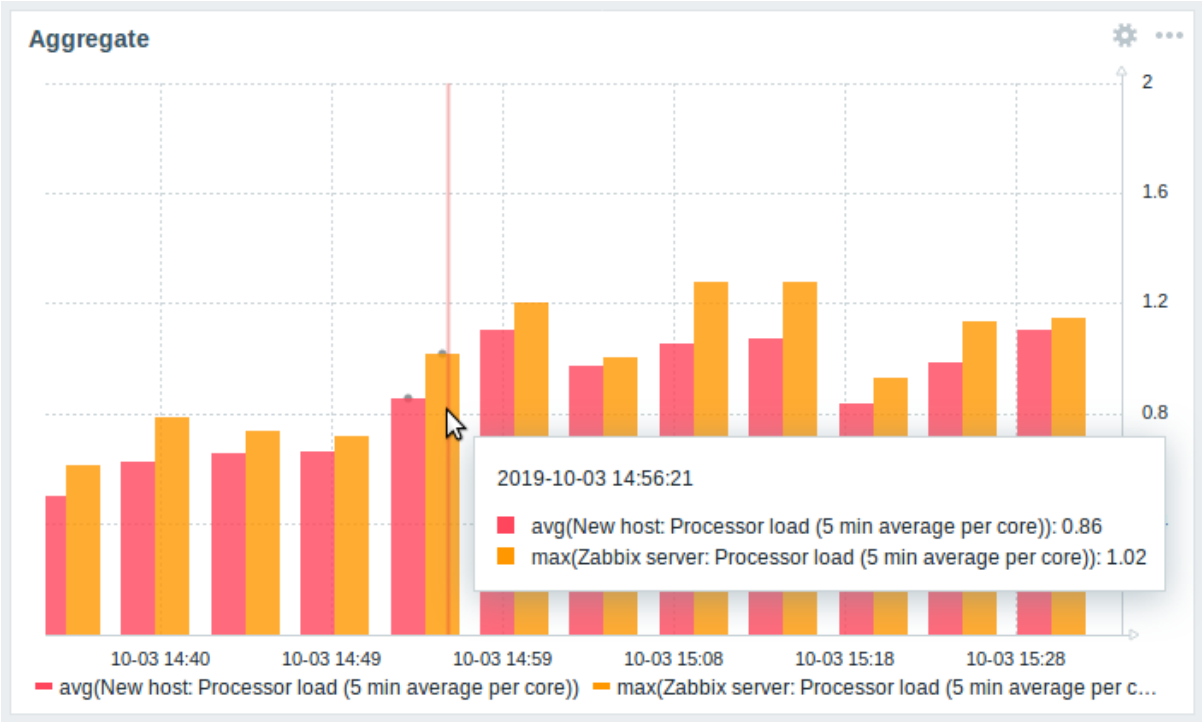
Функции агрегации, доступные в виджетах графика и круговой диаграммы на панели мониторинга, позволяют отображать агрегированное значение за выбранный интервал (5 минут, час, день) вместо всех значений.

В этом разделе подробнее описаны параметры агрегации в виджете графика.

Доступны следующие параметры агрегации:

- min
- max
- avg
- count
- sum
- first (первое отображаемое значение)
- last (последнее отображаемое значение)

Наиболее интересный вариант использования агрегации данных — возможность создавать наглядные сравнения данных за некоторый период рядом друг с другом:

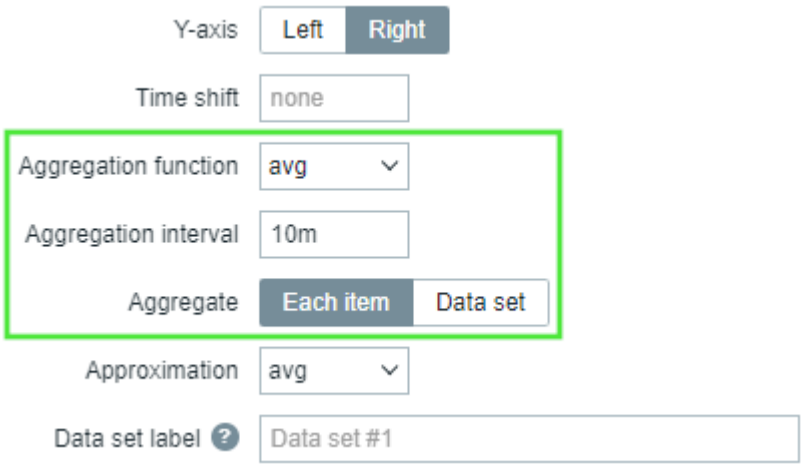


При наведении курсора на точку времени на графике, помимо элементов данных и их агрегированных значений, отображаются дата и время. Элементы данных отображаются в скобках с префиксом в виде используемой функции агрегации. Если в виджете графика настроена *Метка набора данных*, метка отображается в скобках с префиксом в

виде используемой функции агрегации. Обратите внимание, что это дата и время точки на графике, а не фактических значений.

Настройка

Параметры агрегации доступны в настройках набора данных при настройке виджета графика.



Вы можете выбрать функцию агрегации и временной интервал. Поскольку набор данных может включать несколько элементов данных, также доступен дополнительный параметр, позволяющий отображать агрегированные данные для каждого элемента данных отдельно или для всех элементов данных набора как одно агрегированное значение.

Варианты использования

Среднее количество запросов к Nginx серверу

Просмотр среднего количества запросов в секунду за день к Nginx серверу:

- добавьте элемент данных количества запросов в секунду к набору данных
- выберите функцию агрегации и укажите период 1d
- отобразится столбчатый график, где каждый столбец будет представлять среднее количество запросов в секунду за день

Минимальное количество свободного места за неделю среди кластеров

Просмотр минимального количества свободного места среди кластеров за неделю.

- добавьте набор данных: узлы сети `cluster*`, ключ "Free disk space on /data"
- выберите функцию агрегации и укажите период `1w`
- отобразится столбчатый график, где каждый столбец будет представлять минимальное количество места на диске за неделю по каждому /data тому кластера

2 Карты сети

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).

Перейдем к настройке карты сети.

1 Настройка карты сети

Обзор

Настройка карты в Zabbix требует, чтобы вы сначала создали карту, определив ее общие параметры, а затем начали заполнять саму карту элементами и их связями.

Вы можете заполнять карту элементами, которыми могут быть 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, а также настроить специальное отображение недавних проблем. Вы можете связывать значки и определять информацию, которая будет отображаться на связях.

Вы можете добавлять пользовательские URL-адреса, доступные по щелчку на значках. 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.

Управление картами выполняется в *Monitoring* > *Maps*, где их можно настраивать, администрировать и просматривать. В представлении мониторинга вы можете щелкать по значкам и использовать ссылки на некоторые скрипты и URL-адреса.

Карты сети основаны на векторной графике (SVG).

Общедоступные и частные карты

Все пользователи в Zabbix (включая пользователей без прав администратора) могут создавать карты сети. У карт есть владелец — пользователь, который их создал. Карты могут быть общедоступными или частными.

- *Общедоступные* карты видны всем пользователям, однако для их просмотра пользователь должен иметь доступ на чтение как минимум к одному элементу карты. Общедоступные карты можно редактировать, если пользователь/группа пользователей имеет права чтения-записи для этой карты и как минимум права на чтение ко всем элементам соответствующей карты, включая триггеры в связях.
- *Частные* карты видны только их владельцу и пользователям/группам пользователей, которым карта предоставлена в общий доступ владельцем. Обычные пользователи (не Super admin) могут предоставлять общий доступ только группам, в которые они входят, и пользователям, входящим в эти группы. Пользователи уровня Admin могут видеть частные карты независимо от того, являются ли они владельцами или входят ли в список пользователей, которым предоставлен доступ. Частные карты могут редактироваться владельцем карты, а также в случае, если пользователь/группа пользователей имеет права чтения-записи для этой карты и как минимум права на чтение ко всем элементам соответствующей карты, включая триггеры в связях.

Элементы карты, на чтение которых у пользователя нет прав, отображаются значком серого цвета, а вся текстовая информация об элементе скрыта. Однако метка триггера видна, даже если у пользователя нет прав на этот триггер.

Чтобы добавить элемент на карту, пользователь также должен иметь как минимум права на чтение этого элемента.

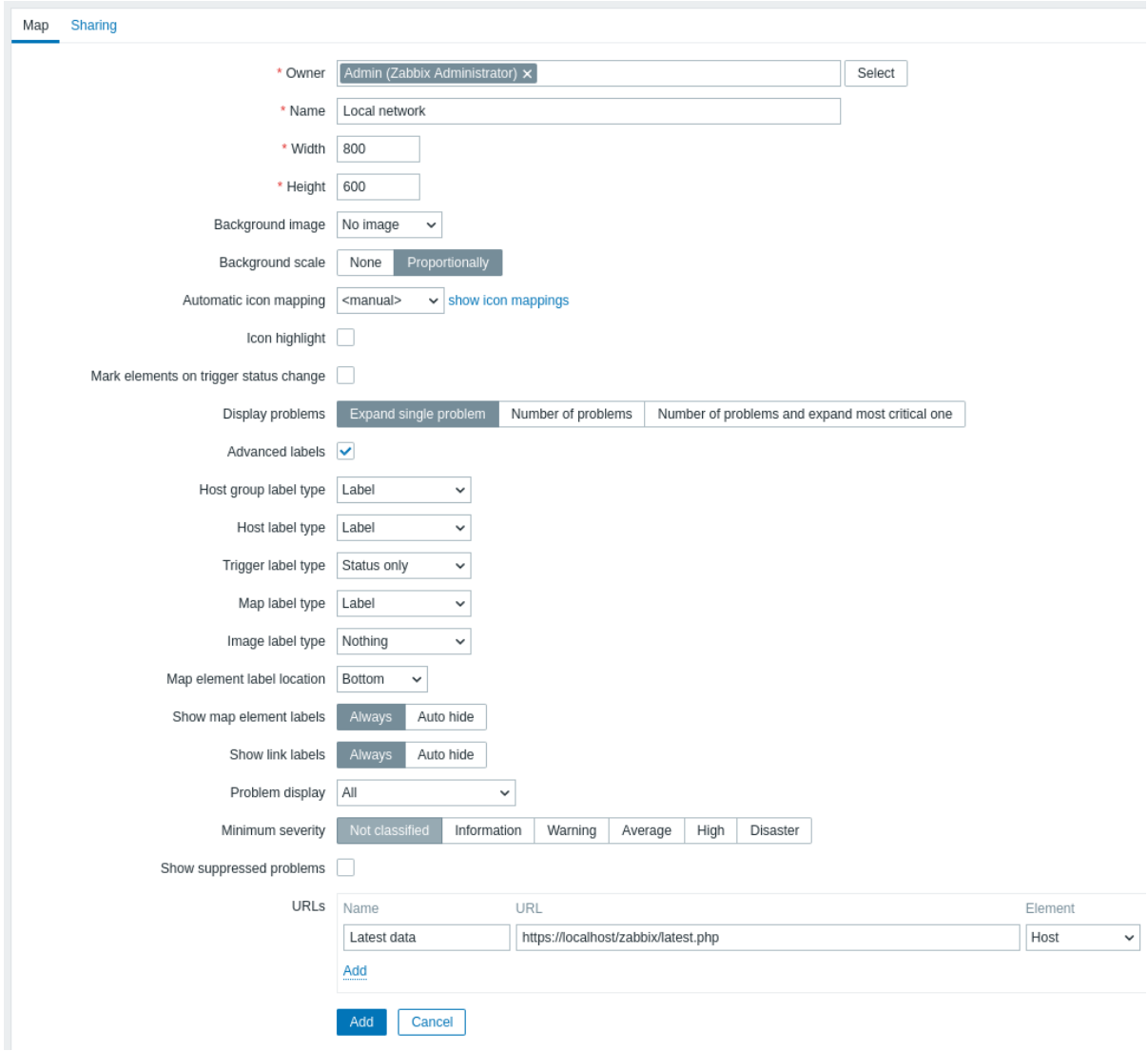
Создание карты

Чтобы создать карту, выполните следующие действия:

1. Перейдите в *Monitoring* > *Maps*.
2. Перейдите к представлению со всеми картами.
3. Нажмите *Create map*.

Вы также можете использовать кнопку *Clone* 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 **Map** содержит общие атрибуты карты:



Все обязательные поля ввода отмечены красной звездочкой.

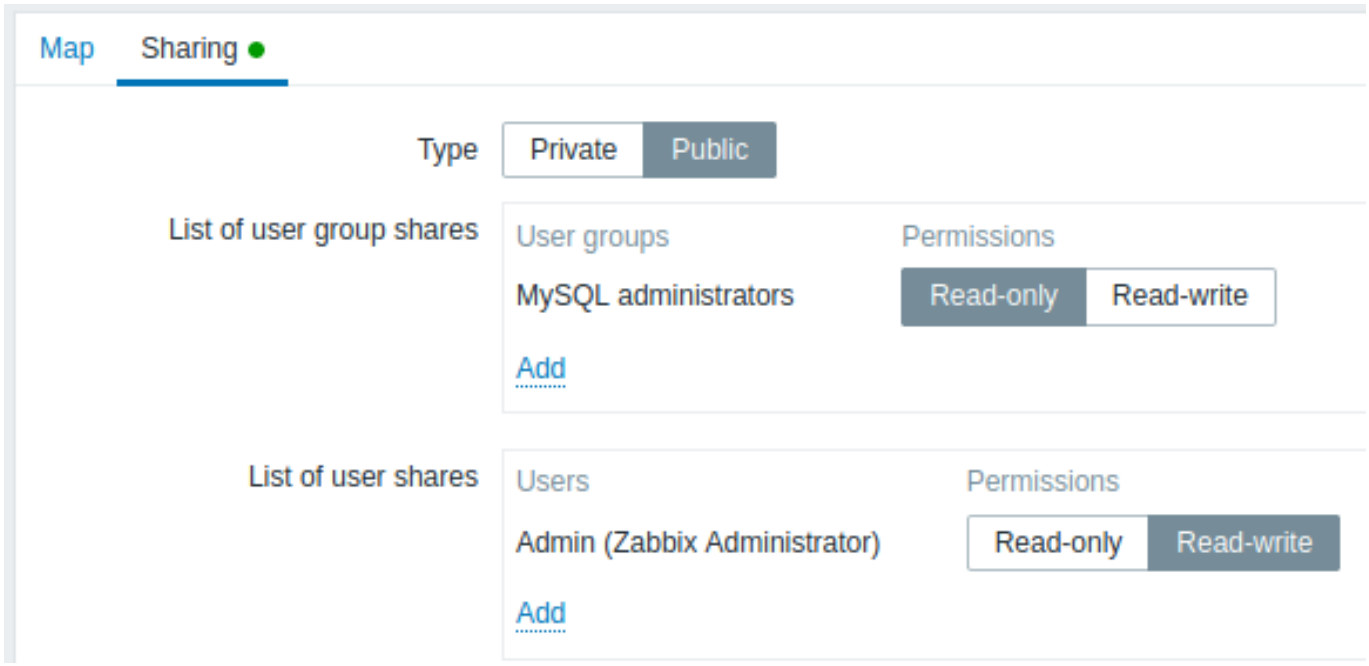
Общие атрибуты карты:

Параметр	Описание
Owner	Имя владельца карты.
Name	Уникальное имя карты.
Width	Ширина карты в пикселях.
Height	Высота карты в пикселях.
Background image	Использование фонового изображения: **No image** — без фонового изображения (белый фон) **Image** — выбранное изображение будет использоваться в качестве фонового. Вы можете использовать географическую карту или любое другое изображение, чтобы улучшить свою карту.
Background scale	Использование масштабирования фона: **None** — без масштабирования; **Proportionally** — заполнить весь фон карты, сохраняя пропорции изображения (по умолчанию).
Automatic icon mapping	Можно настроить использование автоматического сопоставления значков, заданного в *Administration* > *General* > *Icon mapping*. Сопоставление значков позволяет связывать определенные значки с определенными полями инвентарных данных узла сети.
Icon highlight	Если установить этот флажок, элементы карты будут подсвечиваться. Элементы с активным триггером получат круглый фон того же цвета, что и триггер с наивысшей важностью. Кроме того, вокруг круга будет отображаться толстая зеленая линия, если все проблемы подтверждены. Элементы со статусом ”disabled” или ”in maintenance” получат квадратный фон — серый и оранжевый соответственно. См. также: Просмотр карт

Параметр	Описание
Mark elements on trigger status change	Недавнее изменение состояния триггера (недавняя проблема или восстановление) будет выделено маркерами (красными треугольниками, направленными внутрь) с трех сторон значка элемента, свободных от подписи. Маркеры отображаются в течение 30 минут.
Display problems	Выберите, как проблемы отображаются у элемента карты: **Expand single problem** — если проблема только одна, отображается ее имя. В противном случае отображается общее количество проблем. **Number of problems** — отображается общее количество проблем **Number of problems and expand most critical one** — отображаются имя наиболее критичной проблемы и общее количество проблем. «Наиболее критичная» определяется на основе важности проблемы и, при равенстве, ID события проблемы (сначала отображается более высокий ID или более поздняя проблема). Для *trigger map element* это определяется на основе важности проблемы и, при равенстве, позиции триггера в списке триггеров. При наличии нескольких проблем одного и того же триггера будет отображаться самая недавняя.
Advanced labels	Если установить этот флажок, вы сможете определить отдельные типы подписей для отдельных типов элементов.
Host group label type	Тип подписи, используемый для элемента карты: **Label** — подпись элемента карты **IP address** — IP-адрес **Element name** — имя элемента (например, имя узла сети) **Status only** — только статус (OK или PROBLEM) **Nothing** — подписи не отображаются
Host label type *Trigger label type* *Map label type* *Image label type*	
Map element label location	Расположение подписи относительно элемента карты: **Bottom** — под элементом карты **Left** — слева **Right** — справа **Top** — над элементом карты
Show map element labels	Выберите, как отображать подписи элементов карты: **Always** — отображать всегда (по умолчанию); **Auto hide** — скрывать подпись, если на нее не наведен указатель и она не выбрана.
Show link labels	Выберите, как отображать подписи ссылок: **Always** — отображать всегда (по умолчанию); **Auto hide** — скрывать подпись, если на нее не наведен указатель и она не выбрана.
Problem display	Отображать количество проблем как: **All** — будет отображаться полное количество проблем **Separated** — количество неподтвержденных проблем будет отображаться отдельно как часть общего количества проблем **Unacknowledged only** — будет отображаться только количество неподтвержденных проблем
Minimum severity	Проблемы ниже выбранного минимального уровня важности не будут отображаться на карте. Например, если выбрано *Warning*, изменения, вызванные триггерами уровня *Information* и *Not classified*, не будут отражаться на карте.
Show suppressed problems	Установите флажок, чтобы отображать проблемы, которые в противном случае были бы подавлены (не показаны) из-за обслуживания узла сети.
URLs	Здесь можно определить URL-адреса для каждого типа элемента, где: **Name** — метка для URL; **URL** — URL-адрес (до 2048 символов); **Element** — тип элемента (например, *Host*, *Host group* 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 и выполните поиск по строке ’map URL names and values’.

Общий доступ

Вкладка **Общий доступ** содержит тип карты, а также параметры общего доступа (группы пользователей, пользователи) для приватных карт:



Параметр	Описание
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 будет доступна карта. Можно разрешить доступ только на чтение или чтение-запись.
List of user shares	Выберите пользователей, которым будет доступна карта. Можно разрешить доступ только на чтение или чтение-запись.

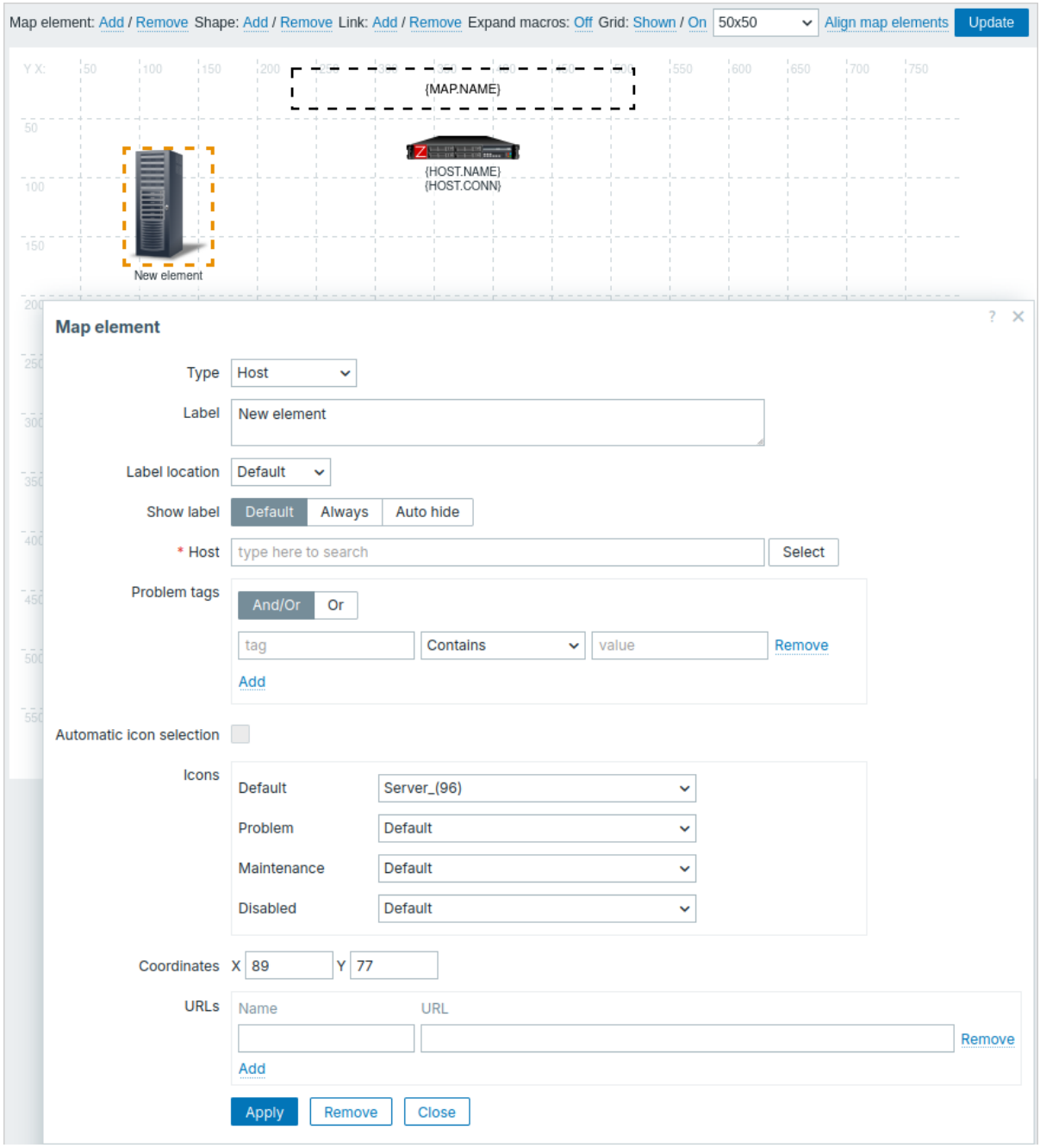
После нажатия *Add* для сохранения этой карты будет создана пустая карта с именем, размерами и определенными настройками. Теперь необходимо добавить некоторые элементы. Для этого нажмите *Edit* в списке карт, чтобы открыть область редактирования.

Добавление элементов

Чтобы добавить элемент, нажмите *Add* рядом с `Map element`. Новый элемент появится в левом верхнем углу карты. Перетащите его в любое нужное место.

Обратите внимание, что при включенной опции Grid (”On”) элементы всегда будут выравниваться по сетке (можно выбрать различные размеры сетки из выпадающего списка, а также скрыть/показать сетку). Если вы хотите размещать элементы где угодно без выравнивания, переключите эту опцию в ”Off”. (Позже можно выровнять произвольно размещенные элементы по сетке, нажав *Align map elements*.)

Теперь, когда на карте уже есть несколько элементов, вы можете начать различать их, задавая имена и т. д. При нажатии на элемент отображается форма, в которой можно задать тип элемента, указать имя, выбрать другую иконку и т. д.



Атрибуты элемента карты:

Parameter	Description
Type	Тип элемента: **Host** - иконка, представляющая состояние всех триггеров выбранного узла сети **Map** - иконка, представляющая состояние всех элементов карты **Trigger** - иконка, представляющая состояние одного или нескольких триггеров **Host group** - иконка, представляющая состояние всех триггеров всех узлов сети, принадлежащих выбранной группе **Image** - иконка, не связанная ни с каким ресурсом
Label	Метка иконки, любая строка. Можно использовать макросы и многострочные строки. В этом поле поддерживаются макросы выражений, но только с функциями `avg`, `last`, `min` и `max`, где в качестве параметра используется время (например, `{?avg(/host/key,1h)}`). Полный список поддерживаемых макросов см. в разделе supported macros, выполнив поиск по строке 'map element labels'.

Parameter	Description
Label location	Расположение метки относительно иконки: **Default** - расположение метки карты по умолчанию; **Bottom** - под иконкой; **Left** - слева; **Right** - справа; **Top** - над иконкой.
Show label	Выберите, как отображать метку элемента: **Default** - использовать настройки *Show map element labels* из конфигурации карты (по умолчанию); **Always** - отображать всегда (по умолчанию); **Auto hide** - скрывать метку, если на нее не наведен указатель и она не выбрана.
Host	Выберите узел сети; либо введите имя узла сети. Это поле поддерживает автодополнение, поэтому при вводе имени узла сети будет отображаться выпадающий список подходящих узлов сети. Прокрутите вниз, чтобы выбрать нужный. Нажмите значок удаления (x), чтобы убрать 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 отображаться выпадающий список подходящих карт. Прокрутите вниз, чтобы выбрать нужную. Нажмите значок удаления (x), чтобы убрать 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 его из выпадающего списка подходящих триггеров. Нажмите значок удаления (x), чтобы убрать выбор. Это поле доступно только для элементов типа *Trigger*. Порядок выбранных триггеров можно изменять, но только в пределах одного и того же уровня важности 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 важности 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 важности) или от 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 важности триггера.
Host group	Выберите группу узлов сети; либо введите группу узлов сети. Это поле поддерживает автодополнение, поэтому при вводе имени группы узлов сети будет отображаться выпадающий список подходящих групп узлов сети. Прокрутите вниз, чтобы выбрать нужную. Нажмите значок удаления (x), чтобы убрать выбор. Это поле доступно только для элементов типа *Host group*.
Problem tags	Укажите теги, чтобы ограничить количество проблем, отображаемых 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 доступно 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 выполняться 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 этот флажок установлен, для определения отображаемой иконки будет использоваться сопоставление иконок.

Parameter	Description
Icons	Можно выбрать отображение разных иконок для элемента в следующих случаях: *Default*, *Problem*, *Maintenance* и *Disabled*. Это поле доступно только для элементов типа *Host*.
Coordinates	Координаты **X** и **Y** элемента карты.
URLs	Здесь для элемента можно задать специфичные URL, где: **Name** - метка 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 и том же меню. В именах и значениях элементов карты можно использовать макросы. Полный список см. в разделе supported macros, выполнив поиск по строке 'map URL names and values'.

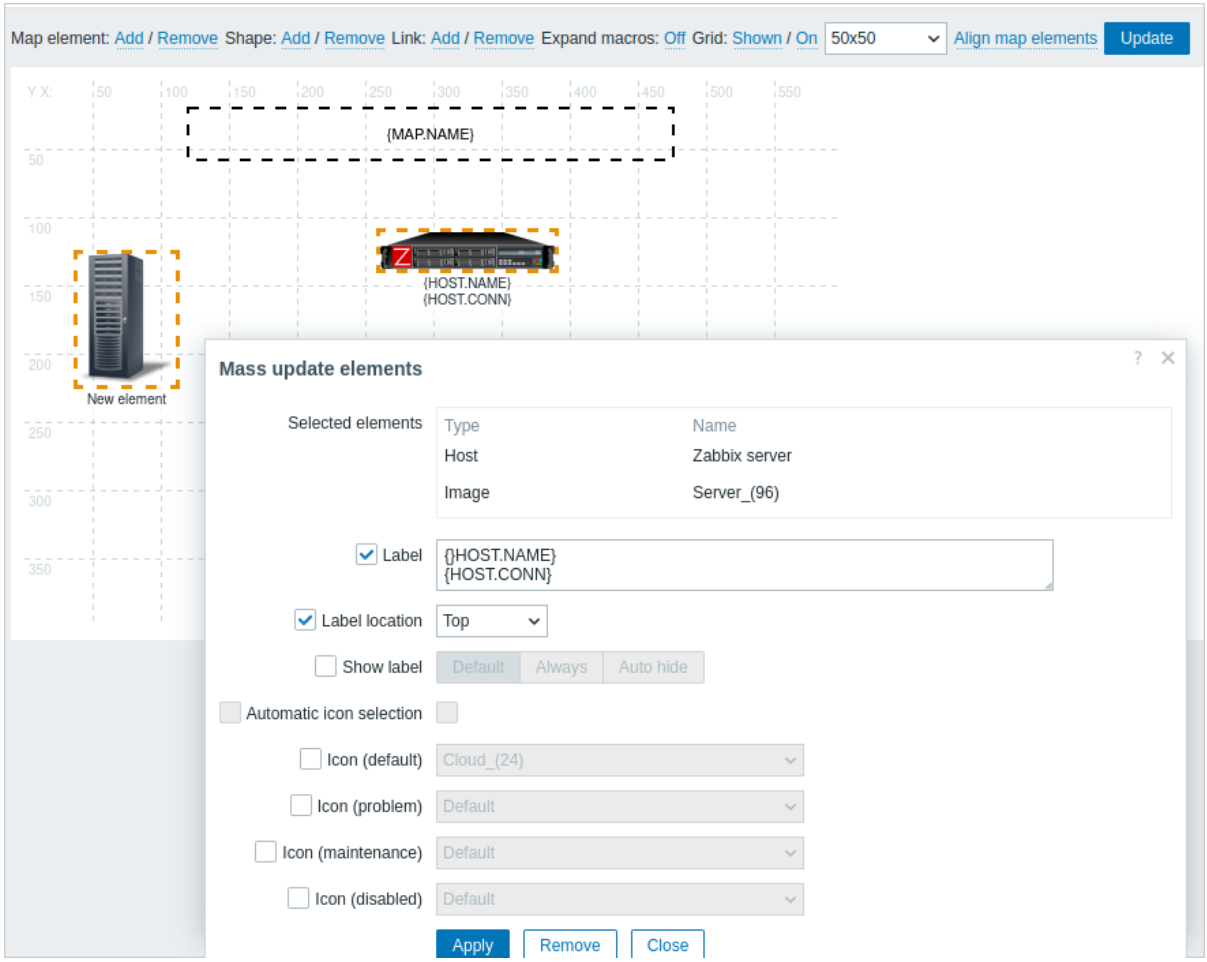
Attention:
Добавленные элементы не сохраняются автоматически. Если вы покинете страницу, все изменения могут быть потеряны.
Поэтому рекомендуется нажать кнопку **Update** в правом верхнем углу. После нажатия изменения будут сохранены независимо от того, что вы выберете во всплывающем окне.
Выбранные параметры сетки также сохраняются вместе с каждой картой.

Выбор элементов

Для выбора нескольких элементов, выберите сначала один элемент, затем нажмите и удерживайте *Ctrl*, чтобы выбрать другие.

Вы также можете выбрать несколько элементов, переместив прямоугольник в области редактирования и тем самым выбрав все элементы находящиеся в нем.

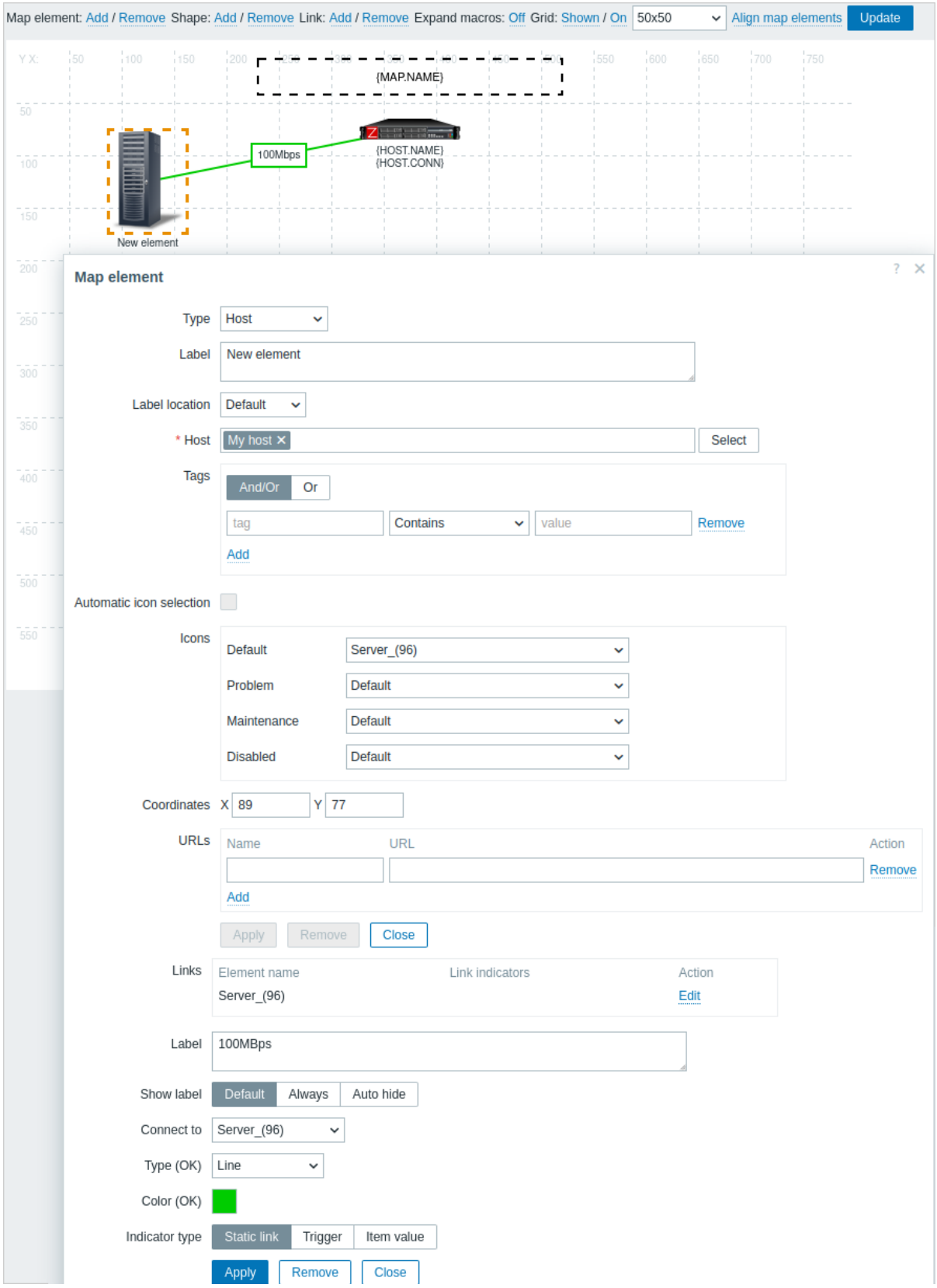
Как только вы выберите более одного элемента, диалог свойств элемента переключится в режим массового обновления, так что вы сможете изменить атрибуты выбранных элементов за раз. Чтобы это сделать, отметьте атрибут, используя флажок, и укажите для него новое значение. Здесь вы можете использовать макросы (такие как, скажем, {HOST.NAME} для подписи элемента).



Связывание элементов данных

После того как вы разместили на карте несколько элементов данных, можно приступить к их связыванию. Чтобы связать два элемента данных, сначала необходимо выбрать их. Когда элементы данных выбраны, нажмите *Add* рядом с Link.

После создания связи форма отдельного элемента данных будет содержать дополнительный раздел *Links*. Нажмите *Edit*, чтобы изменить атрибуты связи.



Атрибуты связи:

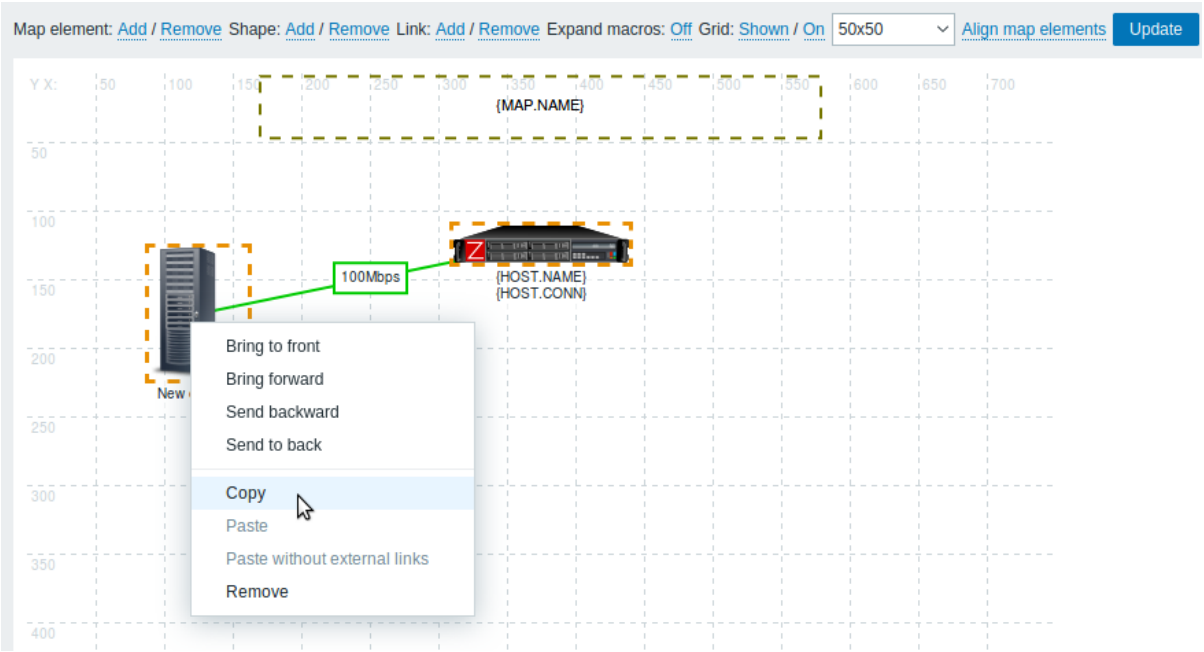
Параметр	Описание
Label	Метка, которая будет отображаться над связью. В этом поле поддерживаются выражения макросов, но только с функциями `avg`, `last`, `min` и `max`, где время указывается как параметр (например, `{?avg(/host/key,1h)}`).

Параметр	Описание
Show label	Выберите способ отображения метки связи: **Default** — использовать настройки *Show map element labels* из конфигурации карты (по умолчанию); **Always** — отображать всегда (по умолчанию); **Auto hide** — скрывать метку, если на неё не наведен курсор и она не выбрана.
Connect to	Элемент данных, 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, если для *Indicator type* выбрано *Item value*.
Indicators	Список индикаторов связи. Это поле доступно, если для *Indicator type* установлено значение *Trigger* или *Item value*. Если установлено *Trigger*, триггеры могут быть назначены как индикаторы связи. Когда триггер находится в состоянии *Problem*, к связи будет применён его стиль (выбранный цвет и тип линии). Если установлено *Item value* и в поле *Item* выбран элемент данных, значения элементов данных могут быть назначены как индикаторы связи. Когда значение элемента данных достигает указанного порога (для числового элемента данных) или соответствует шаблону (для текстового элемента данных), к связи применяется его стиль (выбранный цвет и тип линии).

Перемещение и копирование элементов

Можно **переместить** несколько выбранных элементов в другое место на карте, нажав на один из выбранных элементов затем, удерживая кнопку мыши, переместить курсор в желаемое место.

Можно **скопировать** один и более элементов, выбрав элементы, затем нажав на выбранный элемент правой кнопкой мыши и выбрать в меню *Копировать*.

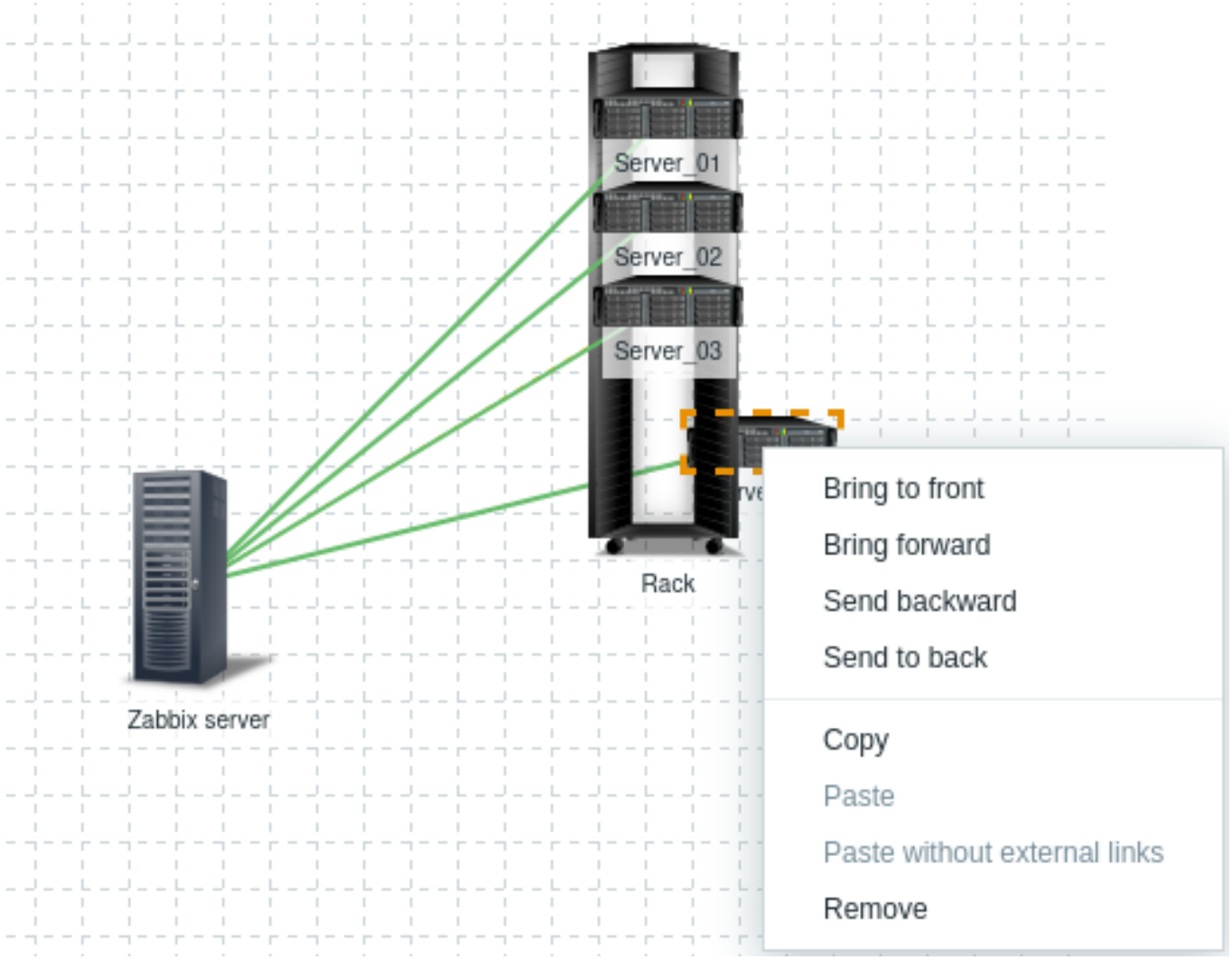


Чтобы вставить элементы, нажмите на область карты правой кнопкой мыши и выберите из меню *Вставить*. Опция *Вставить без внешних связей* позволяет вставить элементы только со связями между выбранными элементами.

Опция копирования-вставки работает в пределах одного окна браузера. Горячие клавиши не поддерживаются.

Порядок элементов

Чтобы переместить один элемент данных перед другим (или наоборот), щёлкните по элементу данных правой кнопкой мыши и выберите *Переместить вперёд*/*На передний план* или *Переместить назад*/*На задний план*.

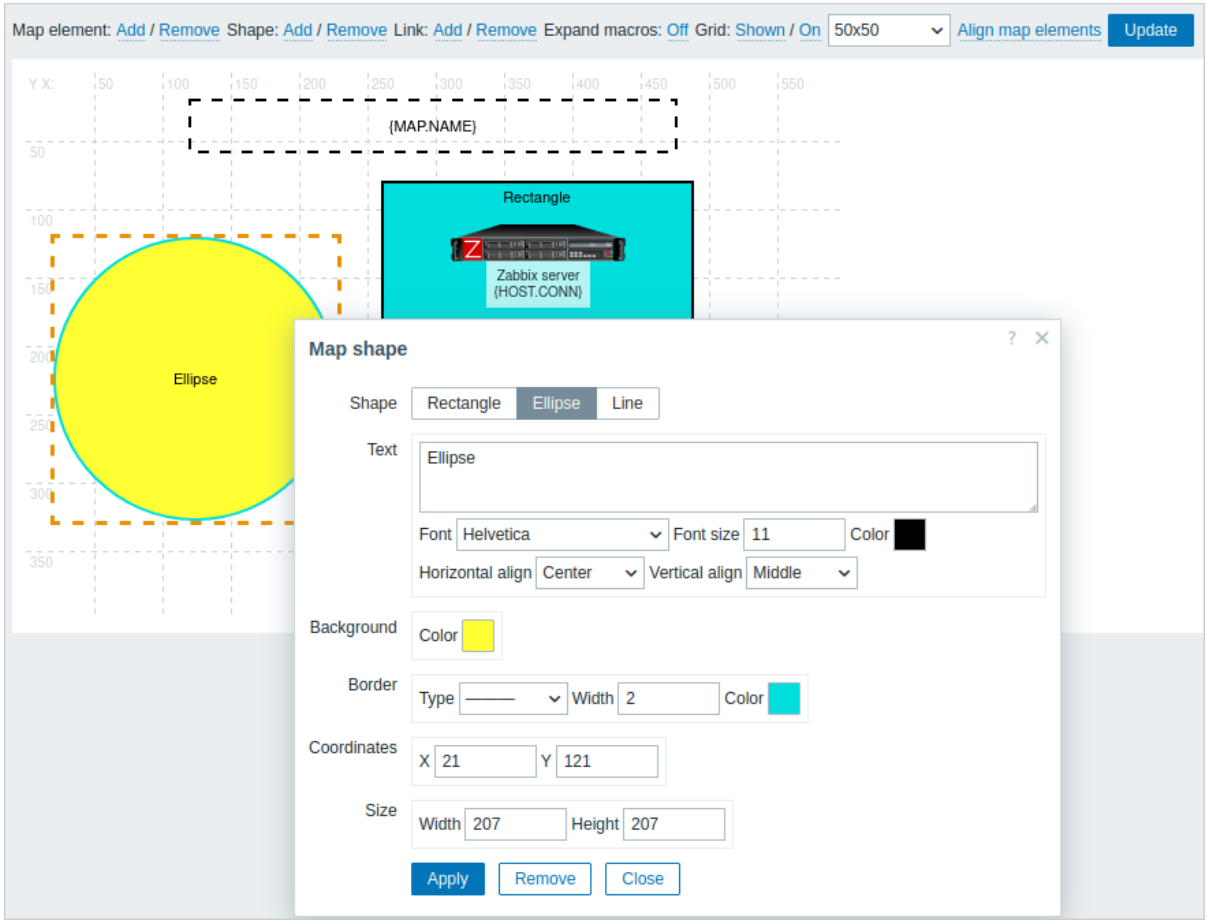


Добавление фигур

Помимо элементов карты, также можно добавлять фигуры. Фигуры не являются элементами карты; это лишь визуальное представление. Например, фигуру прямоугольника можно использовать как фон для группировки некоторых узлов сети. Можно добавлять фигуры прямоугольника и эллипса.

Чтобы добавить фигуру, нажмите *Добавить* рядом с Shape. Новая фигура появится в левом верхнем углу карты. Перетащите её в нужное место.

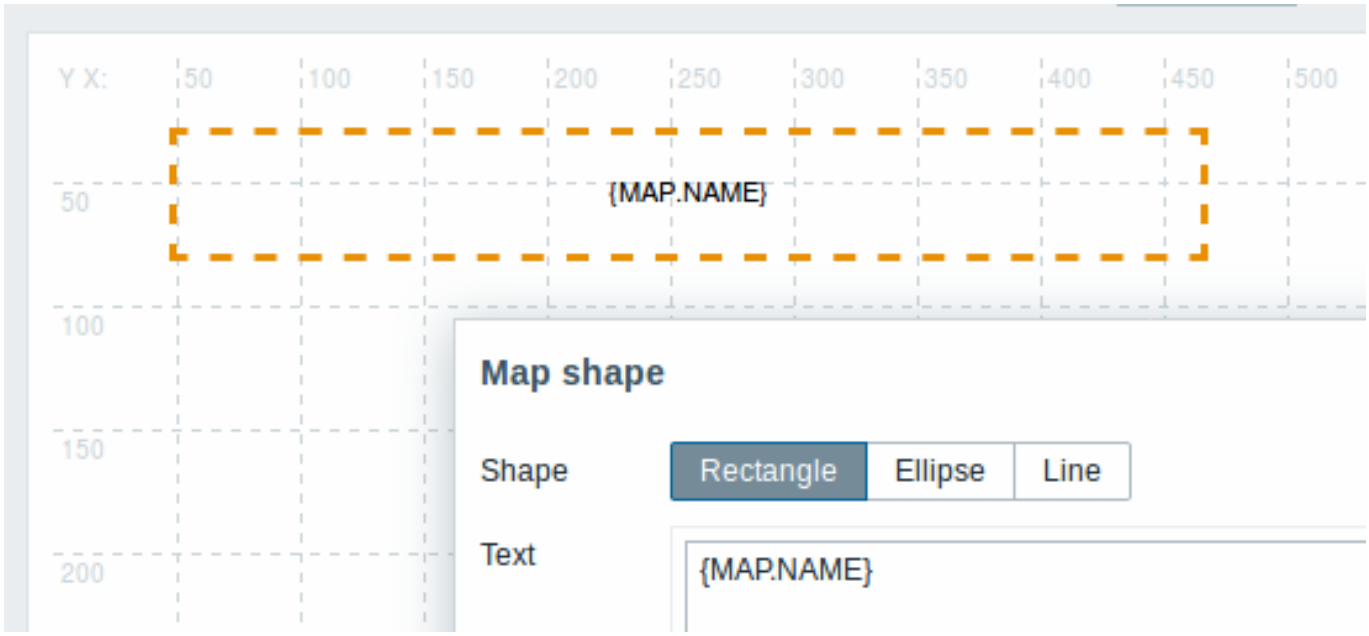
Новая фигура добавляется с цветами по умолчанию. При нажатии на фигуру отображается форма, в которой можно настроить её внешний вид, добавить текст и т. д.



Чтобы выбрать фигуры, выберите одну, а затем, удерживая *Ctrl*, выберите остальные. Если выбрано несколько фигур, их общие свойства можно массово обновить, аналогично элементам.

В фигуры можно добавлять текст. В тексте поддерживаются макросы-выражения, но только с функциями `avg`, `last`, `min` и `max`, где время задаётся как параметр (например, `{?avg(/host/key,1h)}`).

Чтобы отображался только текст, фигуру можно сделать невидимой, убрав границу фигуры (выберите 'None' в поле *Border*). Например, обратите внимание, что макрос {MAP.NAME}, видимый на снимке экрана выше, на самом деле является фигурой прямоугольника с текстом, что можно увидеть, нажав на макрос:



{MAP.NAME} преобразуется в настроенное имя карты при просмотре карты.

Если в тексте используются гиперссылки, при просмотре карты по ним можно щелкнуть.

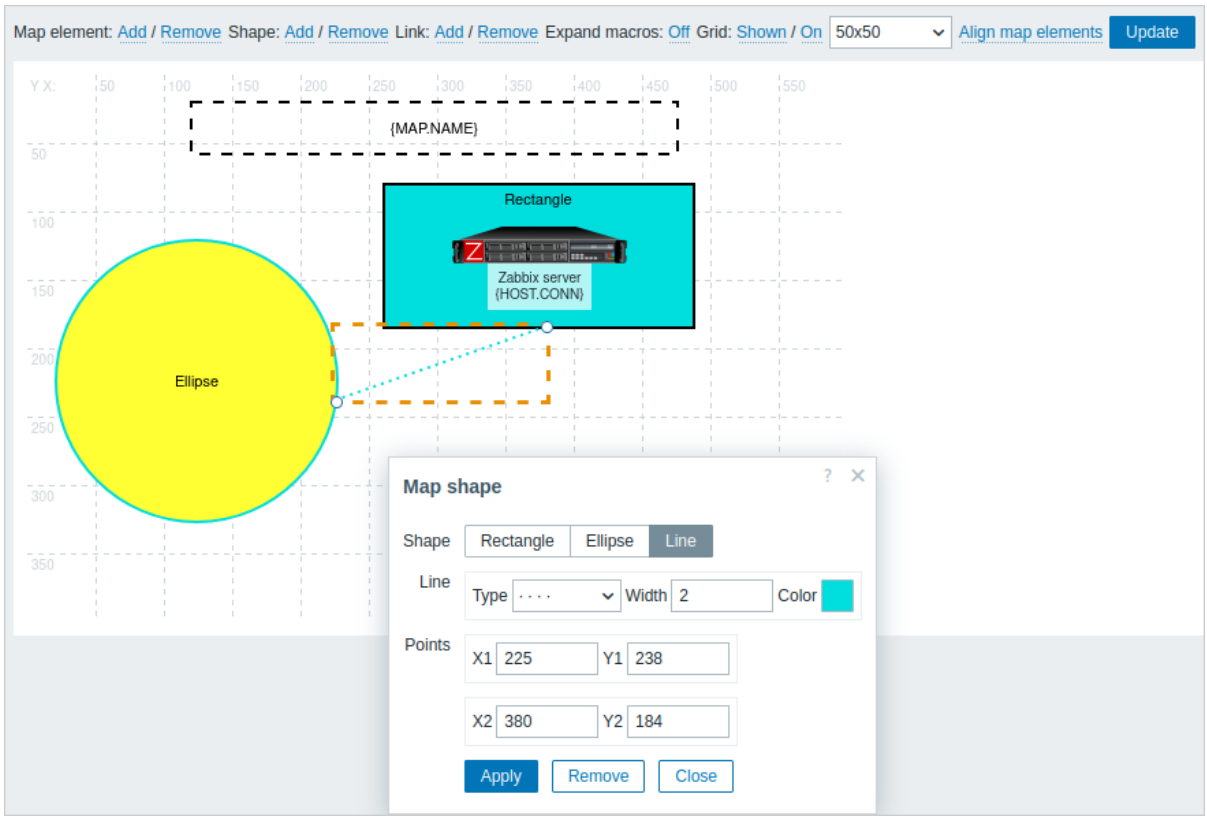
Перенос строк для текста внутри фигур всегда включён. Однако внутри эллипса строки переносятся так, как если бы эллипс был прямоугольником. Перенос по словам не реализован, поэтому длинные слова (слова, которые не помещаются

в фигуру) не переносятся, а маскируются (на странице редактирования карты) или обрезаются (на других страницах с картами).

Добавление линий

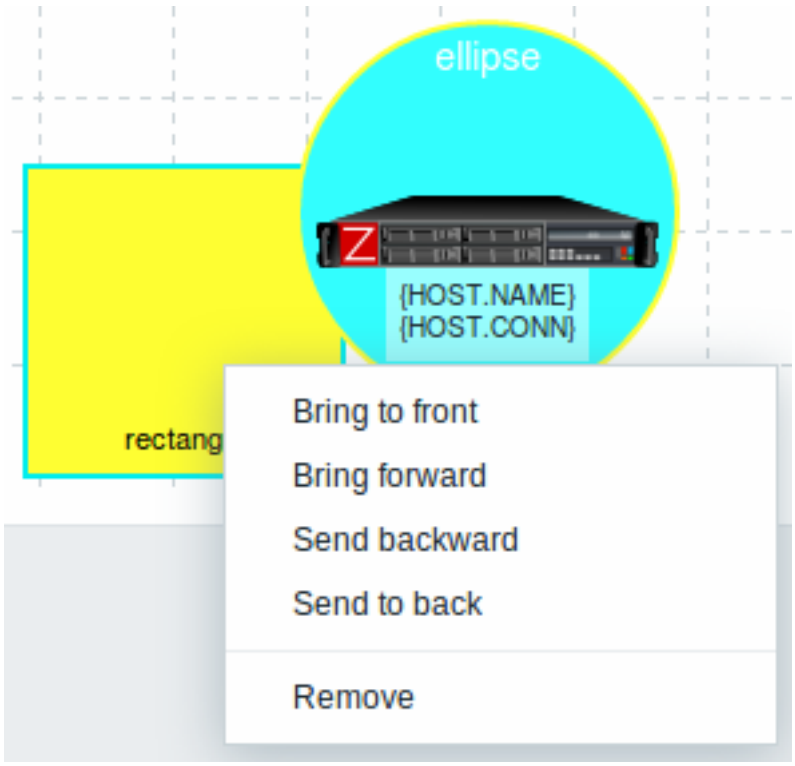
Помимо фигур, также можно добавлять линии. Линии можно использовать для связывания элементов или фигур на карте.

Чтобы добавить линию, нажмите *Добавить* рядом с Shape. Новая фигура появится в левом верхнем углу карты. Выберите ее и нажмите *Линия* в форме редактирования, чтобы изменить фигуру на линию. Затем настройте свойства линии, такие как тип линии, ширина, цвет и т. д.



Порядок фигур и линий

Чтобы переместить одну фигуру поверх другой (или наоборот), щёлкните по фигуре правой кнопкой мыши, чтобы открыть меню фигур карты.

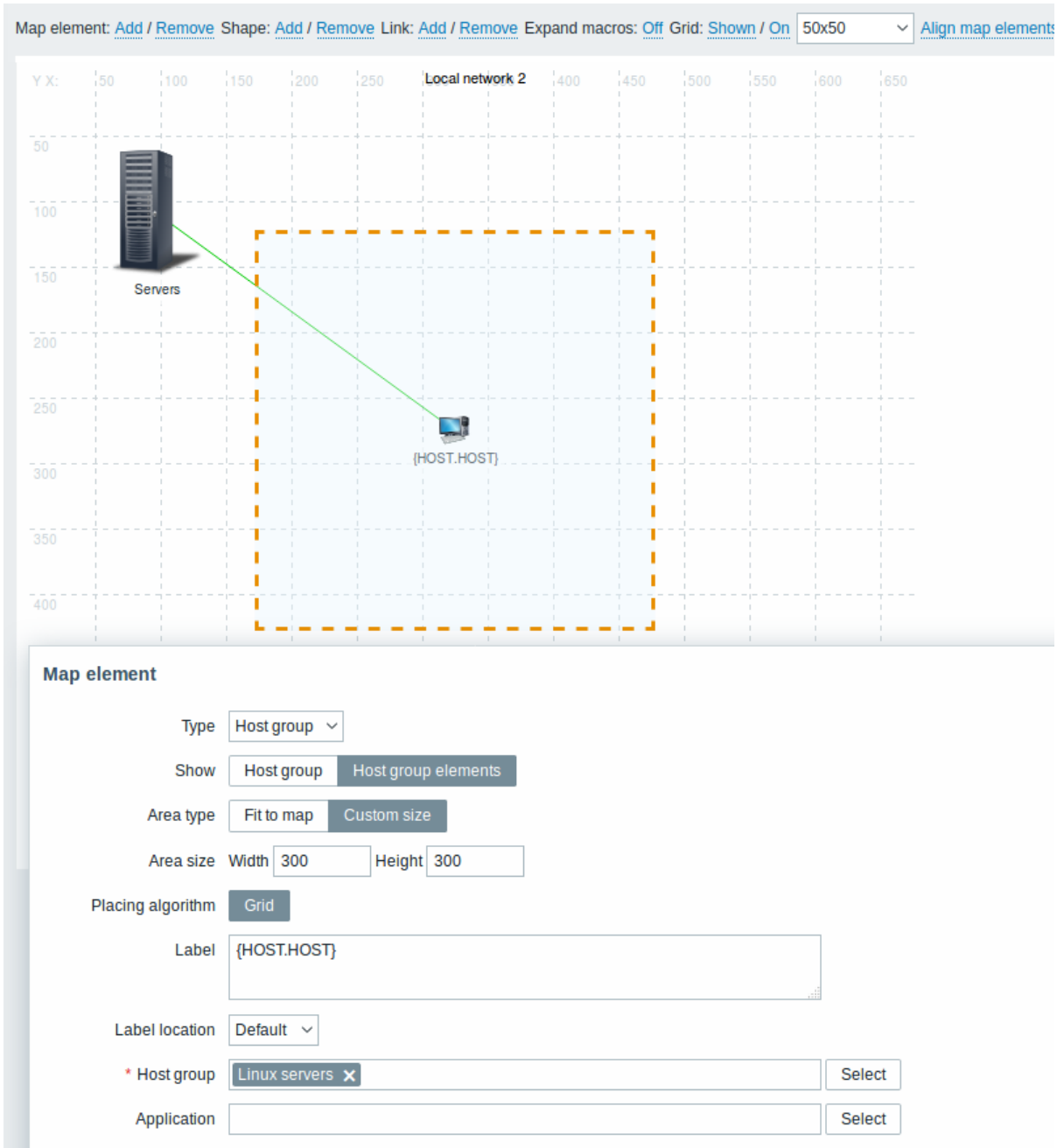


2 Элементы групп узлов сети

Обзор

В этом разделе объясняется, как добавить элемент типа *Группа узлов сети* при настройке карты сети.

Конфигурация



Все обязательные поля ввода отмечены красной звёздочкой.

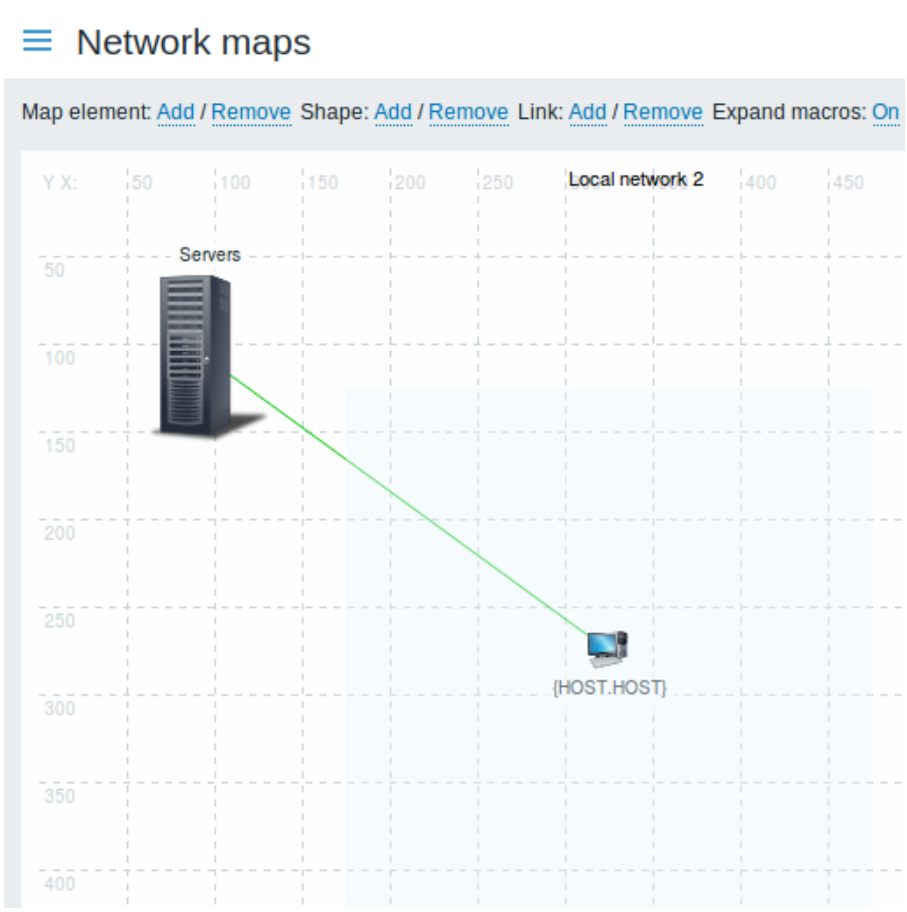
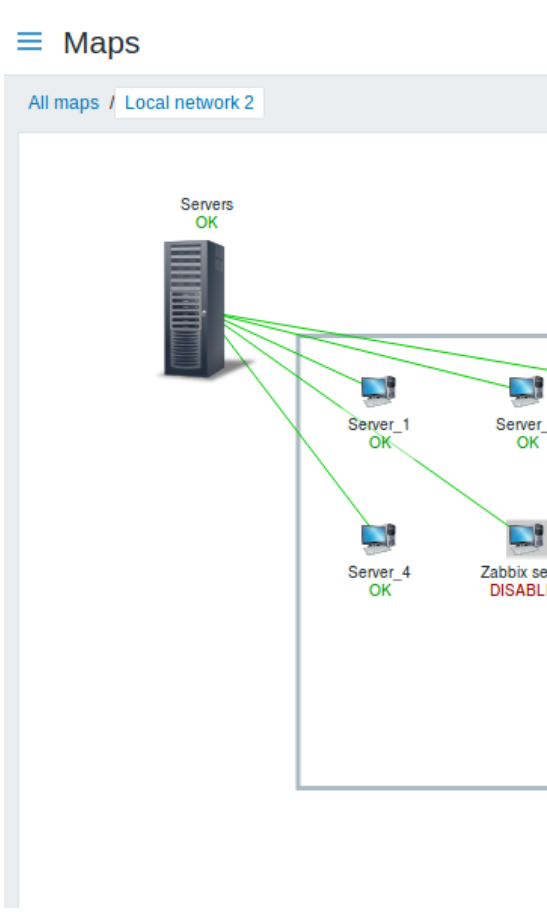
Эта таблица содержит параметры, типичные для типа элемента *Группа узлов сети*:

Параметр	Описание
Тип	Выберите тип элемента: **Группа узлов сети** — значок, представляющий состояние всех триггеров всех узлов сети, принадлежащих выбранной группе.
Показывать	Параметры отображения: **Группа узлов сети** — при выборе этого параметра будет показан один значок, отображающий соответствующую информацию об определённой группе узлов сети. **Элементы группы узлов сети** — при выборе этого параметра будет показано несколько значков, отображающих соответствующую информацию о каждом отдельном элементе (узле сети) определённой группы узлов сети.

Параметр	Описание
Тип области	Этот параметр доступен, если для *Показывать* установлено значение *Элементы группы узлов сети*: **Подогнать по карте** — 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 *Тип области* установлено значение *Пользовательский размер*: **Ширина** — значение в пикселях для указания ширины области карты. **Высота** — значение в пикселях для указания высоты области карты.
Алгоритм размещения	**Сетка** — единственный доступный вариант отображения всех элементов группы узлов сети.
Метка	Метка значка, любая строка. В метках можно использовать макросы и многострочные строки. Если тип элемента карты — *Группа узлов сети*, указание определённых макросов влияет на отображение карты, выводя соответствующую информацию о каждом отдельном узле сети. Например, если используется макрос {HOST.IP}, в режиме редактирования карты будет отображаться только сам макрос {HOST.IP}, тогда как в режиме просмотра карты будут включены и показаны уникальные IP-адреса каждого узла сети.

Просмотр элементов группы узлов сети

Эта опция доступна, если выбрана опция показа *Элементы группы узлов сети*.
При выборе *Элементов группы узлов сети* в качестве опции *показа* сначала вы увидите только один значок для группы узлов сети.
Однако после сохранения карты и перехода к просмотру карты вы увидите, что карта включает все элементы (узлы сети) определённой группы узлов сети:

Редактирование карты	Просмотр карты
	

Обратите внимание, как используется макрос {HOST.HOST}.

При редактировании карты имя макроса не раскрывается, тогда как в режиме просмотра карты отображаются все уникальные имена узлов сети.

3 Индикаторы связей

Обзор

Вы можете назначать индикаторы связи между элементами на карте сети.

Индикаторы могут основываться на триггерах или значениях элементов данных. Можно отображать различный стиль и цвет связи:

- 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 связи вы задаёте тип и цвет связи по умолчанию. Назначив индикаторы связи, можно сделать так, чтобы стиль и цвет связи зависели от состояния триггера или значения элемента данных.

Например, если любой из назначенных триггеров перейдёт в состояние проблемы, стиль и цвет связи изменятся соответствующим образом. Так, возможно, по умолчанию ваша связь была зелёной линией. Теперь, когда триггер находится в состоянии проблемы, связь может стать жирной красной (если вы так определили).

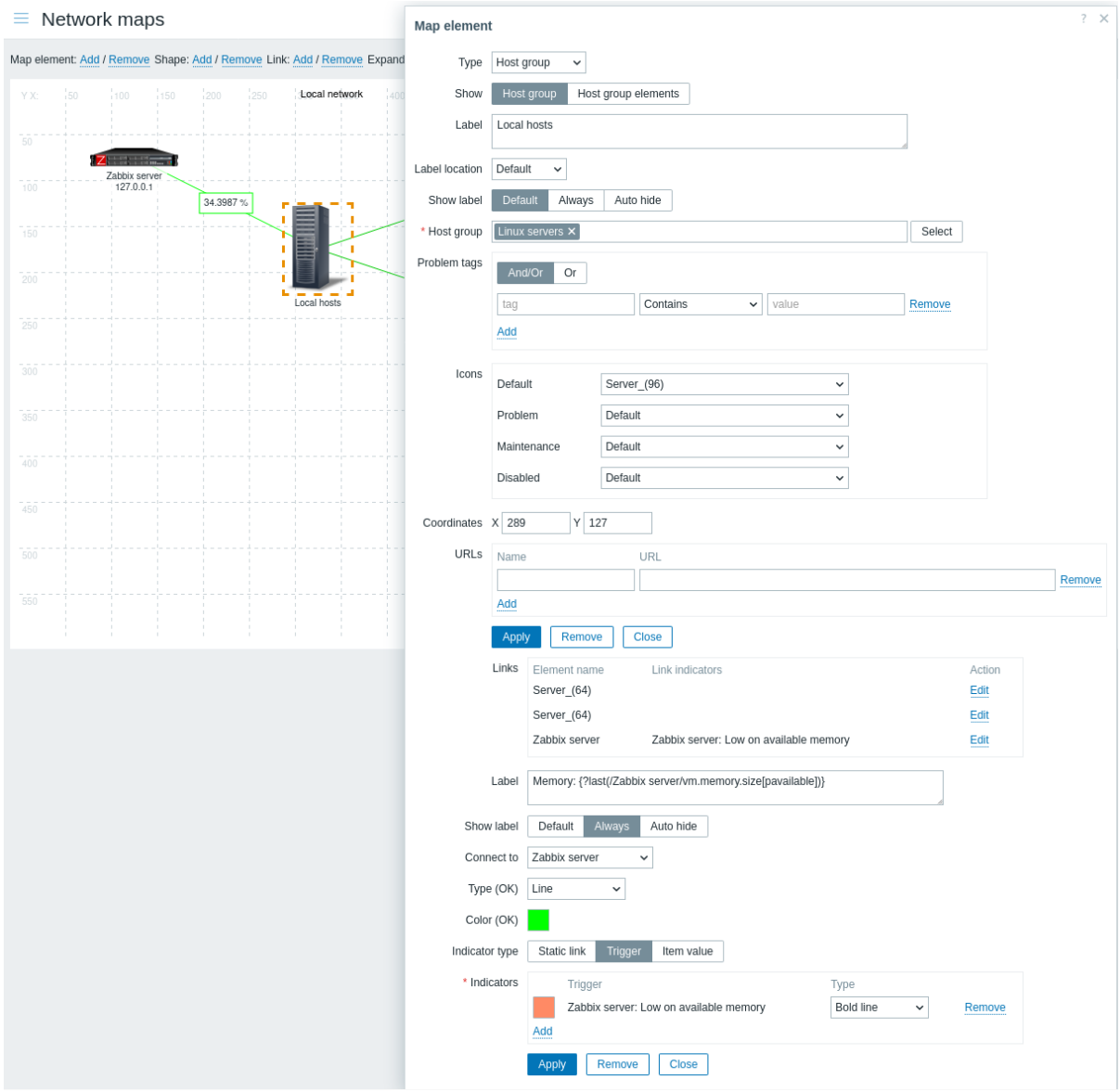
Аналогично, если значение элемента данных достигает указанного порогового значения или соответствует указанному регулярному выражению, стиль связи может измениться соответствующим образом.

Настройка

Триггеры

Чтобы назначить триггеры в качестве индикаторов связей, 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

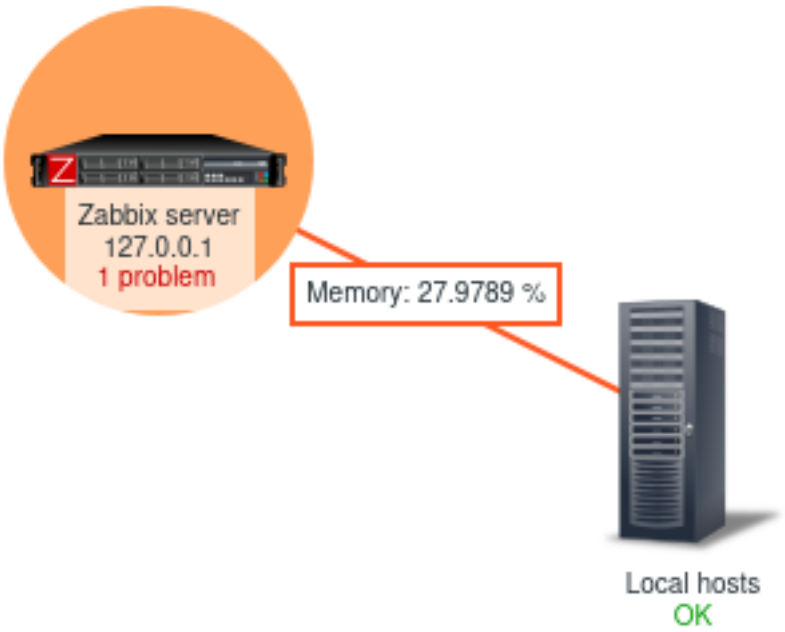


Все обязательные поля ввода отмечены красной звёздочкой.

Добавленные триггеры можно увидеть в списке *Indicators*.

Вы можете задать тип связи и цвет для каждого индикатора непосредственно из списка. После этого нажмите *Apply*, закройте форму и нажмите *Update*, чтобы сохранить изменения карты.

В разделе *Monitoring* → *Maps* будут отображаться соответствующие тип связи и цвет, если **триггер** перейдёт в состояние проблемы.



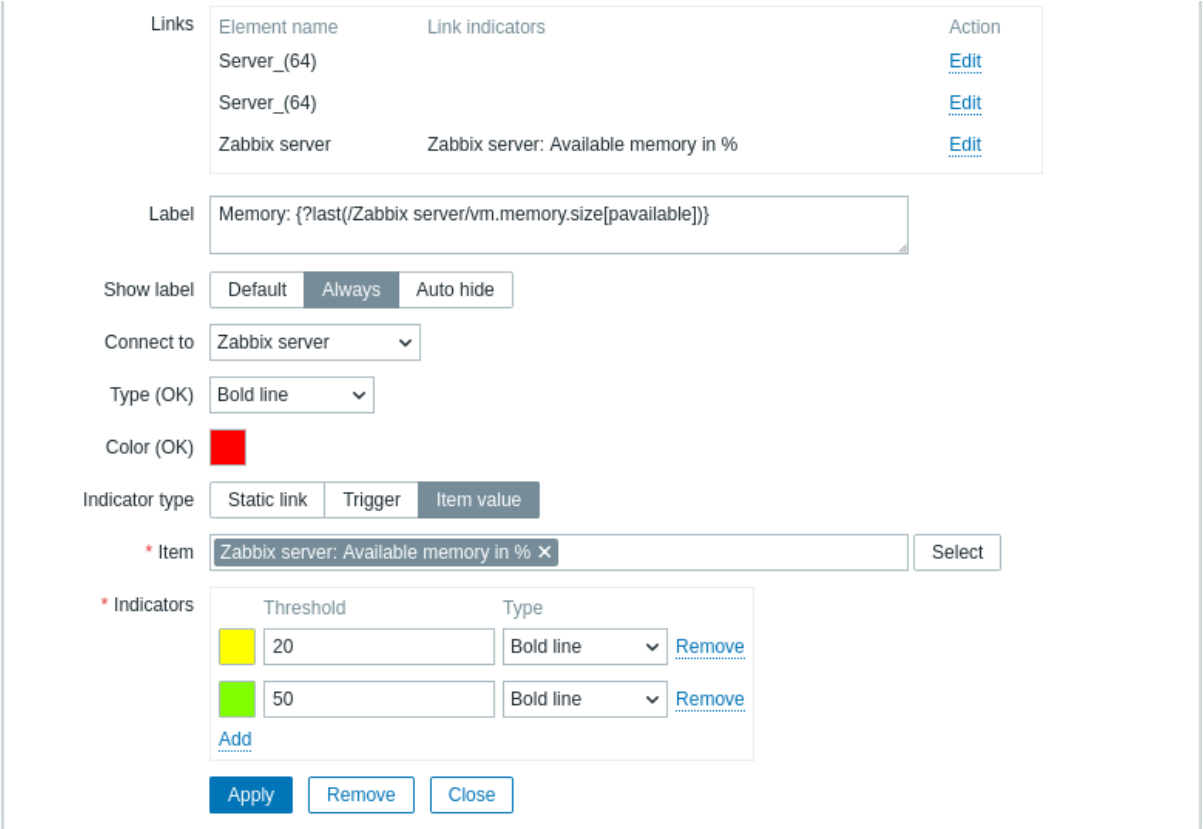
Note:
Если несколько триггеров перейдут в состояние проблемы, стиль и цвет связи будут определяться проблемой с наивысшей серьёзностью. Если несколько триггеров с одинаковой серьёзностью назначены одной и той же связи карты, приоритет имеет триггер с наименьшим ID. Также обратите внимание, что:

1. Настройки *Minimum trigger severity* и *Show suppressed problem* из конфигурации карты влияют на то, какие проблемы будут учитываться.
2. В случае триггеров с несколькими проблемами (multiple problem generation) каждая проблема может иметь серьёзность, отличающуюся от серьёзности триггера (изменённую вручную), может иметь разные теги (из-за макросов) и может быть подавлена.

Значения элементов данных

Чтобы назначить значения элементов данных в качестве индикаторов связей, выполните следующие действия:

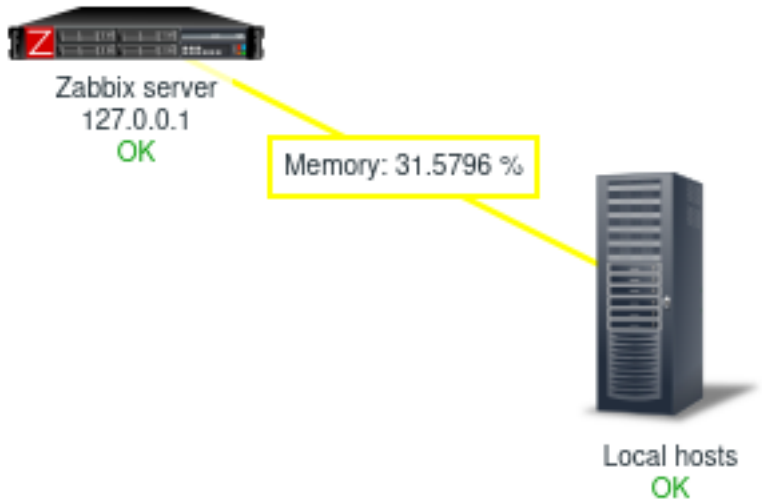
- выберите элемент карты
- нажмите *Edit* в разделе *Links* для соответствующей связи
- выберите *Item value* в качестве типа индикатора
- выберите элемент данных
- добавьте один или несколько порогов значений или шаблонов для элемента данных в блоке *Indicators*



Добавленные пороги/шаблоны элементов данных можно увидеть в списке *Indicators*.

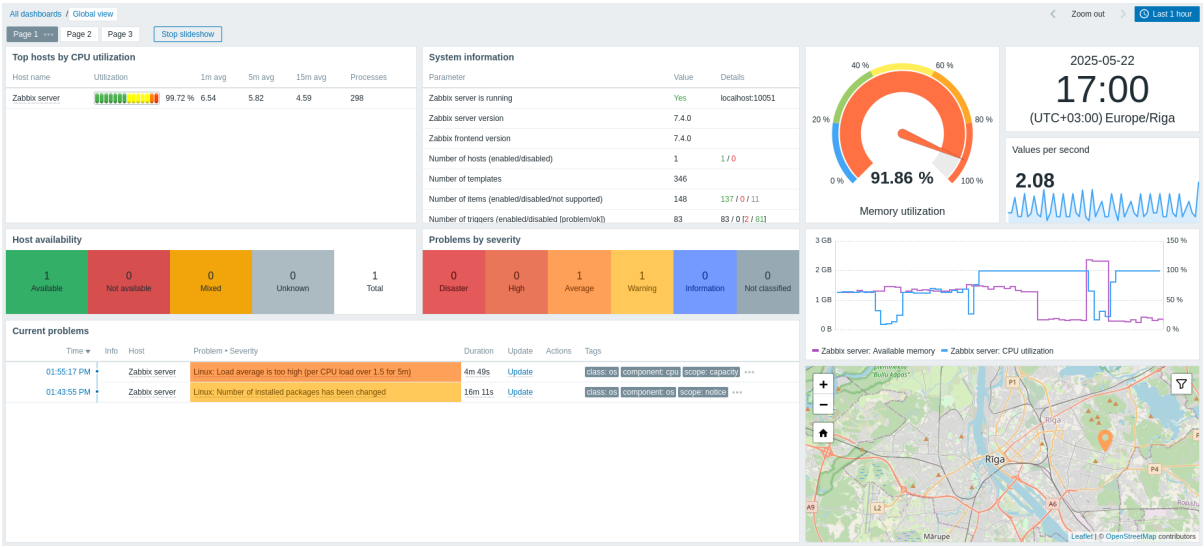
Вы можете задать тип связи и цвет для каждого индикатора непосредственно из списка. После этого нажмите *Apply*, закройте форму и нажмите *Update*, чтобы сохранить изменения карты.

В разделе *Monitoring* → *Maps* будут отображаться соответствующие тип связи и цвет, если **значение элемента данных** достигнет заданного порога (для числовых типов данных) или будет соответствовать шаблону регулярного выражения (для текстовых типов данных).



3 Панели

Панели — как глобальные панели, так и панели узлов сети — предоставляют мощную платформу визуализации с такими виджетами и инструментами, как современные графики, карты, слайд-шоу и многое другое.



8 Шаблоны и группы шаблонов

Обзор

Использование шаблонов — отличный способ уменьшить рабочую нагрузку и упростить настройку Zabbix. Шаблон — это набор сущностей, который можно удобно применять к нескольким узлам сети.

Сущности могут быть следующими:

- элементы данных
- триггеры
- графики
- панели
- правила низкоуровневого обнаружения
- веб-сценарии

Поскольку многие узлы сети в реальной жизни идентичны или достаточно похожи, вполне естественно, что набор сущностей (элементов данных, триггеров, графиков, ...) , который вы создали для одного узла сети, может быть полезен для многих. Конечно, вы можете копировать их на каждый новый узел сети, но это потребовало бы большого объема ручной работы. Вместо этого с помощью шаблонов вы можете скопировать их в один шаблон, а затем применить шаблон к любому количеству нужных узлов сети.

Когда шаблон привязывается к узлу сети, все сущности (элементы данных, триггеры, графики, ...) из шаблона добавляются к узлу сети. Шаблоны назначаются каждому отдельному узлу сети напрямую (а не группе узлов сети).

Шаблоны часто используются для группировки сущностей для определенных сервисов или приложений (например, Apache, MySQL, PostgreSQL, Postfix...), а затем применяются к узлам сети, на которых работают эти сервисы.

Еще одно преимущество использования шаблонов проявляется, когда нужно что-то изменить для всех узлов сети. Однократное изменение на уровне шаблона будет распространено на все связанные узлы сети.

Шаблоны организованы в группы шаблонов.

Перейдите к разделу создание и настройка шаблона.

9 Готовые шаблоны

Обзор

Zabbix предоставляет постоянно растущий набор предварительно настроенных шаблонов, чтобы упростить и ускорить настройку объектов мониторинга.

Все шаблоны, доступные «из коробки», находятся в разделе *Сбор данных* > *Шаблоны*.

При обновлении Zabbix существующие шаблоны не обновляются автоматически, чтобы избежать перезаписи пользовательских изменений. Чтобы обновить существующие шаблоны или добавить новые, смотрите Обновление шаблонов.

Используйте боковую панель для доступа к информации о конкретных типах шаблонов и требованиях к их работе.

Смотрите также:

- Привязка шаблона
- Известные проблемы шаблонов

Обновление шаблона

Чтобы обновить один шаблон или добавить новый после обновления Zabbix:

1. Перейдите в Git-репозиторий Zabbix.
2. В селекторе ветки (под меткой *Source*) выберите ветку, соответствующую вашей версии Zabbix (например, *master* для последней разрабатываемой версии).
3. Откройте шаблон, который хотите добавить или обновить (например, MySQL by Zabbix агент).
4. Скопируйте содержимое файла шаблона (например, `template_db_mysql_agent.yaml`) и сохраните его в локальный файл.
5. В веб-интерфейс Zabbix перейдите в *Сбор данных* > *Шаблоны* и импортируйте локальный файл.

Вы также можете обновить все шаблоны и добавить новые сразу:

1. Загрузите пакет шаблонов в зависимости от вашего веб-сервера: `templates_1M_8.0.7z` (Nginx) или `templates_2M_8.0.7z` (Apache).
2. Распакуйте пакет. Он содержит YAML-файлы (каждый с несколькими шаблонами) и индекс (`index-1M_80.md`) со списком шаблонов, включенных в каждый YAML-файл.
3. В веб-интерфейс Zabbix перейдите в *Сбор данных* > *Шаблоны* и импортируйте YAML-файлы по одному.

Кроме того, вы можете импортировать все YAML-файлы сразу с помощью скрипта `import_templates.sh`. Для работы скрипта требуются утилиты `curl` и `jq`, а также используется метод Zabbix API `configuration.import`. При запуске скрипта укажите каталог, содержащий YAML-файлы, а также ваш URL Zabbix API и токен аутентификации.

> **Warning:**
> Скрипт `import_templates.sh` перезаписывает все существующие шаблоны без возможности просмотра или отмены изменений шаблонов.

Пример:

```
./import_templates.sh /tmp/templates_2M_8.0 https://example.com/zabbix/api_jsonrpc.php d8d6b5c78ee2a8333db
```

Совместимость шаблонов с Host Wizard

Host Wizard совместим со всеми шаблонами, доступными «из коробки».

Вы также можете сделать свои пользовательские шаблоны совместимыми с Host Wizard:

1. В веб-интерфейсе Zabbix перейдите в *Сбор данных* > *Шаблоны* и экспортируйте пользовательский шаблон.
2. Отредактируйте экспортированный файл:
   - Добавьте элемент шаблона `wizard_ready`, чтобы указать совместимость с Host Wizard.
   - При необходимости также добавьте элемент шаблона `readme` или `config`.
3. В веб-интерфейсе Zabbix перейдите в *Сбор данных* > *Шаблоны* и импортируйте обновленный файл шаблона.

1 Работа шаблонов Zabbix агент

Шаги для обеспечения корректной работы шаблонов, которые собирают метрики с помощью Zabbix агент:

1. Убедитесь, что Zabbix агент установлен на узле сети. Для активных проверок также убедитесь, что адрес Zabbix сервер/прокси добавлен в параметр 'ServerActive' файла конфигурации агента.
2. Привяжите шаблон к целевому узлу сети (если шаблон недоступен в вашей установке Zabbix, возможно, сначала потребуется импортировать шаблон).
3. При необходимости скорректируйте значения макросов шаблона.
4. Настройте контролируемый экземпляр так, чтобы он разрешал совместное использование данных с Zabbix.

Подробное описание шаблона, включая полный список макросов, элементов данных и триггеров, доступно в файле README шаблона (доступен по нажатию на 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
- 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
- PHP-FPM by Zabbix agent
- PHP-FPM by Zabbix agent active
- PostgreSQL by Zabbix agent
- PostgreSQL by Zabbix agent active
- RabbitMQ cluster by Zabbix agent

2 Работа шаблонов Zabbix agent 2

Шаги для обеспечения корректной работы шаблонов, которые собирают метрики с помощью Zabbix agent 2:

1. Убедитесь, что agent 2 установлен на узле сети и что установленная версия содержит необходимый плагин. В некоторых случаях может потребоваться сначала обновить agent 2.

2. Привяжите шаблон к целевому узлу сети (если шаблон недоступен в вашей установке Zabbix, возможно, сначала потребуется импортировать шаблон).

3. При необходимости скорректируйте значения макросов шаблона. Обратите внимание, что пользовательские макросы могут использоваться для переопределения параметров конфигурации.

4. Настройте контролируемый экземпляр так, чтобы он разрешал совместное использование данных с Zabbix.

Attention:
Шаблоны Zabbix agent 2 работают совместно с плагинами. Хотя базовую настройку можно выполнить, просто скорректировав пользовательские макросы, более глубокая настройка может быть достигнута путем настройки самого плагина. Например, если плагин поддерживает именованные сессии, можно отслеживать несколько однотипных сущностей (например, MySQL1 и MySQL2), указав именованную сессию с собственными URI, именем пользователя и паролем для каждой сущности в файле конфигурации.

Подробное описание шаблона, включая полный список макросов, элементов данных и триггеров, доступно в файле README шаблона (доступен по щелчку по имени 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

3 Работа HTTP-шаблонов

Шаги для обеспечения корректной работы шаблонов, которые собирают метрики с помощью HTTP agent:

1. Создайте в Zabbix узел сети, соответствующий объекту мониторинга. Укажите IP-адрес/DNS-имя объекта мониторинга в качестве основного интерфейса узла сети. Это необходимо, чтобы макрос {HOST.CONN} корректно разрешался в элементах данных шаблона.

2. Привяжите шаблон к узлу сети, созданному на шаге 1 (если шаблон недоступен в вашей установке Zabbix, возможно, сначала потребуется импортировать шаблон).

3. При необходимости скорректируйте значения макросов на уровне шаблона.

4. Настройте контролируемый экземпляр так, чтобы он разрешал совместное использование данных с Zabbix.

Подробное описание шаблона, включая полный список макросов, элементов данных и триггеров, доступно в файле README шаблона (доступен по щелчку на имени 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
- 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
- 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
- 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
- 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

4 Работа шаблонов IPMI

Шаблоны IPMI не требуют какой-либо специальной настройки. Чтобы начать мониторинг, привяжите шаблон к целевому узлу сети (если шаблон недоступен в вашей установке Zabbix, возможно, сначала потребуется импортировать шаблон).

Подробное описание шаблона, включая полный список макросов, элементов данных и триггеров, доступно в файле README шаблона (доступен по нажатию на имя шаблона).

Доступный шаблон:

- Chassis by IPMI

5 Работа шаблонов JMX

Шаги для обеспечения корректной работы шаблонов, которые собирают метрики через JMX:

1. Убедитесь, что Zabbix Java gateway установлен и настроен должным образом.
2. Привяжите шаблон к целевому узлу сети. На узле сети должен быть настроен интерфейс JMX. Если шаблон недоступен в вашей установке Zabbix, возможно, сначала потребуется 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 по щелчку на имени 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

6 Работа шаблонов 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 скорректируйте значения макросов шаблона.

4. Настройте контролируемый экземпляр так, чтобы он разрешал совместное использование данных с Zabbix.

Подробное описание шаблона, включая полный список макросов, элементов данных и триггеров, доступно в файле README шаблона (доступен по нажатию на имя шаблона).

Доступны следующие шаблоны:

- MariaDB by ODBC
- MSSQL by ODBC
- MySQL by ODBC
- Oracle by ODBC
- Percona by ODBC
- PostgreSQL by ODBC

7 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.

Предоставляются шаблоны для многих семейств сетевых устройств. Все шаблоны охватывают (там, где возможно получить эти элементы данных с устройства):

- Мониторинг неисправностей шасси (блоки питания, вентиляторы и температура, общее состояние)
- Мониторинг производительности шасси (элементы данных CPU и памяти)
- Сбор инвентарных данных шасси (серийные номера, название модели, версия прошивки)
- Мониторинг сетевых интерфейсов с использованием IF-MIB и EtherLike-MIB (состояние интерфейса, нагрузка трафика интерфейса, статус дуплекса для Ethernet)

Если шаблон недоступен в вашей установке Zabbix, возможно, сначала потребуется импортировать шаблон.

Если вы импортируете новые шаблоны, входящие в поставку, возможно, вам также потребуется обновить глобальное регулярное выражение `@Network interfaces for discovery` до следующего вида:

```
Result is FALSE: ^Software Loopback Interface
Result is FALSE: ^(In)?[lL]oop[bB]ack[0-9._]*$
Result is FALSE: ^NULL[0-9.]*$
```

```
Result is FALSE: ^[lL]o[0-9.]*$
Result is FALSE: ^[sS]ystem$
Result is FALSE: ^Nu[0-9.]*$
```

чтобы отфильтровать loopback- и null-интерфейсы в большинстве систем.

Устройства

Список семейств устройств, для которых доступны шаблоны:

Имя шаблона	Производитель	Семейство устройств	Известные модели	ОС	Используемые MIB	**Теги**
Alcatel Timetra TiMOS SNMP	Alcatel	Alcatel Timetra	ALCATEL SR 7750	TiMOS	TIMETRA-SYSTEM-MIB,TIMETRA-CHASSIS-MIB	Сертифицирован
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	Сертифицирован
Brocade FC SNMP	Brocade	FC-коммутаторы Brocade	Brocade 300 SAN Switch-	-	SW-MIB,ENTITY-MIB	Производительность Сбой
Brocade_Foundry Stackable SNMP	Brocade	Brocade ICX	Brocade ICX6610, Brocade ICX7250-48, Brocade ICX7450-48F		FOUNDRY-SN-AGENT-MIB, FOUNDRY-SN-STACKING-MIB	Сертифицирован
Brocade_Foundry Nonstackable SNMP	Brocade, Foundry	Brocade MLX, Foundry	Brocade MLXe, Foundry FLS648, Foundry FWSX424		FOUNDRY-SN-AGENT-MIB	Производительность Сбой
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	Сертифицирован
Ciena 3906 by SNMP	Ciena	CPE Ciena	Ciena 3906	SAOS	WWP-LEOS-BLADE-MIB, WWP-LEOS-CHASSIS-MIB, WWP-LEOS-SYSTEM-CONFIG-MIB	Сертифицирован
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	Сертифицирован
Cisco IOS SNMP	Cisco	Cisco IOS версий > 12.2 3.5	Cisco C2950	IOS	CISCO-PROCESS-MIB,CISCO-MEMORY-POOL-MIB,CISCO-ENVMON-MIB	Сертифицирован

Имя шаблона	Производитель	Семейство устройств	Известные модели	ОС	Используемые MIB	**Теги**
Cisco IOS versions 12.0_3_T-12.2_3.5 SNMP	Cisco	Cisco IOS > 12.0 3 T и 12.2 3.5	-	IOS	CISCO-PROCESS-MIB,CISCO-MEMORY-POOL-MIB,CISCO-ENVMON-MIB	Сертифицирован
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	Сертифицирован
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	Сертифицирован
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 Сбой Интерфейсы
Dell Force S-Series SNMP	Dell	Dell Force S-Series	S4810		F10-S-SERIES-CHASSIS-MIB	Сертифицирован
Extreme Exos SNMP	Extreme	Extreme EXOS	X670V-48x	EXOS	EXTREME-SYSTEM-MIB,EXTREME-SOFTWARE-MONITOR-MIB	Сертифицирован
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	Сертифицирован
HP Enterprise Switch SNMP	HP	HP Enterprise Switch	HP ProCurve J4900B Switch 2626, HP J9728A 2920-48G Switch		STATISTICS-MIB,NETSWITCH-MIB,HP-ICF-CHASSIS,ENTITY-MIB,SEMI-MIB	Сертифицирован
Huawei OceanStor 5300 V5 by SNMP	Huawei	Huawei OceanStor Dorado	Huawei OceanStor Dorado with V5 software		HUAWEI-STORAGE-HARDWARE-MIB, HUAWEI-STORAGE-SPACE-MIB, ISM-PERFORMANCE-MIB, HOST-RESOURCES-MIB, SNMPv2-MIB	Сертифицирован

Имя шаблона	Производитель	Семейство устройств	Известные модели	ОС	Используемые MIB	**Теги**
Huawei OceanStor Dorado by SNMP	Huawei	Huawei OceanStor, Huawei OceanStor Dorado	Huawei OceanStor Dorado 3000/5000/6000/8000/18000/5300/5500/5600/6800/18500/18800 with V6/V700 software		HUAWEI-STORAGE-HARDWARE-MIB, HUAWEI-STORAGE-SPACE-MIB, ISM-PERFORMANCE-MIB, HOST-RESOURCES-MIB, SNMPv2-MIB	Сертифицирован
Huawei VRP by SNMP	Huawei	Huawei VRP	S2352P-EI		ENTITY-MIB,HUAWEI-ENTITY-EXTENT-MIB	Сертифицирован
Intel_Qlogic Infiniband SNMP	Intel/QLogic	Устройства In-tel/QLogic Infini-band	Infiniband 12300		ICS-CHASSIS-MIB	Сбой Инвентаризация
Juniper SNMP	Juniper	Модели MX, SRX, EX	Juniper MX240, Juniper EX4200-24F	JunOS	JUNIPER-MIB	Сертифицирован
Juniper MX NETCONF	Juniper	Модели MX	Juniper MX204 Edge Router	JunOS 24.2R1-S1.10	*Неприменимо (используется NETCONF через SSH)*	Сертифицирован
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	Сертифицирован
Mellanox SNMP	Mellanox	Устройства Mellanox Infini-band	SX1036	MLNX-OS	HOST-RESOURCES-MIB,ENTITY-MIB,ENTITY-SENSOR-MIB,MELLANOX-MIB	Сертифицирован

Имя шаблона	Производитель	Семейство устройств	Известные модели	ОС	Используемые MIB	**Теги**
MikroTik CCR<device model> SNMP	MikroTik	Маршрутизаторы MikroTik Cloud Core (серия CCR)	Доступны отдельные специализированные шаблоны 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	Сертифицирован

Имя шаблона	Производитель	Семейство устройств	Известные модели	ОС	Используемые MIB	**Теги**
MikroTik CRS<device model> SNMP	MikroTik	MikroTik Cloud Router Switches (серия CRS)	Доступны отдельные специализированные шаблоны 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	Сертифицирован
MikroTik CSS<device model> SNMP	MikroTik	MikroTik Cloud Smart Switches (серия CSS)	Доступны отдельные специализированные шаблоны для MikroTik CSS326-24G-2S+RM, MikroTik CSS610-8G-2S+IN	RouterOS	MIKROTIK-MIB,HOST-RESOURCES-MIB	Сертифицирован
MikroTik FiberBox SNMP	MikroTik	MikroTik FiberBox	MikroTik FiberBox	RouterOS	MIKROTIK-MIB,HOST-RESOURCES-MIB	Сертифицирован
MikroTik hEX <device model> SNMP	MikroTik	MikroTik hEX	Доступны отдельные специализированные шаблоны для MikroTik hEX, MikroTik hEX lite, MikroTik hEX PoE, MikroTik hEX PoE lite, MikroTik hEX S	RouterOS	MIKROTIK-MIB,HOST-RESOURCES-MIB	Сертифицирован

Имя шаблона	Производитель	Семейство устройств	Известные модели	ОС	Используемые MIB	**Теги**
MikroTik netPower <device model> SNMP	MikroTik	MikroTik net-Power	Доступны отдельные специализированные шаблоны для MikroTik netPower 15FR, MikroTik netPower 16P SNMP, MikroTik netPower Lite 7R	RouterOS/SwitchOS, SwitchOS Lite	MIKROTIK-MIB,HOST-RESOURCES-MIB	Сертифицирован
MikroTik PowerBox <device model> SNMP	MikroTik	MikroTik Power-Box	Доступны отдельные специализированные шаблоны для MikroTik PowerBox, MikroTik PowerBox Pro	RouterOS	MIKROTIK-MIB,HOST-RESOURCES-MIB	Сертифицирован
MikroTik RB<device model> SNMP	MikroTik	Маршрутизаторы MikroTik серии RB	Доступны отдельные специализированные шаблоны 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	Сертифицирован
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	Сертифицирован
Netgear Fastpath SNMP	Netgear	Netgear Fastpath	M5300-28G		FASTPATH-SWITCHING-MIB,FASTPATH-BOXSERVICES-PRIVATE-MIB	Сбой Инвентаризация

Имя шаблона	Производитель	Семейство устройств	Известные модели	ОС	Используемые MIB	Теги
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	Сертифицирован
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

Проектирование шаблонов

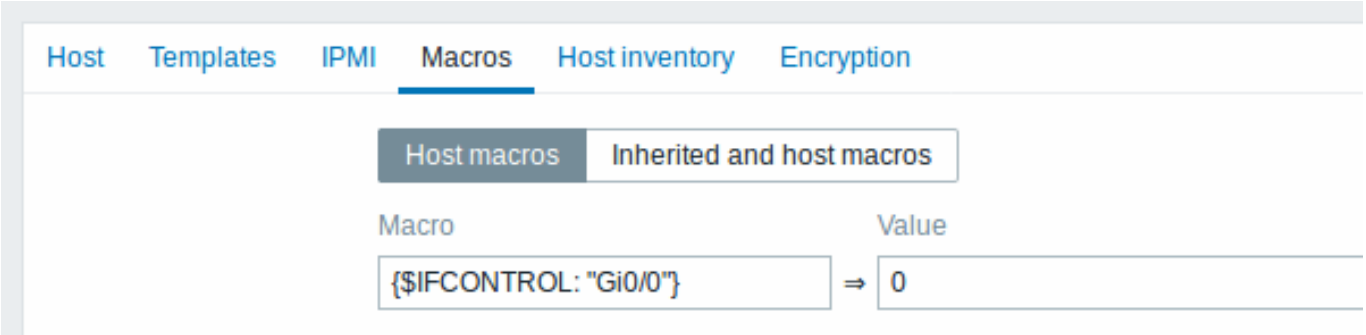
Шаблоны были разработаны с учетом следующего:

- Пользовательские макросы используются максимально широко, чтобы пользователь мог настраивать триггеры;
- Низкоуровневое обнаружение используется максимально широко, чтобы минимизировать количество неподдерживаемых элементов данных;
- Все шаблоны зависят от Template ICMP Ping, поэтому все устройства также проверяются по ICMP;
- Элементы данных не используют никаких MIB — в элементах данных и правилах низкоуровневого обнаружения используются SNMP OID. Поэтому для работы шаблонов не требуется загружать какие-либо MIB в Zabbix;
- При обнаружении отфильтровываются loopback-сетевые интерфейсы, а также интерфейсы с ifAdminStatus = down(2)
- По возможности используются 64-битные счетчики из IF-MIB::ifXTable. Если они не поддерживаются, вместо них используются стандартные 32-битные счетчики.

Для всех обнаруженных сетевых интерфейсов имеется триггер, который отслеживает их рабочее состояние (link), например:

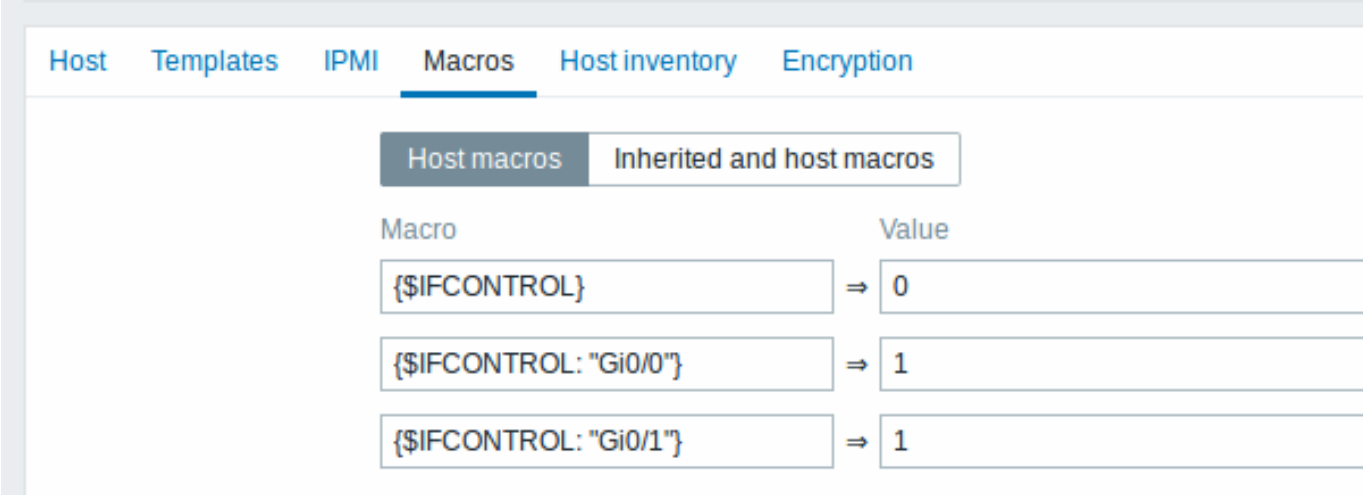
```
{$IFCONTROL:"{#IFNAME}"}=1 and last(/Alcatel Timetra TiMOS SNMP/net.if.status[ifOperStatus.{#SNMPINDEX}])
```

- Если вы не хотите отслеживать это состояние для определенного интерфейса, создайте пользовательский макрос с контекстом и значением 0. Например:



где Gi0/0 — это {#IFNAME}. Таким образом, триггер больше не будет использоваться для этого конкретного интерфейса.

- Вы также можете изменить поведение по умолчанию так, чтобы все триггеры не срабатывали, и активировать этот триггер только для ограниченного числа интерфейсов, например uplink:



Теги

- Performance (Прозводительность) – MIB'ы семейства устройств дают возможность мониторинга CPU и элементов данных 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.

8 Работа 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.

Настройка макросов узла сети

Чтобы использовать простые проверки VMware, на узле сети должны быть определены следующие пользовательские макросы:

- {$VMWARE.URL} - URL SDK сервиса VMware (vCenter или гипервизора ESXi) (https://servername/sdk)
- {$VMWARE.USERNAME} - имя пользователя сервиса VMware
- {$VMWARE.PASSWORD} - пароль пользователя {$VMWARE.USERNAME} сервиса VMware

10 Оповещения на события

Обзор

Предположим, мы уже настроили несколько элементов данных и триггеров и теперь мы получаем некоторые события, которые появляются в результате 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 оповещен при наступлении определенного 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 удовлетворяют, будут произведены операции. Имеется две основных операции это отправка сообщений (оповещений) и выполнение удаленных команд.

Для созданных событий обнаружениями и автоматической регистрацией, доступны несколько дополнительных операций. Они включают в себя добавление и удаление узлов сети, присоединение шаблонов и др.

1 Способы оповещений

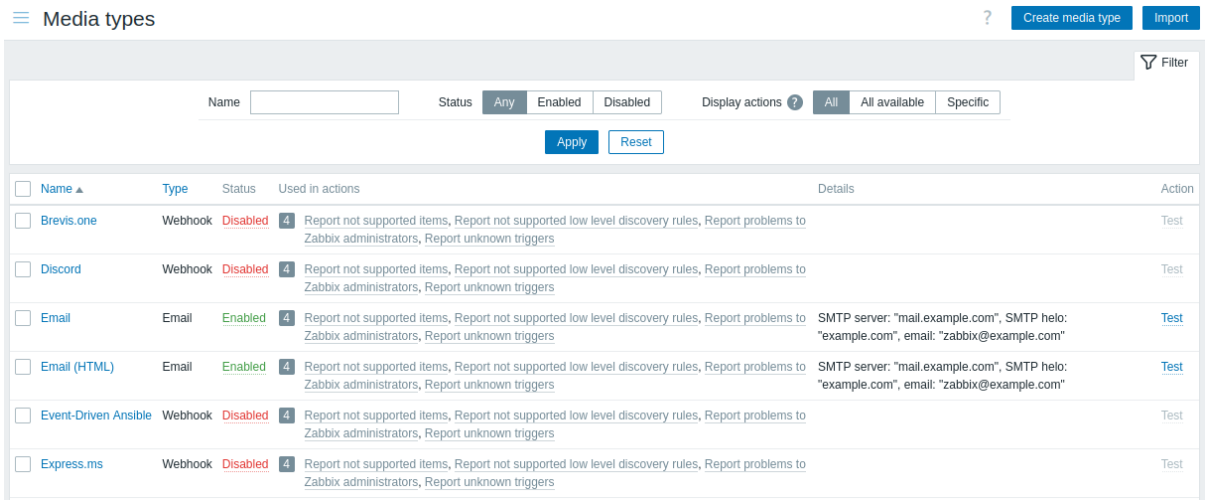
Обзор

Способы оповещения — это каналы доставки, используемые для отправки уведомлений и предупреждений из Zabbix.

Способы оповещения поддерживают следующие методы доставки:

- Email
- SMS
- Пользовательский скрипт
- Вебхук

Способы оповещения настраиваются в *Оповещения* > *Способы оповещения*. Некоторые способы оповещения уже предопределены в наборе данных по умолчанию.
Вам нужно лишь скорректировать их параметры, чтобы они заработали.



Чтобы понять, как способы оповещения вписываются в процесс отправки предупреждений, рассмотрим три требования для доставки уведомлений из Zabbix конечным пользователям:

1. Должна быть определена операция действия, которая отправляет уведомления
2. Должен быть определён рабочий способ оповещения (например, *Email*, который отправляет предупреждения с использованием SMTP)
3. На уровне пользователя должны быть определены данные для доставки (например, адреса электронной почты, номера телефонов и т. д.) в средствах оповещения пользователя

Тестирование способа оповещения

Чтобы проверить, работает ли настроенный способ оповещения, нажмите *Тест* в списке способов оповещения.

Запрос на тестирование будет отправлен на сервер Zabbix. Сервер Zabbix попытается отправить оповещение, используя указанный способ оповещения, и вернет результат в веб-интерфейс. Веб-интерфейс будет ожидать, пока сервер не вернет результаты. По умолчанию для тестирования способа оповещения установлен тайм-аут 65 секунд (настраивается в *Администрирование* > *Общие* > *Тайм-ауты*).

См. также подробности тестирования для:

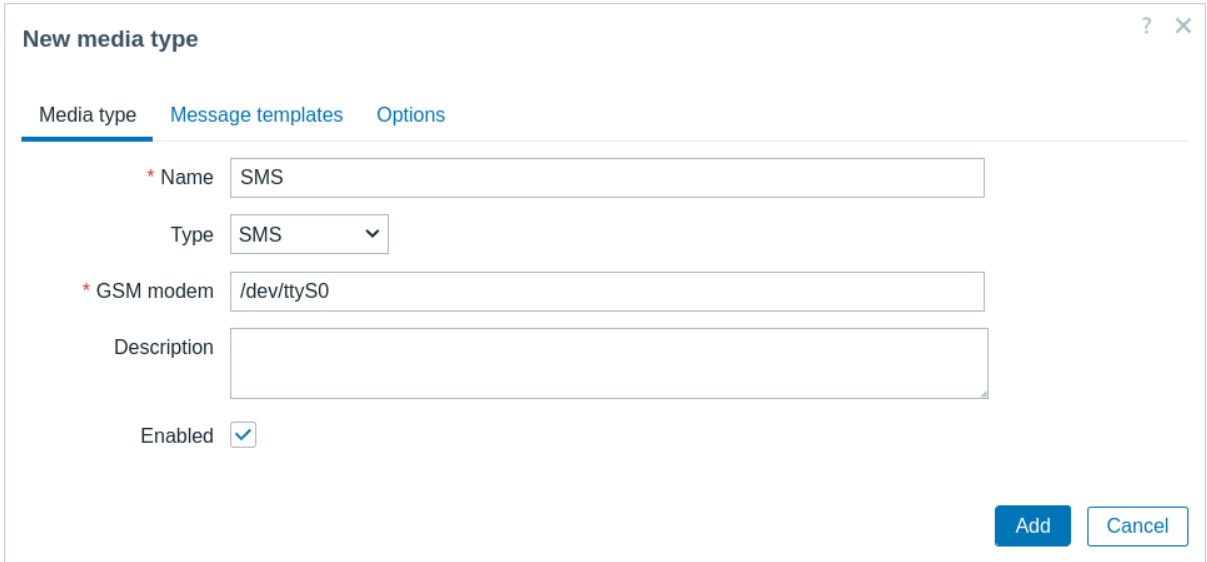
- Email
- Webhook
- Script

Настройка

Чтобы создать способ оповещения в веб-интерфейсе Zabbix:

- Перейдите в: *Оповещения* > *Способы оповещения*
- Нажмите *Создать способ оповещения*
- Введите параметры способа оповещения в форме

Некоторые параметры являются **общими** для всех методов доставки.

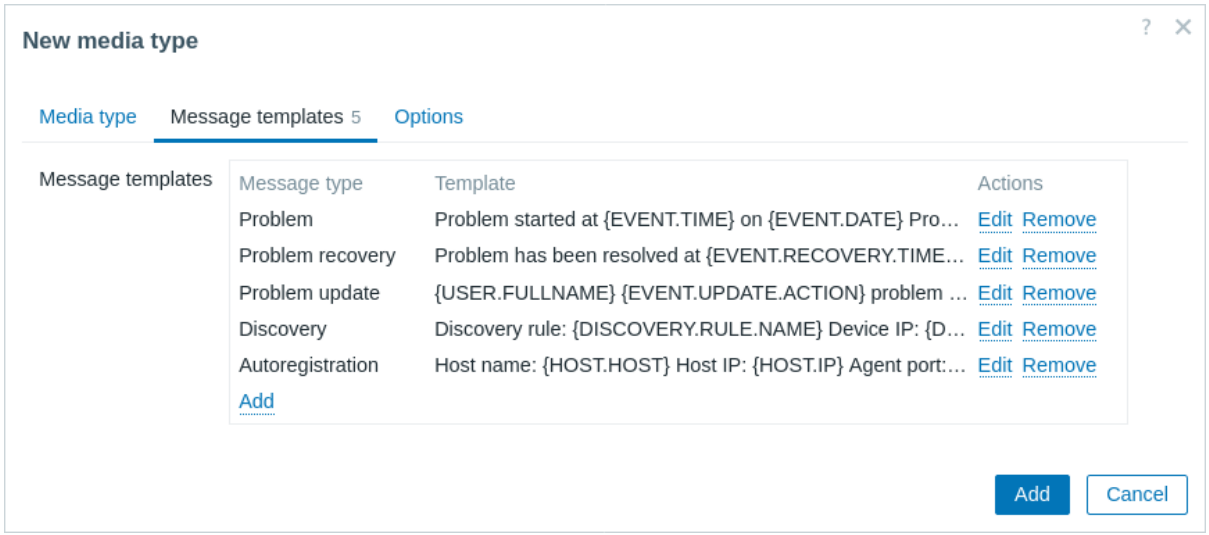


Параметр	Описание
Name	Имя способа оповещения.
Type	Выберите метод доставки для способа оповещения.
Description	Введите описание способа оповещения.
Enabled	Установите флажок, чтобы включить способ оповещения.

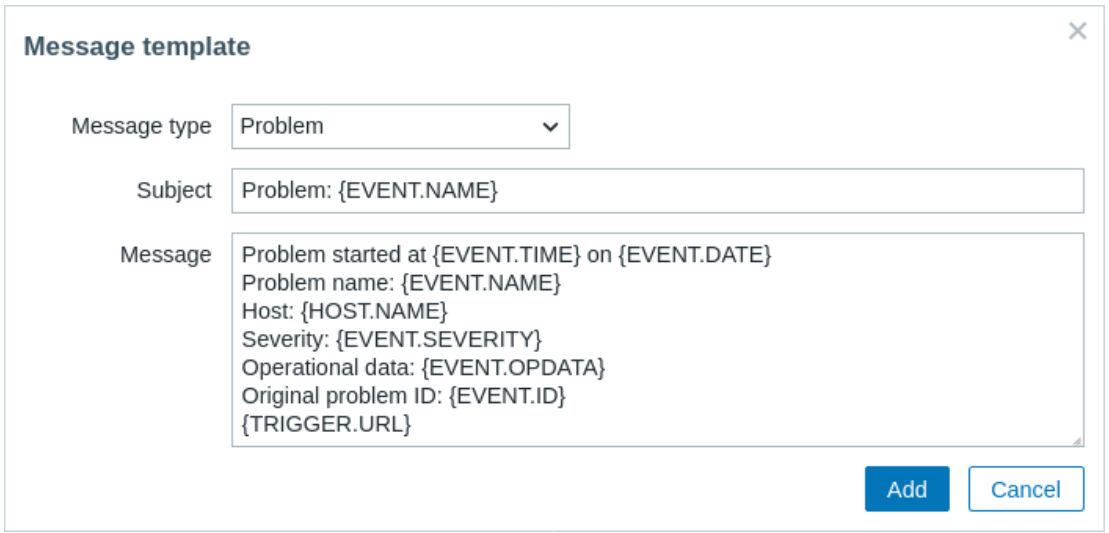
Параметры, специфичные для отдельных методов, см. на страницах email, SMS, пользовательский alertscript или вебхук.

Шаблоны сообщений

Вкладка **Шаблоны сообщений** содержит сообщения по умолчанию для каждого типа события (проблема, восстановление проблемы, обнаружение и т. д.).



Нажмите **Добавить**, чтобы определить сообщение по умолчанию (или **Изменить**, чтобы обновить существующее сообщение):



Параметр	Описание
Тип сообщения	Тип события, для которого должно использоваться сообщение по умолчанию. Для каждого типа события можно определить только одно сообщение по умолчанию.
Тема	Тема сообщения по умолчанию. Тема может содержать макросы. Она ограничена 255 символами. Тема недоступна для типа медиа SMS.
Сообщение	Сообщение по умолчанию. Оно ограничено определенным количеством символов в зависимости от типа базы данных (подробнее см. в разделе Отправка сообщений). Сообщение может содержать поддерживаемые макросы. В сообщениях о проблеме и обновлении проблемы поддерживаются макросы выражений (например, `{?avg(/host/key,1h)}`).

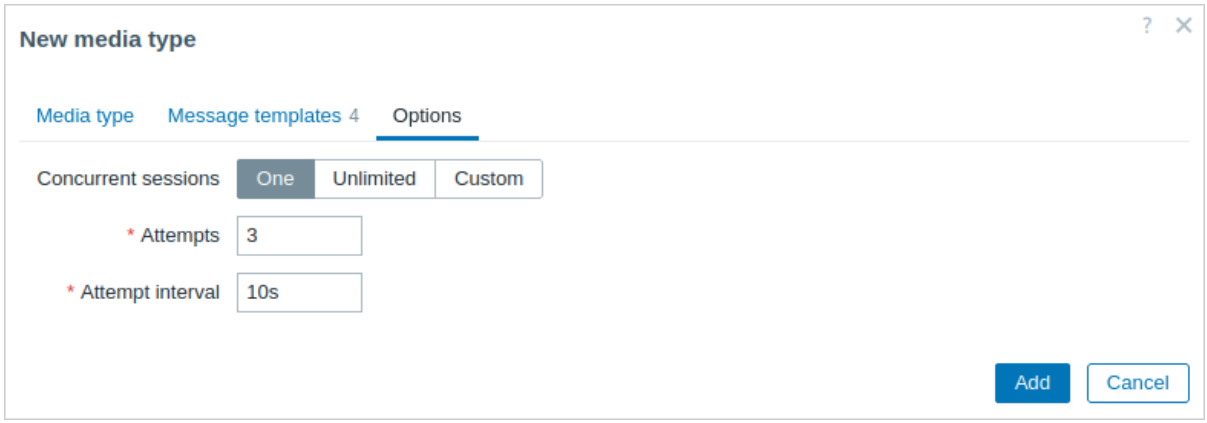
Обратите внимание, что сообщения по умолчанию переопределяются пользовательскими сообщениями, если они заданы в операциях действия.

Warning:
Определение шаблонов сообщений обязательно для всех методов доставки, включая вебхуки или пользовательские скрипты оповещений, которые не используют сообщения по умолчанию для уведомлений. Например, действие "Отправить сообщение в вебхук Pushover" не сможет отправить уведомления о проблемах, если сообщение о проблеме для вебхука Pushover не определено.

Параметры

Вкладка **Параметры** содержит настройки обработки оповещений.
Один и тот же набор параметров можно настроить для каждого типа средства оповещения.

Все типы средств оповещения обрабатываются параллельно.
Хотя максимальное количество одновременных сеансов настраивается для каждого типа средства оповещения отдельно, общее количество процессов оповещателя на сервере может быть ограничено только параметром StartAlerters parameter.
Оповещения, созданные одним триггером, обрабатываются последовательно.
Таким образом, несколько уведомлений могут обрабатываться одновременно только в том случае, если они созданы несколькими триггерами.



Параметр	Описание
Параллельные сеансы	Выберите количество параллельных сеансов оповещателя для типа средства оповещения: **Один** — один сеанс **Неограниченно** — неограниченное количество сеансов **Пользовательское** — выберите пользовательское количество сеансов Неограниченные/высокие значения означают большее количество параллельных сеансов и увеличение пропускной способности отправки уведомлений. Неограниченные/высокие значения следует использовать в крупных средах, где может потребоваться одновременная отправка большого количества уведомлений. Если требуется отправить больше уведомлений, чем доступно параллельных сеансов, оставшиеся уведомления будут поставлены 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 выполнять повторные попытки в случае сбоя отправки.
Интервал попыток	Частота повторных попыток отправки уведомления в случае сбоя, в секундах (0-3600). Если указано '0', Zabbix выполнит повторную попытку немедленно. Поддерживаются суффиксы времени, например 5s, 3m, 1h.

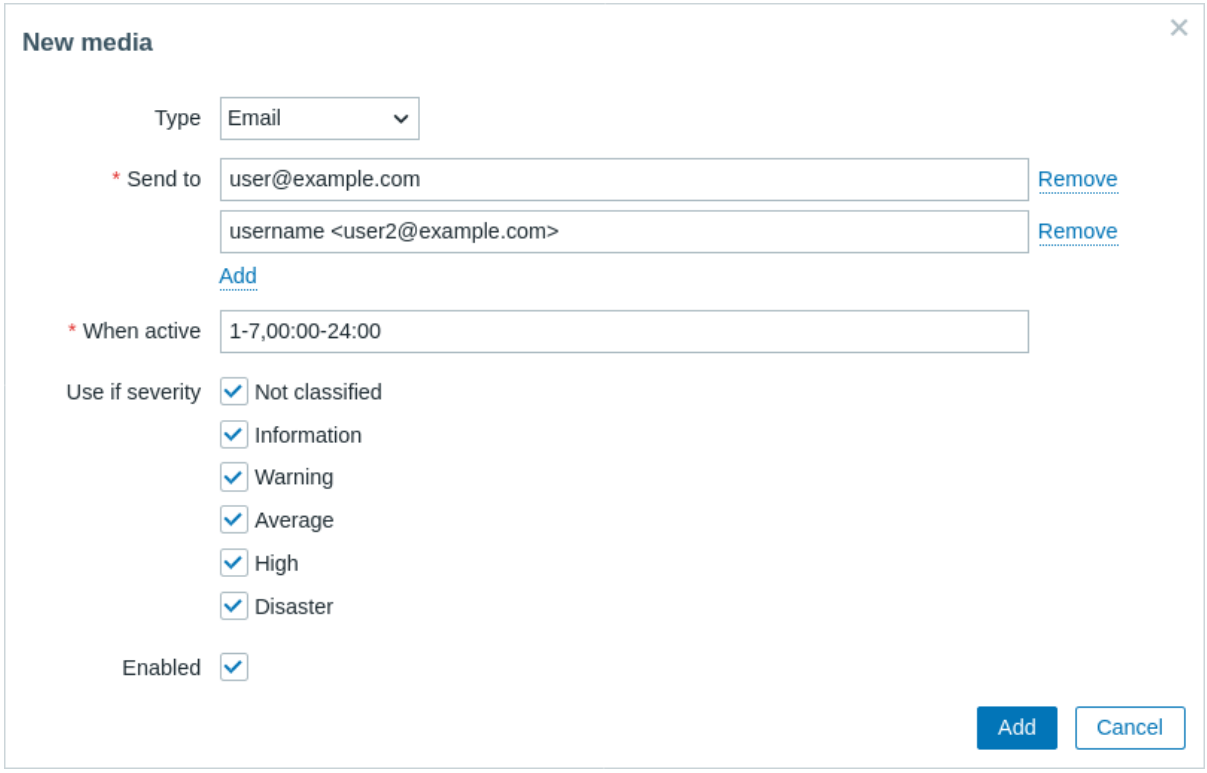
Способы оповещения пользователя

Хотя типы средств оповещения определяют, **как** будет отправлено уведомление, способы оповещения пользователя определяют, **куда** это уведомление должно быть отправлено.

Способы оповещения пользователя (например, адрес электронной почты, ID пользователя вебхука и т. д.) должны быть определены в профиле пользователя независимо от метода доставки. Действие, отправляющее сообщения пользователю *Admin* с использованием вебхука *X*, не сможет доставить их, если данные доставки для вебхука *X* не определены в профиле пользователя Admin.

Чтобы определить способы оповещения пользователя:

- Перейдите в *Users* > *Users* и откройте форму свойств пользователя (или перейдите в *User settings* > *Notifications* в профиле вашего пользователя)
- Нажмите *Add* на вкладке Media



Параметр	Описание
Type	В раскрывающемся списке содержатся названия включённых типов средств оповещения. Обратите внимание, что при редактировании способа оповещения отключённого типа средств оповещения этот тип будет отображаться красным цветом.
Send to	Введите контактную информацию, на которую должны отправляться сообщения. Для типа средства оповещения email можно добавить несколько адресов, нажав кнопку Add под полем адреса. В этом случае уведомления будут отправляться на все указанные адреса. Примеры адресов см. в описании параметра *Email* для типа средства оповещения email.
When active	Вы можете ограничить время отправки сообщений, например, указать только рабочие дни (1-5,09:00-18:00). Обратите внимание, что это ограничение основано на часовом поясе пользователя. Если часовой пояс пользователя изменён и отличается от часового пояса системы, это ограничение может потребовать соответствующей корректировки, чтобы не пропустить важные сообщения. Описание формата см. на странице Спецификация периода времени. Поддерживаются пользовательские макросы.
Use if severity	Отметьте флажками важности триггеров, для которых вы хотите получать уведомления. *Обратите внимание*, что важность по умолчанию ('Not classified') **должна быть** отмечена, если вы хотите получать уведомления о событиях, не связанных с триггерами. После сохранения выбранные важности триггеров будут отображаться соответствующими цветами важности, а невыбранные будут показаны серым цветом.
Status	Состояние способа оповещения пользователя. **Enabled** — используется. **Disabled** — не используется.

1 E-mail

Обзор

Чтобы настроить электронную почту в качестве канала доставки сообщений, необходимо настроить электронную почту как тип оповещения и назначить пользователям соответствующие адреса.

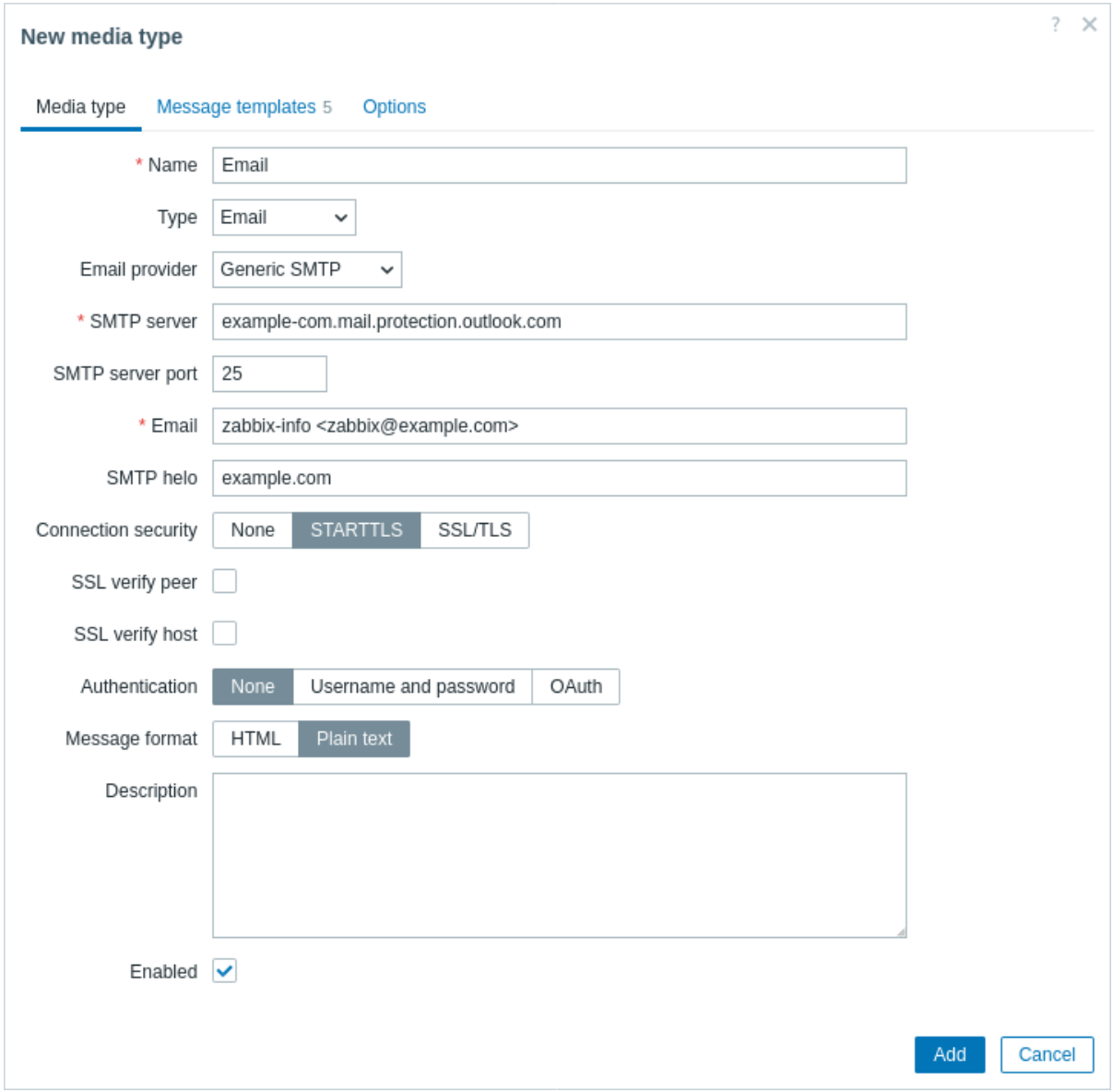
Note:
Несколько уведомлений для одного события будут сгруппированы в одной цепочке писем.

Конфигурация

Чтобы настроить электронную почту как тип средства оповещения:

1. Перейдите в *Оповещения* > *Типы средств оповещения*.
2. Нажмите *Создать тип средства оповещения* (или нажмите *Email* в списке предопределённых типов средств оповещения).

Вкладка **Тип средства оповещения** содержит общие атрибуты типа средства оповещения:



Все обязательные поля ввода отмечены красной звёздочкой.

Warning:
Пароль экспортируется в открытом виде при экспорте конфигурации типа средства оповещения электронной почты.

Следующие параметры являются специфичными для типа средства оповещения электронной почты:

Параметр	Описание
Поставщик электронной почты	Выберите поставщика электронной почты: *Generic SMTP*, *Gmail*, *Gmail relay*, *Office365* или *Office365 relay*. Если вы выберете параметры, связанные с Gmail/Office365, вам потребуется указать только адрес электронной почты отправителя и пароль; такие параметры, как *SMTP server*, *SMTP server port*, *SMTP helo* и *Connection security*, будут автоматически заполнены Zabbix. См. также: Автоматизированные типы средств оповещения Gmail/Office365.
SMTP server	Укажите SMTP-сервер для обработки исходящих сообщений. Это поле доступно, если в качестве поставщика электронной почты выбран *Generic SMTP*.
SMTP server port	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 как адрес **From** для отправляемых сообщений. Поддерживается добавление отображаемого имени отправителя (например, ”Zabbix_info” в *Zabbix_info <zabbix@company.com>* на снимке экрана выше) вместе с фактическим адресом электронной почты. Для отображаемых имён в электронных письмах Zabbix существуют некоторые ограничения по сравнению с тем, что допускается RFC 5322, как показано в примерах: Допустимые примеры: *zabbix@company.com* (только адрес электронной почты, угловые скобки не требуются) *Zabbix_info <zabbix@company.com>* (отображаемое имя и 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 заключён в угловые скобки) *”Zabbix\@\<H(comment)Q\>” <zabbix@company.com>* (хотя это допустимо согласно RFC 5322, экранированные пары и комментарии не поддерживаются в электронных письмах Zabbix)
SMTP helo	Укажите корректное значение SMTP helo, обычно это доменное имя. Если поле пустое, будет отправлено доменное имя из адреса электронной почты (то есть часть после @ в поле *Email*). Если получить доменное имя невозможно, в журнал будет записано предупреждение уровня отладки, и в качестве домена для команды HELO будет отправлено имя узла сервера. Это поле доступно, если в качестве поставщика электронной почты выбран *Generic SMTP*.
Connection security	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 является необязательным
SSL verify peer	Установите этот флажок, чтобы проверять SSL-сертификат SMTP-сервера. Значение директивы конфигурации сервера ”SSLCALocation” должно быть передано в CURLOPT_CAPATH для проверки сертификата. Это задаёт параметр cURL CURLOPT_SSL_VERIFYPEER.
SSL verify host	Установите этот флажок, чтобы проверять соответствие поля *Common Name* или поля *Subject Alternate Name* сертификата SMTP-сервера. Это задаёт параметр cURL CURLOPT_SSL_VERIFYHOST.
Authentication	Выберите уровень аутентификации: **None** — параметры cURL не задаются **Username and password** — подразумевает ”AUTH=*”, оставляя выбор механизма аутентификации cURL **OAuth** — аутентификация OAuth Аутентификация OAuth не поддерживается для поставщика электронной почты *Office365 relay*.
Username	Имя пользователя для аутентификации. Это задаёт значение CURLOPT_USERNAME. Поддерживаются пользовательские макросы.

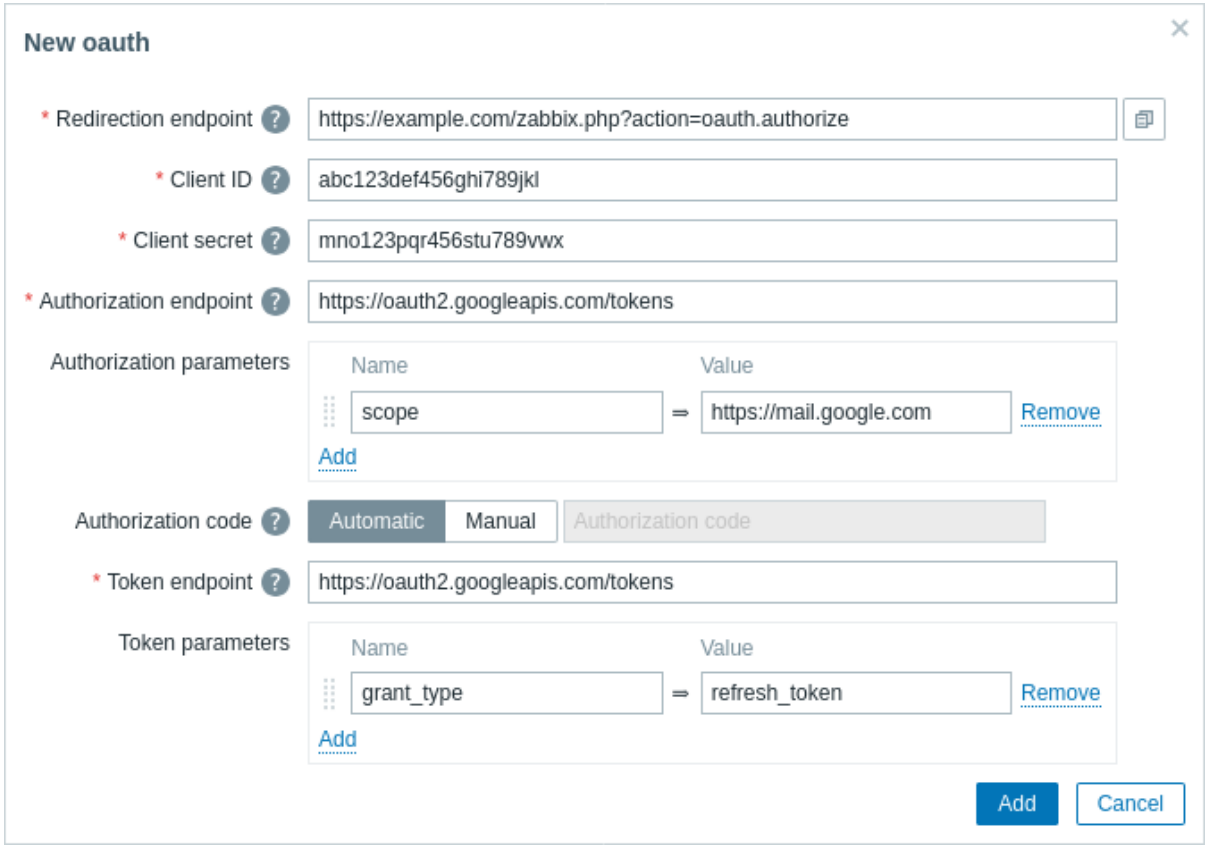
Параметр	Описание
Password	Пароль для аутентификации. Это задаёт значение CURLOPT_PASSWORD. Поддерживаются пользовательские макросы. Введённое здесь значение экспортируется как есть при экспорте конфигурации типа средства оповещения. Чтобы избежать раскрытия конфиденциальных учётных данных в экспортируемых файлах, используйте пользовательский макрос (например, `{$EMAIL_SMTP_PASSWORD}`) вместо буквального пароля — при экспорте будет сохранена ссылка на макрос, а секретное значение должно быть настроено (повторно) в целевой системе.
OAuth tokens	Нажмите *Configure*, чтобы настроить параметры для получения OAuth tokens в новом окне. Это поле доступно только если в поле *Authentication* выбрано "OAuth".
Message format	Выберите формат сообщения: **HTML** — отправлять как HTML **Plain text** — отправлять как обычный текст

Attention:
Чтобы включить SMTP-аутентификацию, сервер Zabbix должен быть скомпилирован с параметром компиляции `--with-libcurl`, для которого требуется libcurl (подробности о версии см. в требованиях к серверу).

См. также общие параметры типов средств оповещения для получения подробной информации о настройке сообщений по умолчанию и параметров обработки оповещений.

Токены OAuth

Для получения токенов OAuth требуются следующие параметры:



Параметр	Описание
Конечная точка перенаправления	Введите URL веб-интерфейса Zabbix, на который сервис OAuth будет перенаправлять обратно после авторизации OAuth (используя формат `https://<zabbix-frontend-url>/zabbix.php?action=oauth.authorize`). Для нового типа медиа email задается автоматически, если определен URL веб-интерфейса.

Параметр	Описание
Client ID	Введите уникальный идентификатор клиентского приложения, зарегистрированного на сервере авторизации OAuth.
Client secret	Введите приватный секрет клиентского приложения, зарегистрированного на сервере авторизации OAuth.
Конечная точка авторизации	Введите URL сервера авторизации OAuth для запроса авторизации пользователя.
Параметры авторизации	Введите параметры для конечной точки авторизации.
Код авторизации	Введите код авторизации: **Автоматически** - код будет получен автоматически через страницу перенаправления **Вручную** - введите код вручную, если автоматическое получение не удалось
Конечная точка токена	Введите URL сервера авторизации OAuth для обмена кода авторизации на токены доступа и обновления.
Параметры токена	Введите параметры для токена доступа.

Note:
Получение токенов OAuth имеет автоматизированные возможности для провайдеров электронной почты **Gmail**, **Gmail relay** и **Office365**. Требуется указать только значения параметров *Конечная точка перенаправления*, *Client ID* и *Client secret*. Zabbix автоматически заполнит остальные необходимые значения (см. Значения URL OAuth по умолчанию в зависимости от провайдера).

Получение токенов доступа и обновления OAuth

После отправки параметров OAuth:

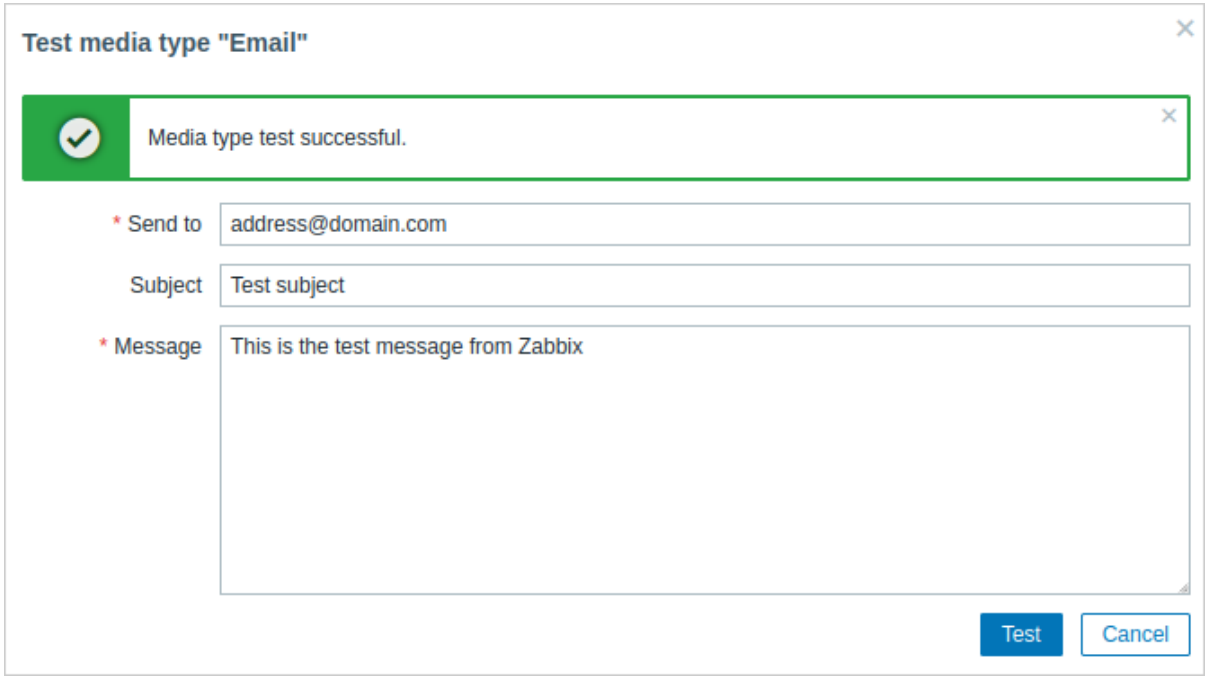
1. Открывается всплывающее окно браузера, перенаправляющее пользователя к *Authorization endpoint*.
2. Пользователь авторизует Zabbix в сервисе OAuth.
3. Сервис OAuth перенаправляет пользователя к действию веб-интерфейса Zabbix `oauth.authorize` с кодом авторизации и значением области действия.
4. В ответ Zabbix выполнит запрос к *Token endpoint*, чтобы обменять код авторизации на токены доступа и обновления.

Тестирование

Чтобы проверить, правильно ли работает настроенный тип оповещения по электронной почте:

1. Найдите соответствующее письмо в списке типов оповещений.
2. Нажмите *Тест* в последнем столбце списка (откроется окно тестирования).
3. Введите адрес получателя в поле *Отправить кому*, текст сообщения и, при необходимости, тему.
4. Нажмите *Тест*, чтобы отправить тестовое сообщение.

Сообщение об успешном или неуспешном результате теста будет отображено в том же окне:



Способы оповещения пользователя

После настройки способа оповещения Email перейдите в раздел *Пользователи* > *Пользователи* и отредактируйте профиль пользователя, чтобы назначить пользователю способ оповещения Email. Шаги по настройке способа оповещения пользователя, общие для всех типов способов оповещения, описаны на странице Типы способов оповещения.

1 Автоматизированные типы медиа для Gmail/Office365

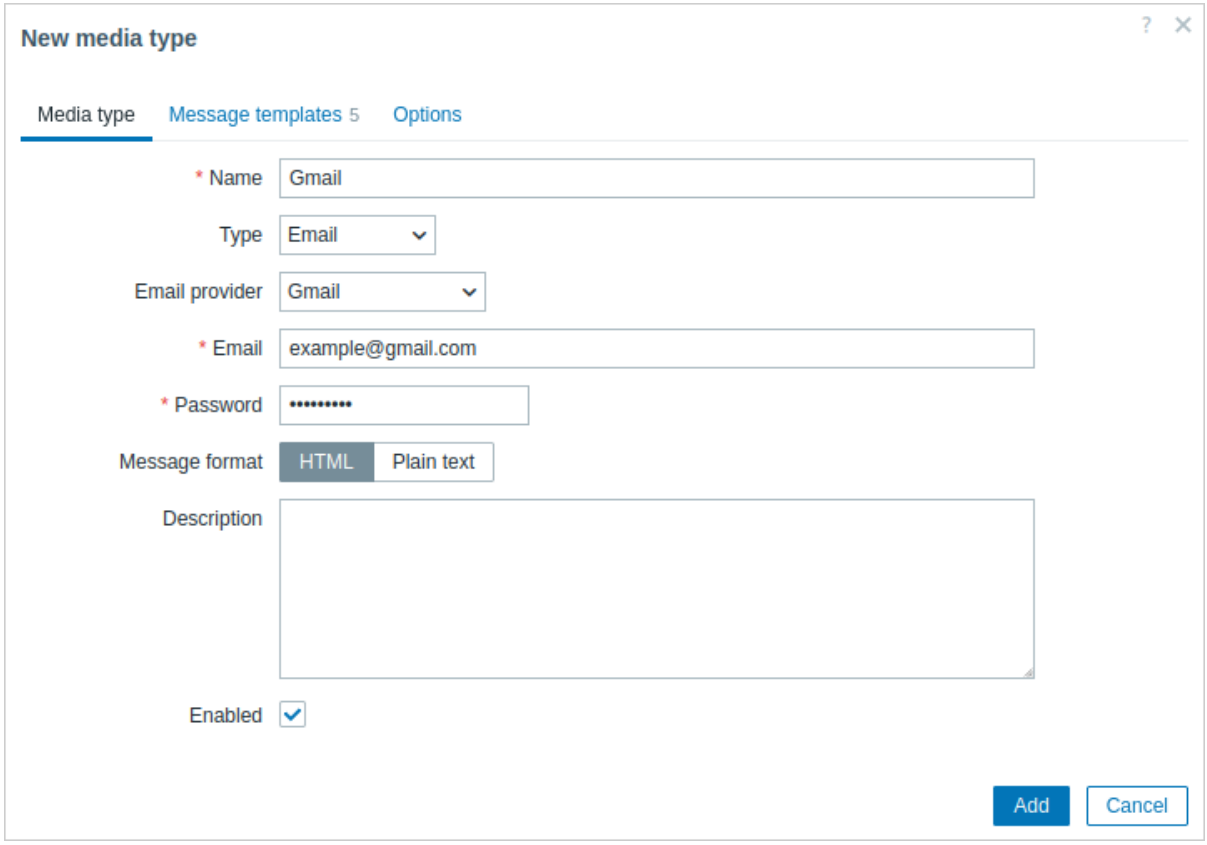
Обзор

Пользователи Gmail или Office365 могут воспользоваться автоматизированными возможностями в настройке типа оповещения.

Конфигурация

Поле *Поставщик электронной почты* в конфигурации типа медиа позволяет выбрать предварительно настроенные варианты для Gmail и Office 365.

При выборе вариантов, связанных с Gmail/Office365, для создания рабочего типа медиа требуется указать только адрес электронной почты отправителя/пароль.



Как только адрес электронной почты/пароль указаны, Zabbix сможет автоматически заполнить все необходимые настройки для типов медиа Gmail/Office365 актуальными/рекомендуемыми значениями, а именно: *SMTP server*, *SMTP server port*, *SMTP helo* и *Connection security*. Благодаря этой автоматизации эти поля даже не отображаются, однако можно увидеть SMTP-сервер и сведения об электронной почте в списке типов медиа (см. столбец *Details*).

Также обратите внимание:

- Для вариантов relay пароль не требуется.
- Для Office365 relay доменное имя указанного адреса электронной почты будет использоваться для динамического заполнения SMTP-сервера (то есть, замены "example.com" в `example-com.mail.protection.outlook.com` на реальное значение).

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.

2 SMS

Обзор

Zabbix поддерживает отправку SMS сообщений с использованием GSM модемов, подключенных к последовательному порту Zabbix сервера.

Убедитесь, что:

- Скорость последовательного устройства (обычно /dev/ttyS0 порт 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 в последовательное устройство. Выполните команду ls -l /dev/ttyS0, для просмотра текущих прав доступа к последовательному устройству.
- На GSM модеме введен ПИН-код и он сохраняется даже после перезагрузки. Помимо этого, вы можете отключить запрос ПИН-кода на самой SIM карте. ПИН-код можно ввести путем использования команды AT+CPIN="NNNN" (где, NNNN - это ваш ПИН-код, кавычки должны присутствовать) в терминальном приложении, в таких как Unix minicom или Windows HyperTerminal.

Zabbix был протестирован на следующих GSM модемах:

- Siemens MC35
- Teltonika ModemCOM/G10

Для настройки SMS каналом отправки сообщений, вам также необходимо настроить SMS способом оповещений и назначить соответствующие номера телефонов пользователям.

Настройка

Чтобы настроить SMS как тип медиа:

- Перейдите в *Оповещения → Способы оповещения*
- Нажмите *Создать способ оповещения* (или нажмите *SMS* в списке предопределённых способов оповещения).

Следующий параметр является специфичным для типа медиа SMS:

Параметр	Описание
GSM modem	Укажите имя последовательного устройства GSM-модема. Указанный здесь путь будет проверен по параметру сервера SMSDevices (если он задан).

Подробную информацию о том, как настроить сообщения по умолчанию и параметры обработки оповещений, смотрите в разделе общие параметры типа медиа. Обратите внимание, что параллельная обработка отправки SMS-уведомлений невозможна.

Способ оповещения пользователя

После настройки типа оповещения SMS перейдите в раздел *Пользователи → Пользователи* и отредактируйте профиль пользователя, чтобы назначить пользователю SMS как способ оповещения. Шаги по настройке способа оповещения пользователя, общие для всех типов оповещений, описаны на странице Типы оповещений.

3 Пользовательские скрипты оповещений

Обзор

Если вас не устраивают существующие способы оповещения для отправки предупреждений, есть альтернативный способ сделать это. Вы можете создать скрипт, который будет обрабатывать уведомления нужным вам образом.

Пользовательские скрипты оповещений выполняются на сервере Zabbix. Эти скрипты должны находиться в каталоге, указанном в параметре AlertScriptsPath файла конфигурации сервера.

Ниже приведён 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 выполняемых команд и скриптов. Любой код завершения, отличный от **0**, считается ошибкой выполнения команды. В таких случаях Zabbix попытается повторить неудавшееся выполнение.

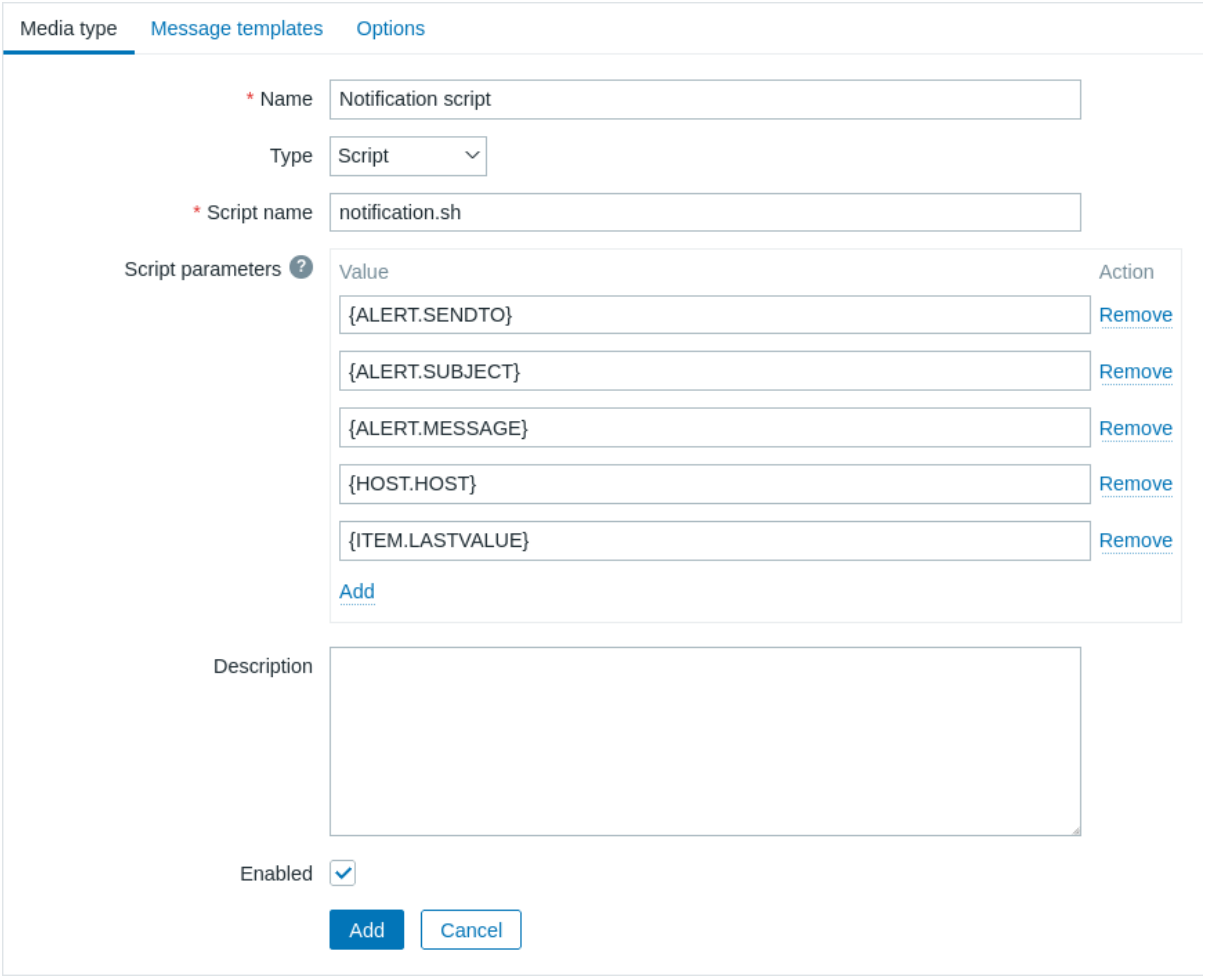
Переменные окружения не сохраняются и не создаются для скрипта, поэтому их следует обрабатывать явно.

Настройка

Чтобы настроить пользовательские скрипты оповещений как тип средства оповещения:

1. Перейдите в *Оповещения → Способы оповещений*.
2. Нажмите *Создать способ оповещения*.

Вкладка **Способ оповещения** содержит общие атрибуты типа средства оповещения:



Все обязательные поля ввода отмечены красной звёздочкой.

Следующие параметры относятся только к типу средства оповещения «скрипт»:

Parameter	Description
Script name	Введите имя файла скрипта (например, notification.sh), расположенного в каталоге, указанном в параметре конфигурации сервера AlertScriptsPath.
Script parameters	Добавьте необязательные параметры скрипта, которые будут переданы скрипту как аргументы командной строки в том порядке, в котором они определены. Параметры скрипта поддерживают макросы {ALERT.SENDTO}, {ALERT.SUBJECT}, {ALERT.MESSAGE}, а также все макросы, поддерживаемые в уведомлениях, и пользовательские макросы.

Подробную информацию о настройке сообщений по умолчанию и параметров обработки оповещений смотрите в разделе общие параметры типа средства оповещения.

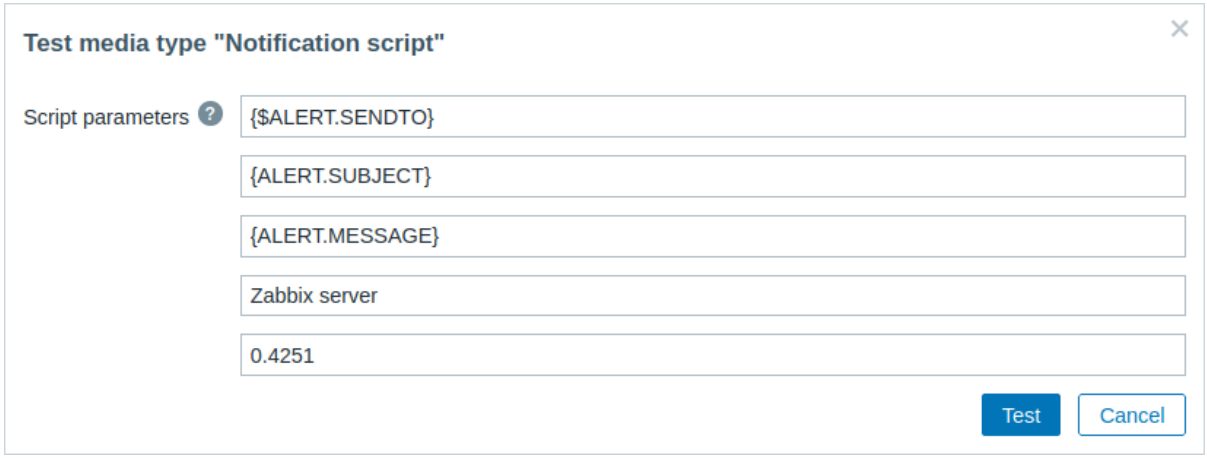
Warning:
Даже если скрипт оповещения не использует сообщения по умолчанию, шаблоны сообщений для типов операций, используемых этим типом средства оповещения, всё равно должны быть определены. В противном случае уведомление не будет отправлено.

Attention:
Если настроено более одного типа средства оповещения «скрипт», эти скрипты могут обрабатываться параллельно процессами оповещений. Общее количество процессов оповещений ограничено параметром `StartAlerters` в файле конфигурации сервера.

Тестирование

Чтобы протестировать настроенный скриптовый тип медиа:

1. Найдите соответствующий скрипт в списке типов медиа.
2. Нажмите *Test* в последнем столбце списка; форма тестирования откроется во всплывающем окне. Форма тестирования будет содержать такое же количество параметров, которое настроено для скриптового типа медиа.
3. При необходимости измените значения параметров скрипта. Редактирование влияет только на процедуру тестирования; фактические значения изменены не будут.
4. Нажмите *Test*.



Note:
При тестировании настроенного скриптового типа медиа {ALERT.SENDTO}, {ALERT.SUBJECT}, {ALERT.MESSAGE} и пользовательские макросы будут подставлены своими значениями, однако макросы, связанные с событиями (например, {HOST.HOST}, {ITEM.LASTVALUE} и т. д.), подставлены не будут, так как во время тестирования нет связанного события, из которого можно получить 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} подставлен не будет.

Способ оповещения пользователя

После настройки типа оповещения перейдите в раздел *Пользователи* → *Пользователи* и отредактируйте профиль пользователя, назначив этот тип оповещения пользователю. Шаги по настройке способа оповещения пользователя, общие для всех типов оповещений, описаны на странице Типы оповещений.

Обратите внимание, что при определении способа оповещения пользователя поле *Отправлять на* не может быть пустым. Если это поле не используется в скрипте оповещения, введите любую комбинацию поддерживаемых символов, чтобы обойти требования проверки.

4 Вебхук

Обзор

Тип медиа вебхук полезен для выполнения HTTP-вызовов с использованием пользовательского кода JavaScript для простой интеграции с внешним программным обеспечением, таким как системы helpdesk, чаты или мессенджеры. Вы можете импортировать интеграцию, предоставленную Zabbix, или создать собственную интеграцию с нуля.

Интеграции

Доступны следующие интеграции, позволяющие использовать предопределенные типы медиа с вебхуком 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
- 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 с **Spiceworks** (вебхук не требуется). Чтобы преобразовывать уведомления Zabbix в заявки Spiceworks, создайте тип медиа для электронной почты и укажите адрес электронной почты службы поддержки Spiceworks (например, help@zabbix.on.spiceworks.com) в настройках профиля назначенного пользователя Zabbix.

Настройка

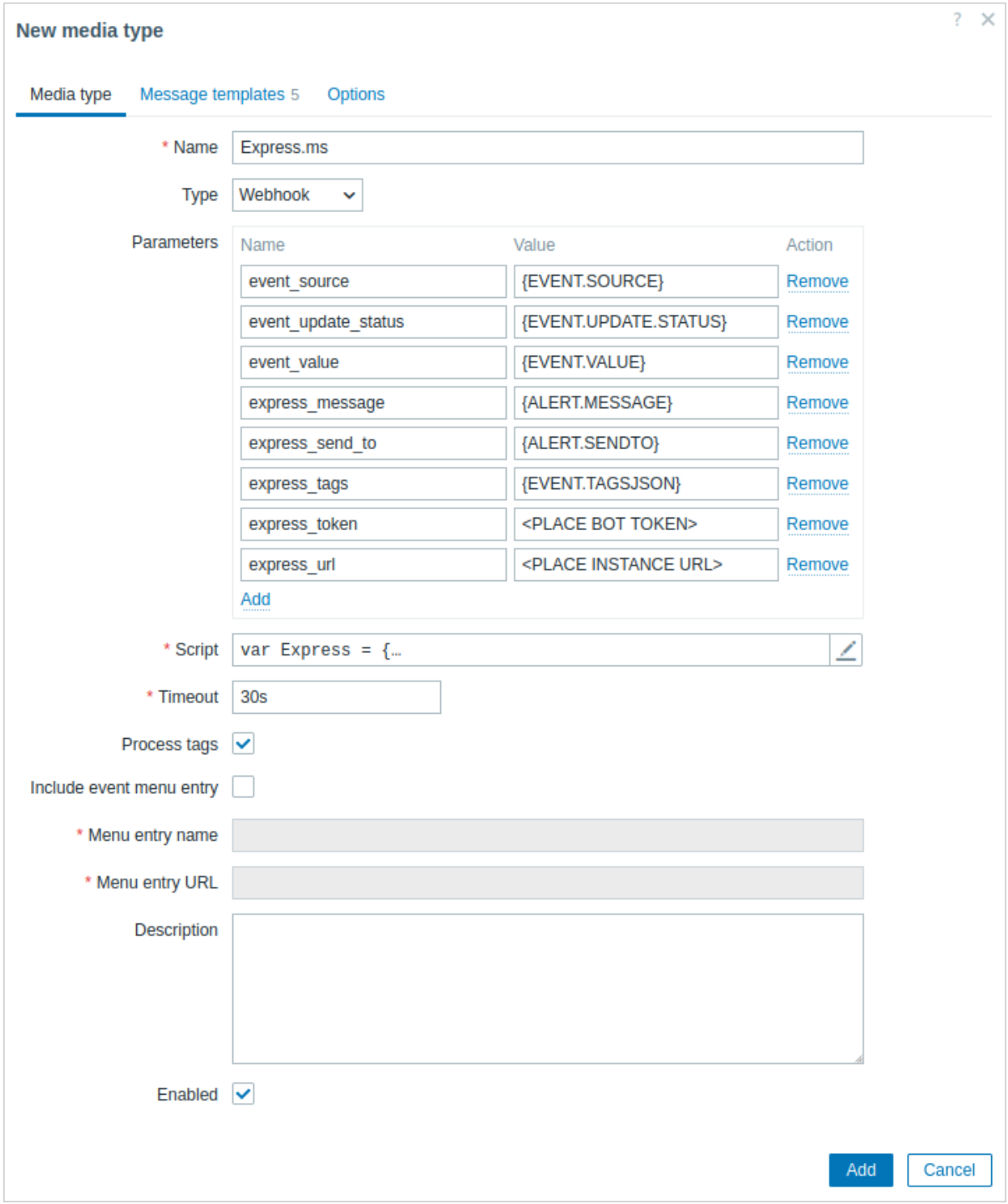
Чтобы начать использовать интеграцию с вебхуком:

1. Найдите требуемый файл .yaml в каталоге `templates/media` загруженной версии Zabbix или скачайте его из git-репозитория Zabbix.
2. Импортируйте файл в вашу установку Zabbix. Вебхук появится в списке способов оповещения.
3. Настройте вебхук в соответствии с инструкциями в файле *Readme.md* (для быстрого доступа к *Readme.md* можно нажать на имя вебхука выше).

Чтобы создать пользовательский вебхук с нуля:

1. Перейдите в *Оповещения* > *Способы оповещения*.
2. Нажмите *Создать способ оповещения*.

Вкладка **Способ оповещения** содержит различные атрибуты, специфичные для этого способа оповещения:



Все обязательные поля ввода отмечены красной звёздочкой.

Следующие параметры являются специфичными для способа оповещения вебхук:

Параметр	Описание
Параметры	Укажите переменные вебхука как пары атрибут-значение. Для предварительно настроенных вебхуков список параметров различается в зависимости от сервиса. Описание параметров смотрите в файле *Readme.md* соответствующего вебхука. Для новых вебхуков по умолчанию включены несколько общих переменных (URL:<empty>, HTTPProxy:<empty>, To:{ALERT.SENDTO}, Subject:{ALERT.SUBJECT}, Message:{ALERT.MESSAGE}); при необходимости их можно оставить или удалить. Параметры вебхука поддерживают пользовательские макросы, все макросы, поддерживаемые в уведомлениях о проблемах, а также дополнительно макросы {ALERT.SENDTO}, {ALERT.SUBJECT} и {ALERT.MESSAGE}. Если вы указываете HTTP-прокси, поле поддерживает ту же функциональность, что и поле HTTP proxy в настройке элемента данных. Строка прокси может иметь префикс `[scheme]://` для указания типа используемого прокси (например, https, socks4, socks5; см. документацию).
Скрипт	Введите код JavaScript в модальном редакторе, который открывается при нажатии на поле параметра или на значок карандаша рядом с ним. Этот код будет выполнять операцию вебхука. Скрипт представляет собой код функции, принимающей пары параметр-значение. Значения должны быть преобразованы в объекты JSON с помощью метода JSON.parse(), например: `var params = JSON.parse(value);`. Код имеет доступ ко всем параметрам, может выполнять HTTP-запросы GET, POST, PUT и DELETE, поддерживает дополнительные методы, такие как CONNECT, PATCH, HEAD, OPTIONS и TRACE, а также позволяет управлять HTTP-заголовками и телом запроса. Скрипт должен содержать оператор return, иначе он будет недействительным. Он может возвращать статус OK вместе с необязательным списком тегов и значений тегов (см. параметр *Обрабатывать теги*) либо строку ошибки. События восстановления (созданные автоматически или в результате ручного закрытия) создаются сервером и включают теги разрешённого события (включая теги, унаследованные от шаблонов, узлов сети и триггеров). Скрипты вебхуков выполняются после создания оповещения; поэтому теги, возвращаемые скриптом вебхука, добавляются только после первоначального создания оповещения и не будут присутствовать в макросах `{EVENT.TAGS}` и `{EVENT.RECOVERY.TAGS}` исходного 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 собственными данными и чтобы избежать конфликтов между одновременными вызовами (см. известные проблемы). См. также: Рекомендации по разработке вебхуков, Примеры скриптов вебхуков, Дополнительные объекты JavaScript.
Тайм-аут	Тайм-аут выполнения JavaScript (1-60s, по умолчанию 30s). Поддерживаются суффиксы времени, например 30s, 1m.
Обрабатывать теги	Установите флажок, чтобы обрабатывать значения возвращаемых свойств JSON как теги. Эти теги добавляются к любым существующим тегам проблемы. Обратите внимание, что при использовании тегов вебхука вебхук должен возвращать объект JSON, содержащий как минимум пустой объект tags: `var result = {tags: {}};` Примеры тегов, которые могут быть возвращены: *jira-id:prod-1234*, *responsible:John Smith*, *processed:<no value>*

Параметр	Описание
Включить запись в меню события	Установите флажок, чтобы добавить запись в меню события со ссылкой на созданный внешний тикет. Запись будет добавлена для каждого вебхука, который включён и для которого установлен этот флажок. Обратите внимание, что если параметры *Имя записи меню* и *URL записи меню* содержат какие-либо макросы {EVENT.TAGS.<tag name>}, запись будет добавлена только в том случае, если эти макросы могут быть разрешены (то есть если для события определены эти теги). Если этот параметр отмечен, вебхук не следует использовать для отправки уведомлений разным пользователям (вместо этого рассмотрите возможность создания выделенного пользователя); также его не следует использовать в нескольких действиях оповещений для одного события проблемы.
Имя записи меню	Укажите имя записи меню. Поддерживается макрос {EVENT.TAGS.<tag name>}. Это поле является обязательным только в том случае, если отмечен параметр *Включить запись в меню события*.
URL записи меню	Укажите URL записи меню. Поддерживается макрос {EVENT.TAGS.<tag name>}. Это поле является обязательным только в том случае, если отмечен параметр *Включить запись в меню события*.

Подробные сведения о настройке сообщений по умолчанию и параметров обработки оповещений см. в разделе общие параметры способа оповещения.

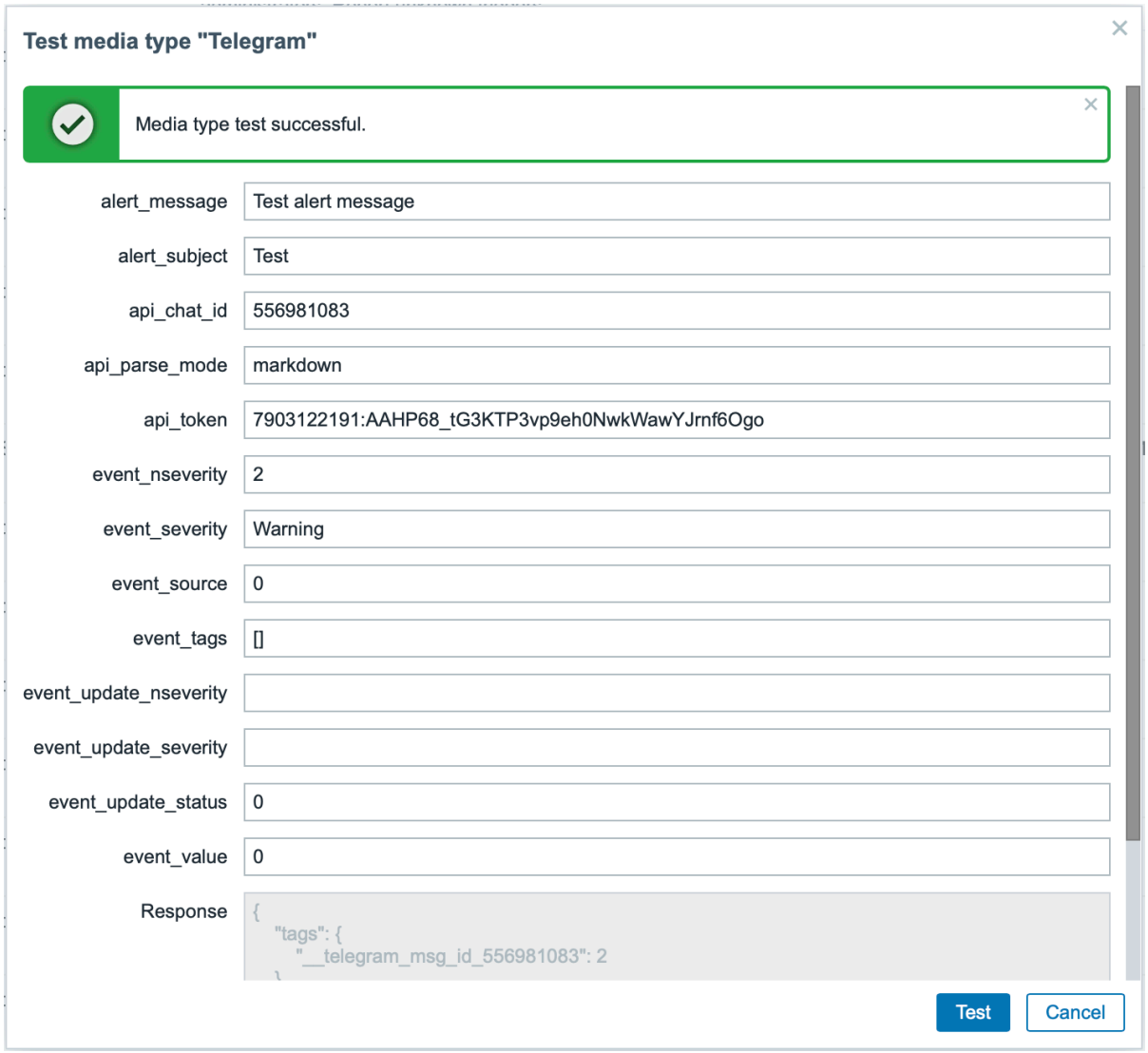
Warning:
Даже если вебхук не использует сообщения по умолчанию, шаблоны сообщений для типов операций, используемых этим вебхуком, всё равно должны быть определены.

Тестирование

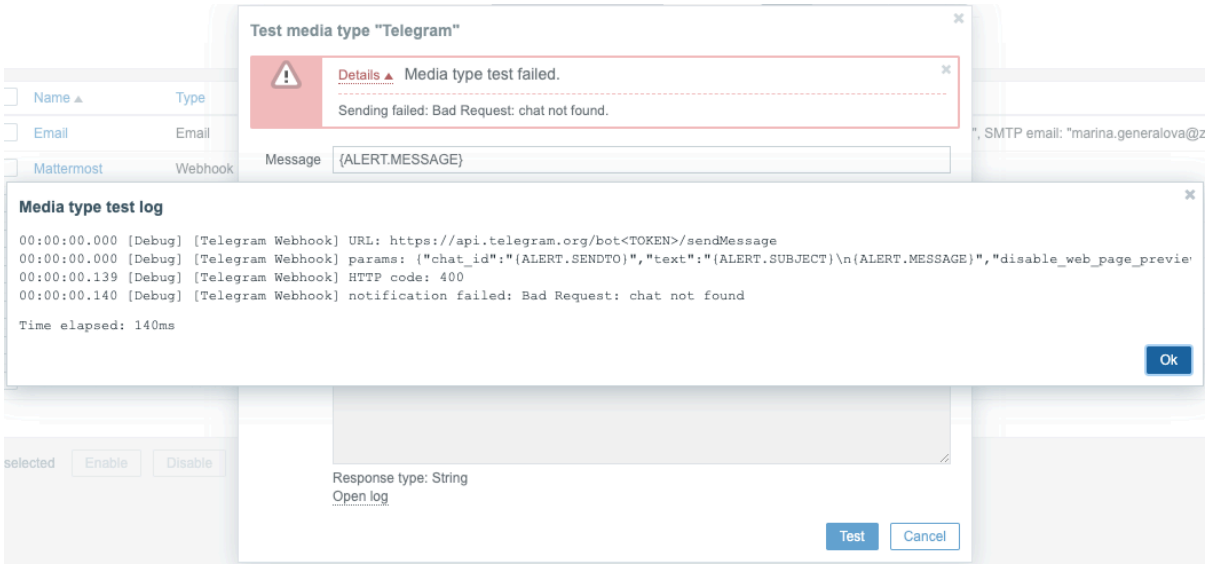
Чтобы протестировать настроенный медиа-тип вебхука:

1. Найдите соответствующий вебхук в списке медиа-типов.
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 разрешены, 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 медиа-типа, не покидая окно тестирования, нажмите *Открыть журнал* (откроется новое всплывающее окно).



Если тест вебхука выполнен успешно:

- Отображается сообщение *"Media type test successful."*.
- Ответ сервера отображается в сером поле *Response*.
- Тип ответа (JSON или String) указывается под полем *Response*.

Если тест вебхука завершился неудачей:

- Отображается сообщение *"Media type test failed."*, за которым следуют дополнительные сведения о сбое.

Пользовательские медиа

После настройки типа медиа перейдите в раздел *Users* > *Users* и назначьте медиа вебхука существующему пользователю или создайте нового пользователя, который будет представлять вебхук. Шаги по настройке пользовательских медиа для существующего пользователя, общие для всех типов медиа, описаны на странице Типы медиа.

Если вебхук использует теги для хранения ID заявки\сообщения, не назначайте один и тот же вебхук в качестве медиа разным пользователям, так как это может вызвать ошибки вебхука (это относится к большинству вебхуков, использующих опцию *Include event menu entry*). В этом случае рекомендуется создать отдельного пользователя, который будет представлять вебхук:

1. После настройки типа медиа вебхука перейдите в раздел *Users* > *Users* и создайте отдельного пользователя Zabbix, который будет представлять вебхук — например, с именем пользователя *Slack* для вебхука Slack. Все настройки, кроме медиа, можно оставить по умолчанию, так как этот пользователь не будет входить в Zabbix.
2. В профиле пользователя перейдите на вкладку *Media* и добавьте вебхук, указав необходимую контактную информацию. Если вебхук не использует поле *Send to*, введите любую комбинацию поддерживаемых символов, чтобы обойти требования проверки.
3. Предоставьте этому пользователю как минимум права на чтение для всех узлов сети, для которых он должен отправлять оповещения, permissions.

При настройке действия оповещения добавьте этого пользователя в поле *Send to users* в сведениях об операции — это укажет Zabbix использовать вебхук для уведомлений из этого действия.

Настройка действий оповещений

Действия определяют, какие уведомления должны отправляться через вебхук. Шаги по настройке действий с использованием вебхуков такие же, как и для всех остальных типов средств оповещения, за исключением следующих случаев:

- Если вебхук использует теги вебхука для хранения ID заявки\сообщения и обработки операций обновления\решения, избегайте использования одного и того же вебхука в нескольких действиях оповещений для одного события проблемы. Если {EVENT.TAGS.<tag name>} существует и обновляется в вебхуке, его итоговое значение будет неопределённым. Чтобы избежать этого, используйте в вебхуке новое имя тега для хранения обновлённых 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 обратите внимание, что один вебхук можно использовать в нескольких операциях или шагах эскалации одного и того же действия, а также в разных действиях, которые не будут запускаться одним и тем же событием проблемы благодаря различным условиям.
- 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 не будет отправлено.

1 Примеры скриптов вебхуков

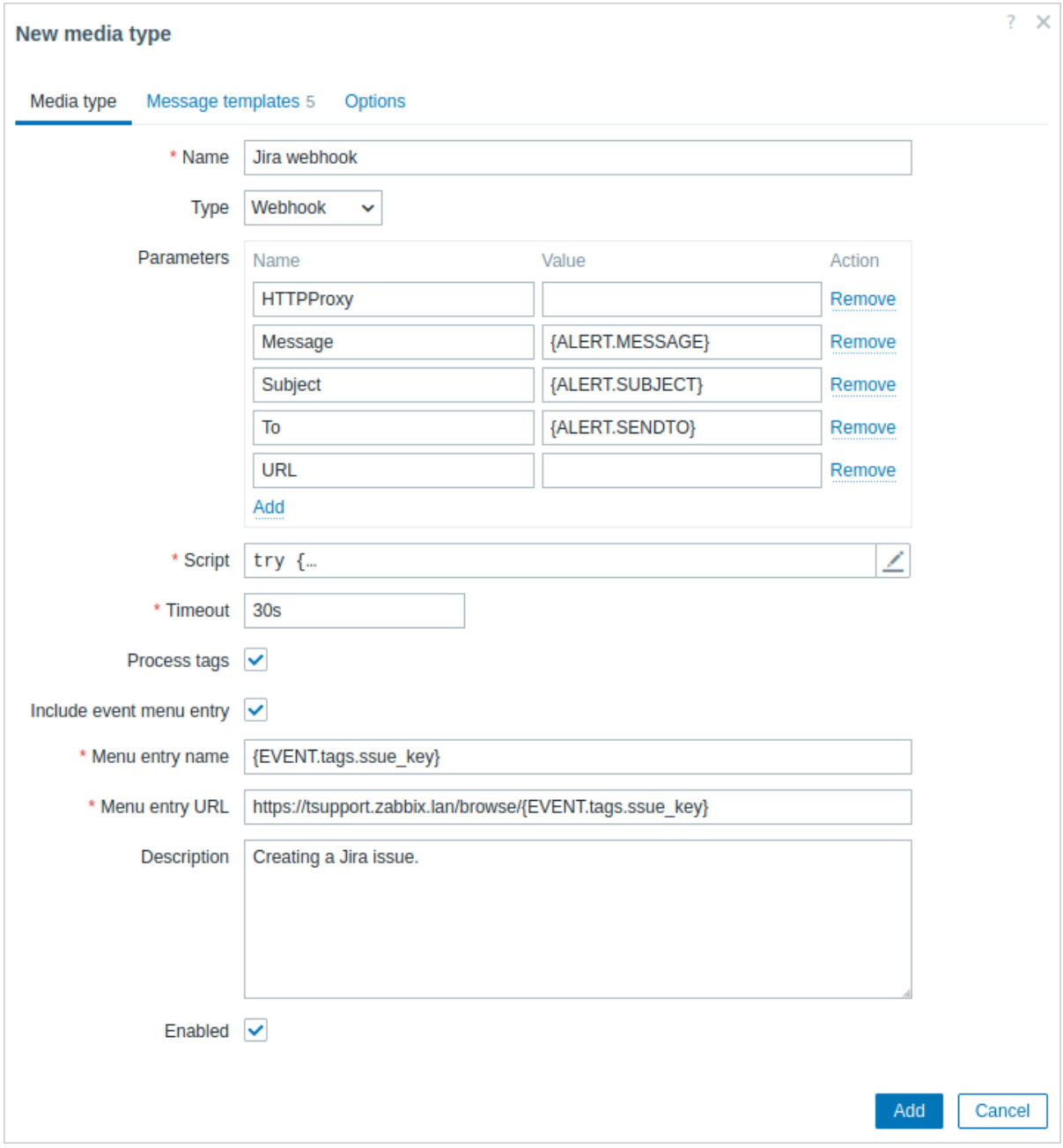
Обзор

Хотя Zabbix предлагает большое количество интеграций с вебхуками, доступных «из коробки», вместо этого вы можете захотеть создать собственные вебхуки. В этом разделе приведены примеры пользовательских скриптов вебхуков (используются в параметре *Script*). Описание других параметров вебхука см. в разделе webhook.

Attention:
Не используйте необъявленные присваивания в JavaScript предобработки. Используйте `var` для объявления локальных переменных.

События восстановления (созданные автоматически или после manual close) включают теги устранённого события (включая теги, унаследованные от шаблонов, узлов сети и триггеров). Скрипты вебхуков выполняются после создания оповещения; поэтому теги, возвращаемые скриптом вебхука, применяются после первоначального создания оповещения и могут отсутствовать в оповещении, которое запустило вебхук. Если 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 корреляцию там.

Вебхук Jira (пользовательский)



Этот скрипт создаст задачу JIRA и вернет некоторую информацию о созданной задаче.

```
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();
    }

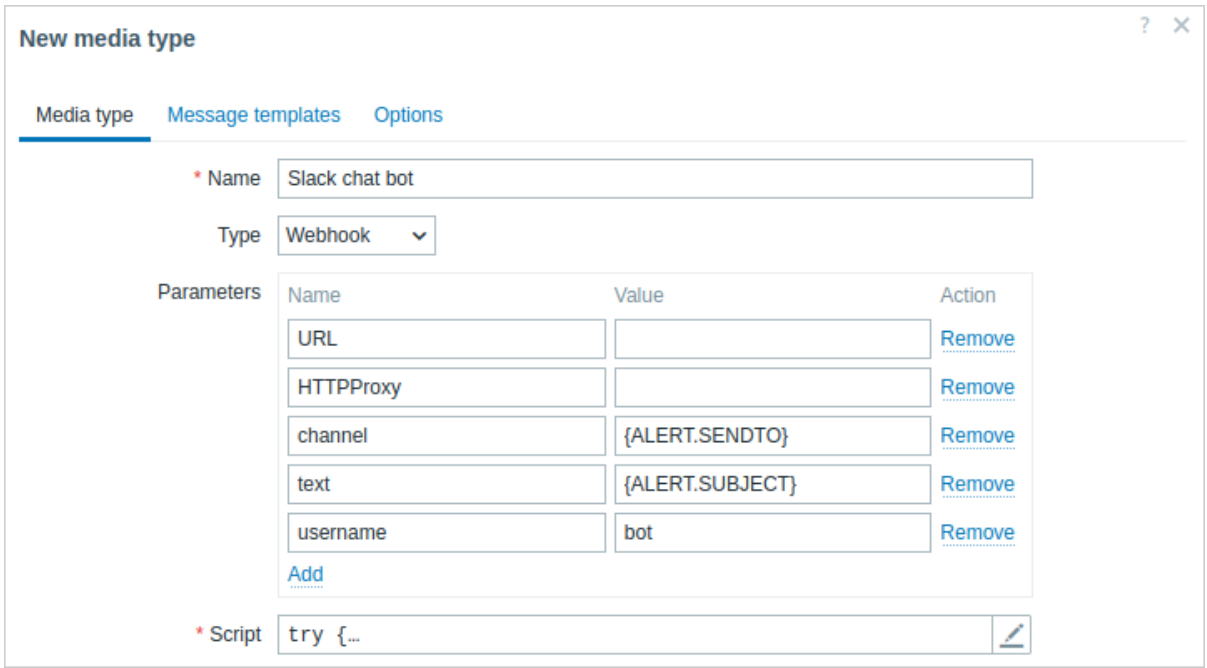
    resp = JSON.parse(resp);
    result.tags.issue_id = resp.id;
    result.tags.issue_key = resp.key;

    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 (пользовательский)

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.hook_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;
}
```

2 Действия

Обзор

Если вы хотите, чтобы в результате событий выполнялись какие-либо операции (например, отправлялись уведомления), необходимо настроить действия.

Действия могут быть определены в ответ на события всех поддерживаемых типов:

- Действия триггеров — для событий, когда состояние триггера изменяется между OK и
- Действия сервисов — для событий, когда состояние сервиса изменяется между OK и
- Действия обнаружения — для событий, когда выполняется обнаружение сети
- Действия авторегистрации — для событий, когда новые активные агенты проходят авторегистрацию (или изменяются метаданные узла сети у уже зарегистрированных)
- Внутренние действия — для событий, когда элементы данных становятся неподдерживаемыми или триггеры переходят в неизвестное состояние

Основные отличия действий сервисов:

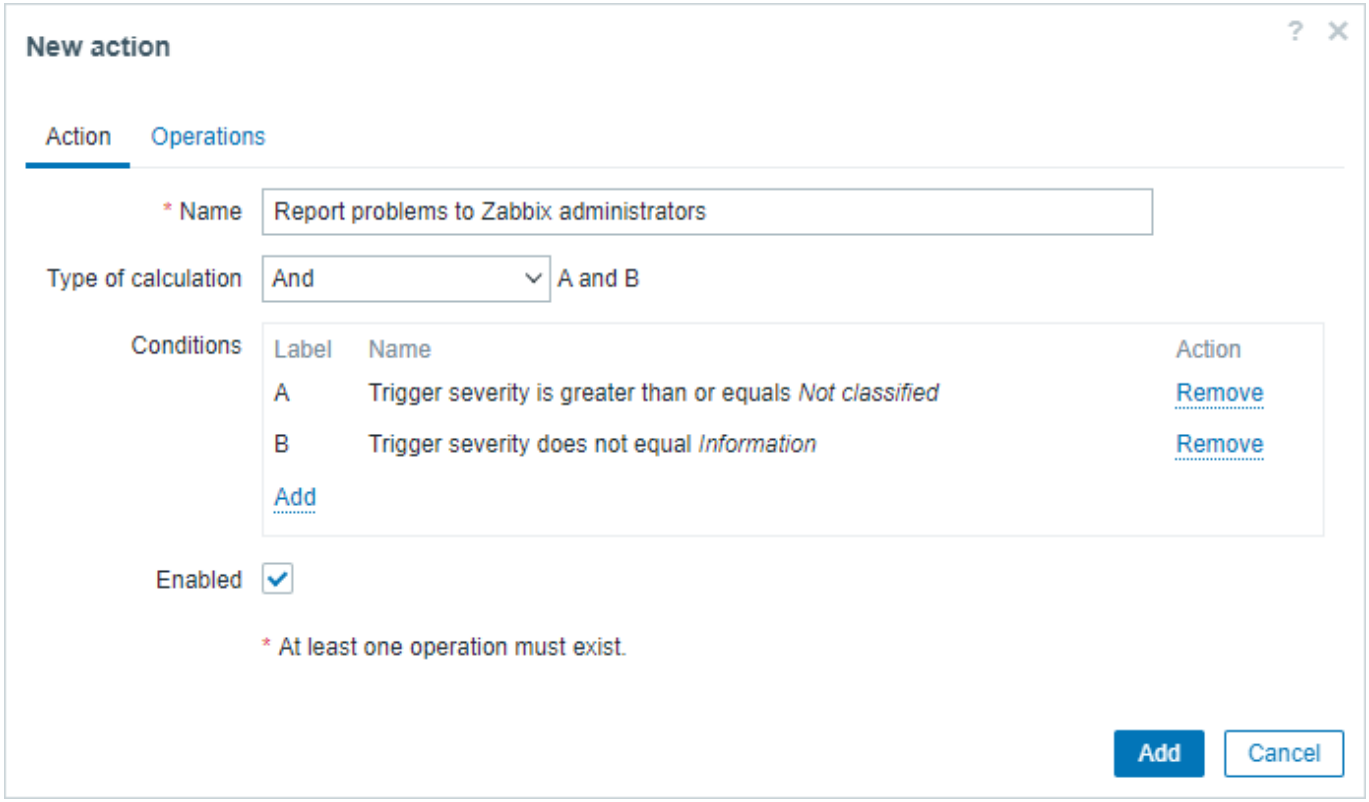
- Доступ пользователей к действиям сервисов зависит от прав доступа к сервисам, предоставленных ролью
- Действия сервисов поддерживают другой набор условий

Настройка действия

Чтобы настроить действие, выполните следующее:

- Перейдите в *Оповещения* > *Действия* и выберите требуемый тип действия в подменю (позже вы сможете переключиться на другой тип, используя выпадающий список в заголовке).
- Нажмите *Создать действие*.
- Укажите имя действия.
- Выберите условия, при которых будут выполняться операции.
- Выберите операции, которые необходимо выполнить.

Общие атрибуты действия:



Все обязательные поля ввода отмечены красной звёздочкой.

Параметр	Описание
Имя	Уникальное имя действия.
Тип вычисления	Выберите вариант вычисления для условий действия (если задано более одного условия): **И** - должны быть выполнены все условия. **Или** - достаточно выполнения одного условия. **И/Или** - комбинация двух вариантов: И для разных типов условий и Или для одинаковых типов условий. **Пользовательское выражение** - заданная пользователем формула вычисления для оценки условий действия.
Условия	Список условий действия. Нажмите *Добавить*, чтобы добавить новое условие. Если условия не настроены, действие будет выполняться для каждого события, соответствующего настраиваемому типу действия.
Включено	Установите флажок, чтобы включить действие. В противном случае оно будет отключено.

1 Условия

Обзор

Можно задать, что действие будет выполнено только, если событие удовлетворяет определенному набору условий. Условия задаются при настройке действия.

Сопоставление условий чувствительно к регистру.

Действия триггеров

Следующие условия могут использоваться в действиях на основе триггеров:

Тип условия	Поддерживаемые операторы	Описание
Группа узлов сети	равно не равно	Укажите группы узлов сети или группы узлов сети, которые нужно исключить. **равно** — событие принадлежит этой группе узлов сети. **не равно** — событие не принадлежит этой группе узлов сети. Указание родительской группы узлов сети неявно выбирает все вложенные группы узлов сети. Чтобы указать только родительскую группу, все вложенные группы необходимо дополнительно задать с оператором **не равно**.
Шаблон	равно не равно	Укажите шаблоны или шаблоны, которые нужно исключить. **равно** — событие относится к триггеру, унаследованному от этого шаблона. **не равно** — событие не относится к триггеру, унаследованному от этого шаблона.
Узел сети	равно не равно	Укажите узлы сети или узлы сети, которые нужно исключить. **равно** — событие принадлежит этому узлу сети. **не равно** — событие не принадлежит этому узлу сети.
Имя тега	равно не равно содержит не содержит	Укажите тег события или тег события, который нужно исключить. **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	равно не равно содержит не содержит	Укажите комбинацию тега и значения события или комбинацию тега и значения, которую нужно исключить. **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.
Триггер	равно не равно	Укажите триггеры или триггеры, которые нужно исключить. **равно** — событие сгенерировано этим триггером. **не равно** — событие сгенерировано любым другим триггером, кроме этого.
Имя события	содержит не содержит	Укажите строку в имени события, сгенерированного триггером, или строку, которую нужно исключить. По умолчанию имя события совпадает с именем триггера, если только в настройках триггера не указано пользовательское имя события. **содержит** — имя события содержит эту строку. **не содержит** — эта строка исключается из имени события. Примечание: введённое значение будет сравниваться с именем события после раскрытия всех макросов.
Важность триггера	равно не равно больше или равно меньше или равно	Укажите важность триггера. **равно** — равно важности триггера. **не равно** — не равно важности триггера. **больше или равно** — больше или равно важности триггера. **меньше или равно** — меньше или равно важности триггера.
Период времени	в не в	Укажите период времени или период времени, который нужно исключить. **в** — время события находится в пределах периода времени. **не в** — время события не находится в пределах периода времени. Описание формата см. на странице спецификация периода времени. Поддерживаются пользовательские макросы.
Проблема подавлена	нет да	Укажите, подавлена ли проблема (не отображается) из-за обслуживания узла сети. **нет** — проблема не подавлена. **да** — проблема подавлена.

Действия сервиса

Следующие условия могут использоваться в действиях сервиса:

Тип условия	Поддерживаемые операторы	Описание
Сервис	равно не равно	Укажите сервис или сервис, который необходимо исключить. **равно** — событие относится к этому сервису. **не равно** — событие не относится к этому сервису. При указании родительского сервиса неявно выбираются все дочерние сервисы. Чтобы указать только родительский сервис, необходимо дополнительно задать все вложенные сервисы с оператором **не равно**.
Имя сервиса	содержит не содержит	Укажите строку в имени сервиса или строку, которую необходимо исключить. **содержит** — событие сгенерировано сервисом, имя которого содержит эту строку. **не содержит** — эта строка не может быть найдена в имени сервиса.
Имя тега сервиса	равно не равно содержит не содержит	Укажите тег события или тег события, который необходимо исключить. Теги событий сервиса можно определить в разделе конфигурации сервиса *Теги*. **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 и значения события или комбинацию тега и значения, которую необходимо исключить. Теги событий сервиса можно определить в разделе конфигурации сервиса *Теги*. **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:
Убедитесь, что для действий сервиса определены шаблоны сообщений в меню *Оповещения* → *Способы оповещения*. В противном случае уведомления не будут отправляться.

Действия обнаружения

Следующие условия могут использоваться в событиях на основе обнаружения:

Тип условия	Поддерживаемые операторы	Описание
IP-адрес узла сети	равно не равно	Укажите диапазон IP-адресов или диапазон для исключения для обнаруженного узла сети. **равно** — IP-адрес узла сети находится в диапазоне. **не равно** — IP-адрес узла сети не находится в диапазоне. Может иметь следующие форматы: Один IP-адрес: 192.168.1.33 Диапазон IP-адресов: 192.168.1-10.1-254 Маска IP: 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 Пробелы в формате списка поддерживаются.

Тип условия	Поддерживаемые операторы	Описание
Тип сервиса	равно не равно	Укажите тип сервиса обнаруженного сервиса или тип сервиса для исключения. **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, ICMP ping, telnet.
Порт сервиса	равно не равно	Укажите диапазон TCP-портов обнаруженного сервиса или диапазон для исключения. **равно** — порт сервиса находится в диапазоне. **не равно** — порт сервиса не находится в диапазоне.
Правило обнаружения	равно не равно	Укажите правило обнаружения или правило обнаружения для исключения. **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 для исключения. **равно** — используется эта проверка обнаружения. **не равно** — используется любая другая проверка обнаружения, кроме этой.
Объект обнаружения	равно	Укажите обнаруженный объект. **равно** — равно обнаруженному объекту (устройству или сервису).
Статус обнаружения	равно	**В сети** — 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 работы/простоя	больше или равно меньше или равно	Время работы для событий «Узел сети доступен» и «Сервис доступен». Время простоя для событий «Узел сети недоступен» и «Сервис недоступен». **больше или равно** — больше или равно указанному значению. Параметр задаётся в секундах. **меньше или равно** — меньше или равно указанному значению. Параметр задаётся в секундах.
Полученное значение	равно не равно больше или равно меньше или равно содержит не содержит	Укажите значение, полученное из проверки агента (Zabbix, SNMP) в правиле обнаружения. Сравнение строковое. Если для правила настроено несколько проверок агента Zabbix или SNMP, проверяются полученные значения для каждой из них (каждая проверка генерирует 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 задаётся как строка. **не содержит** — не содержит подстроку. Параметр задаётся как строка.
Прокси	равно не равно	Укажите прокси или прокси для исключения. **равно** — используется этот прокси. **не равно** — используется любой другой прокси, кроме этого.

Note:
Проверки сервисов в правиле обнаружения, которые приводят к событиям обнаружения, не выполняются одновременно. Поэтому, если для условий `Service type`, `Service port` или `Received value` в действии настроено **несколько** значений, они будут сравниваться с одним событием обнаружения за раз, но **не** с несколькими событиями одновременно. В результате действия с несколькими значениями для одинаковых типов проверок могут выполняться некорректно.

Действия на авторегистрацию

В действиях на основе авторегистрации можно использовать следующие условия:

Тип условия	Поддерживаемые операторы	Описание
Метаданные узла сети	содержит не содержит совпадает не соответствует	Укажите метаданные узла сети или метаданные узла сети для исключения. **содержит** - метаданные узла сети содержат указанную строку **не содержит** - метаданные узла сети не содержат указанную строку. Метаданные узла сети можно задать в файле конфигурации агента **совпадает** - метаданные узла сети соответствуют регулярному выражению. **не соответствует** - метаданные узла сети не соответствуют регулярному выражению.
Имя узла сети	содержит не содержит совпадает не соответствует	Укажите имя узла сети или имя узла сети для исключения. **содержит** - имя узла сети содержит указанную строку **не содержит** - имя узла сети не содержит указанную строку. ** совпадает** - 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: **=** - используется указанный прокси. **<>** - используется любой другой прокси отличный от указанного.

Действия на внутренние события

В действиях на основе внутренних событий можно использовать следующие условия:

Тип условия	Поддерживаемые операторы	Описание
Тип события	равно	**Элемент данных в состоянии "не поддерживается"** - соответствует событию, когда элемент данных переходит из состояния 'активировано' в состояние 'не поддерживается' **Правило низкоуровневого обнаружения в состоянии "не поддерживается"** - соответствует событию, когда правило низкоуровневого обнаружения переходит из состояния 'активировано' в состояние 'не поддерживается' **Триггер в состоянии "неизвестно"** - соответствует событию, когда триггер переходит из состояния 'активировано' в состояние 'неизвестно'
Группа узла сети	равно не равно	Укажите группу узла 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.
Имя тега	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 имеет тег, который не содержит указанную строку

Тип условия	Поддерживаемые операторы	Описание
Значение тега	равно не равно содержит не содержит	Укажите комбинацию тега события и его значение 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	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 относиться к элементу данных / триггеру / правилу низкоуровневого обнаружения унаследованному из указанного шаблона.
Узел сети	равно не равно	Укажите узел сети или узел сети для исключения. **равно** - событие относиться к указанному узлу сети. **не равно** - событие не относиться к указанному узлу сети.

Тип вычисления

Доступны следующие варианты вычисления условий:

- **И** — все условия должны быть выполнены

Обратите внимание, что использование вычисления ”И” запрещено между несколькими триггерами, если они выбраны как условие `Trigger=`. Действия могут выполняться только на основе события одного триггера.

- **Или** — достаточно, чтобы было выполнено одно условие
- **И/Или** — комбинация двух вариантов: И для разных типов условий и Или для одинакового типа условий, например:

Группа узлов сети равно Oracle servers
Группа узлов сети равно MySQL servers
Имя события содержит ’Database is down’
Имя события содержит ’Database is unavailable’

оценивается как

(Группа узлов сети равно Oracle servers **or** Группа узлов сети равно MySQL servers**) and (**Имя события содержит ’Database is down’ **or** Имя события содержит ’Database is unavailable’**)**

- **Пользовательское выражение** — заданная пользователем 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).

В то время как предыдущий пример с `And/Or` был бы представлен как (A or B) and (C or D), в пользовательском выражении также можно использовать множество других способов 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 т. д.

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 удалённые объекты;
- содержат объекты, которые больше не обнаруживаются низкоуровневым обнаружением и были удалены (после периода *Delete lost resources*).

Условия/операции действия не удаляются, если в них отсутствуют объекты. Вместо этого удалённые объекты отображаются как *Deleted host group*, *Deleted host*, *Deleted trigger* и т. д. В столбце ”Info” списка действий отображается значок предупреждения с сообщением ”This action has conditions or operations referencing deleted object(s).”.

2 Операции

Обзор

Для всех событий можно определить следующие операции:

- Отправить сообщение
- Выполнить удалённую команду

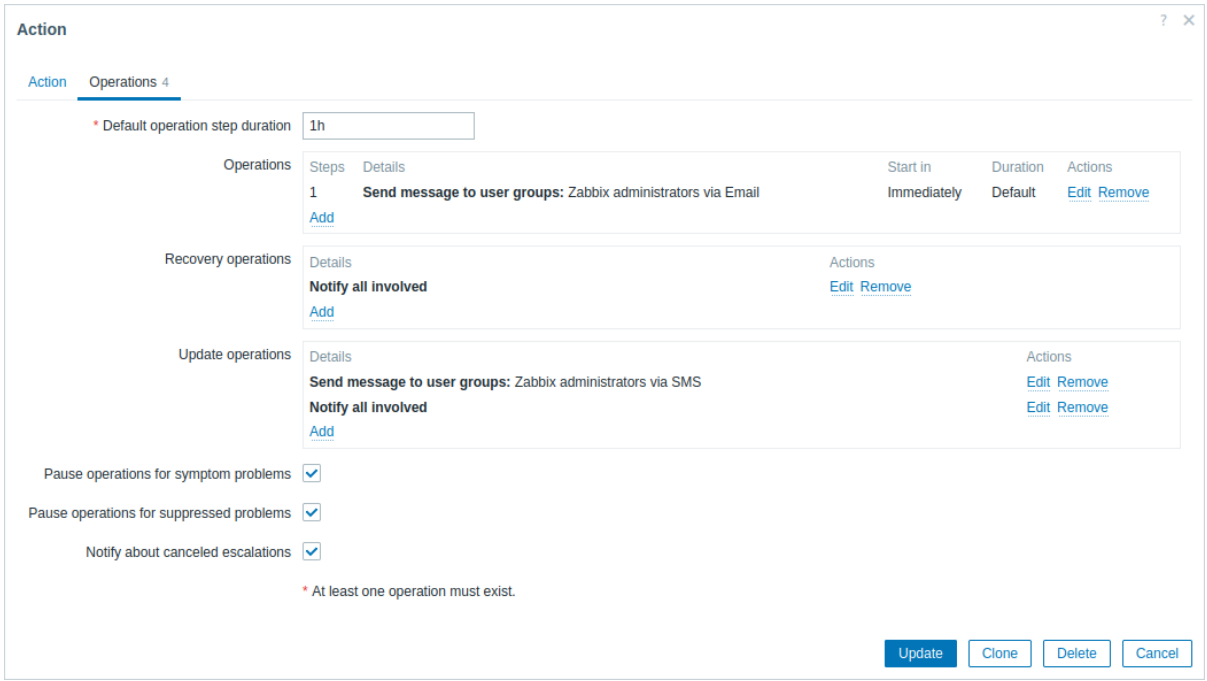
Attention:
Сервер Zabbix не создаёт оповещения, если доступ к узлу сети явно "запрещён" для пользователя, определённого как получатель операции действия, или если для пользователя вообще не определены права доступа к узлу сети.

Для событий обнаружения и авторегистрации доступны дополнительные операции:

- Добавить узел сети
- Удалить узел сети
- Включить узел сети
- Отключить узел сети
- Добавить в группу узлов сети
- Удалить из группы узлов сети
- Добавить теги узла сети
- Удалить теги узла сети
- Привязать шаблон
- Отвязать шаблон
- Установить режим инвентаризации узла сети

Настройка операции

Чтобы настроить операцию, перейдите на вкладку *Операции* в настройках action.



Общие атрибуты операции:

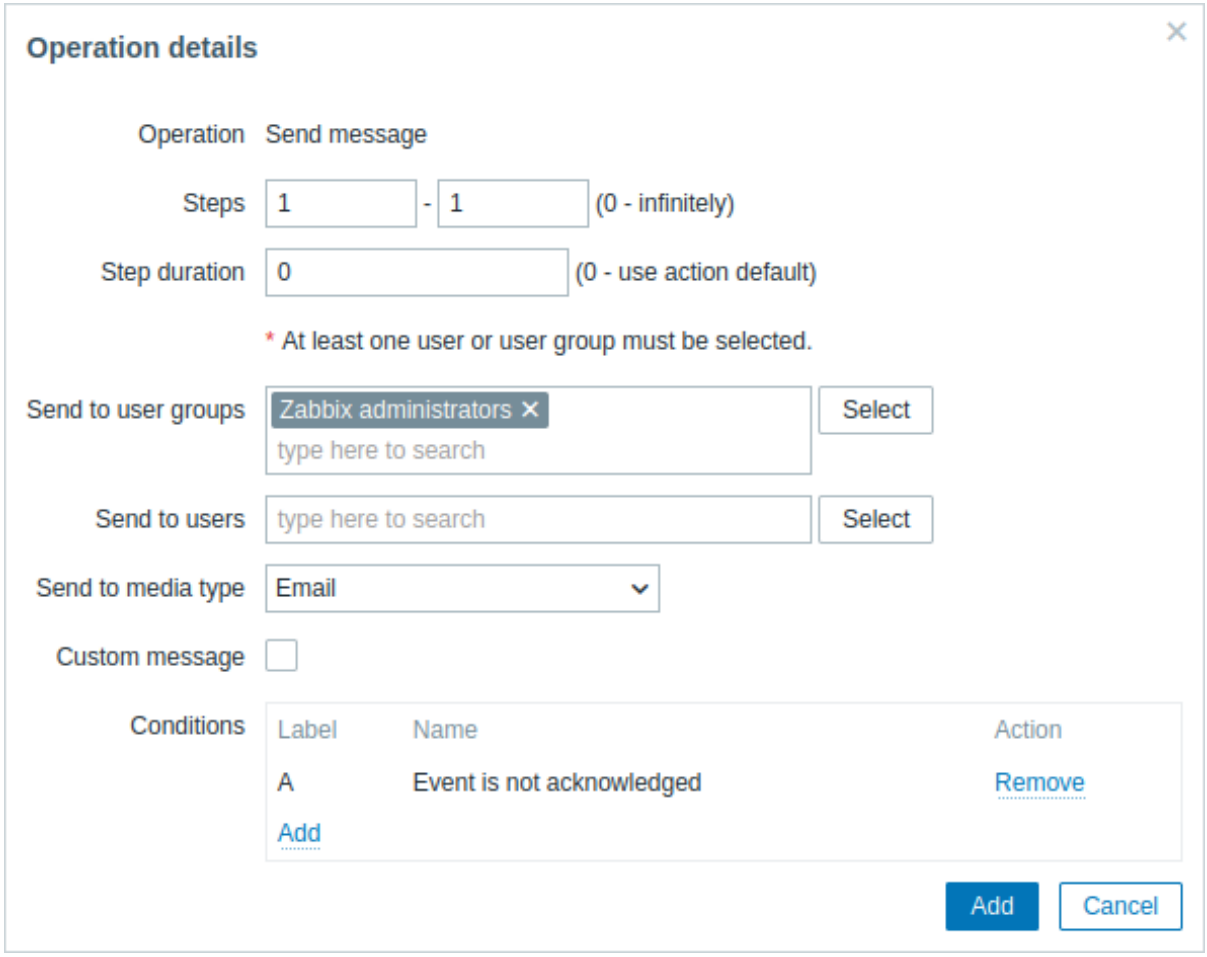
Параметр	Описание
Длительность шага операции по умолчанию	Длительность одного шага операции по умолчанию (от 60 секунд до 1 недели). Например, длительность шага в один час означает, что если операция выполняется, то до следующего шага пройдет один час. Поддерживаются суффиксы времени, например: 60s, 1m, 2h, 1d. Поддерживаются пользовательские макросы.

Параметр	Описание
Операции	Отображаются операции действия (если они есть) со следующими сведениями: **Шаги** — шаг(и) эскалации, которым назначена операция. **Сведения** — тип операции и ее получатель/цель. В списке операций также отображается используемый тип средства оповещения (email, SMS или скрипт), а также имя и фамилия (в скобках после имени пользователя) получателя уведомления. **Начать через** — через какое время после события выполняется операция. **Длительность (сек)** — отображается длительность шага. Отображается *По умолчанию*, если шаг использует длительность по умолчанию, и конкретное время, если используется пользовательская длительность. **Действия** — отображаются ссылки для редактирования и удаления операции.
Операции восстановления	Отображаются операции действия (если они есть) со следующими сведениями: **Сведения** — тип операции и ее получатель/цель. В списке операций также отображается используемый тип средства оповещения (email, 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: **Сведения** — тип операции и ее получатель/цель. В списке операций также отображается используемый тип средства оповещения (email, SMS или скрипт), а также имя и фамилия (в скобках после имени пользователя) получателя уведомления. **Действия** — отображаются ссылки для редактирования и удаления операции.
Приостанавливать операции для проблем-симптомов	Установите этот флажок, чтобы приостанавливать операции (после первой операции) для проблем-симптомов. Обратите внимание, что этот параметр влияет только на эскалации проблем; операции восстановления и обновления затронуты не будут. Этот параметр доступен только для *Действий триггера*.
Приостанавливать операции для подавленных проблем	Установите этот флажок, чтобы отложить начало операций на время периода обслуживания. Когда операции будут запущены после обслуживания, будут выполнены все операции, 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 сервиса*.
Уведомлять об отмененных эскалациях	Снимите этот флажок, чтобы отключить уведомления об отмененных эскалациях (когда узел сети, элемент данных, триггер или действие отключены).

Все обязательные для заполнения поля отмечены красной звездочкой.

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 шага операции.

Сведения об операции



Параметр	Описание
Операция	Выберите операцию: **Отправить сообщение** — отправить сообщение пользователю. **<имя удалённой команды>** — выполнить удалённую команду. Команды доступны для выполнения, если они были предварительно определены в глобальных скриптах с выбранной областью действия *Операция действия*. Для событий, основанных на обнаружении и авторегистрации, доступно больше операций (см. выше).
Шаги	Выберите шаг(и), которым будет назначена операция в расписании эскалации: **От** — выполнять, начиная с этого шага. **До** — выполнять до этого шага (0=бесконечность, выполнение не будет ограничено).
Длительность шага	Пользовательская длительность для этих шагов (0=использовать длительность шага по умолчанию). Поддерживаются суффиксы времени, например 60s, 1m, 2h, 1d. Поддерживаются пользовательские макросы. Несколько операций могут быть назначены одному и тому же шагу. Если для этих операций определена разная длительность шага, учитывается и применяется к шагу наименьшая из них.
Тип операции: отправка сообщения	
Отправить группам пользователей	Выберите группы пользователей, которым будет отправлено сообщение. Чтобы получать уведомления, группа пользователей должна иметь как минимум права доступа ”чтение” к узлу сети.
Отправить пользователям	Выберите пользователей, которым будет отправлено сообщение. Чтобы получать уведомления, пользователь должен иметь как минимум права доступа ”чтение” к узлу сети.

Параметр	Описание
Отправить через способ оповещения	Отправить сообщение через все доступные (настроенные и включённые) способы оповещения или только через один конкретный.
Пользовательское сообщение	Если выбрано, можно настроить пользовательское сообщение. Для уведомлений о внутренних событиях через вебхуки пользовательское сообщение является обязательным.
Тема	Тема пользовательского сообщения. Тема может содержать макросы. Ограничена 255 символами.
Сообщение	Пользовательское сообщение. Сообщение может содержать макросы. Его длина ограничена определённым количеством символов в зависимости от типа базы данных (подробнее см. Отправка сообщения).
Тип операции: удалённая команда	
Список целей	Выберите цели, на которых будет выполнена команда: **Текущий узел сети** — команда выполняется на узле сети триггера, вызвавшего событие проблемы. Этот вариант не будет работать, если в триггере несколько узлов сети. **Узел сети** — выберите узел(ы) сети, на которых будет выполнена команда. **Группа узлов сети** — выберите группу(ы) узлов сети, на которых будет выполнена команда. Указание родительской группы узлов сети неявно выбирает все вложенные группы узлов сети. Таким образом, удалённая команда также будет выполнена на узлах сети из вложенных групп. Команда на узле сети выполняется только один раз, даже если узел сети совпадает более одного раза (например, из нескольких групп узлов сети; индивидуально и из группы узлов сети). Список целей не имеет значения, если пользовательский скрипт выполняется на сервере Zabbix. Выбор большего количества целей в этом случае лишь приводит к тому, что скрипт будет выполнен на сервере большее число раз. 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 сервиса*, поскольку в этом случае удалённые 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 триггера*.

Когда закончите, нажмите *Добавить*, чтобы добавить операцию в список *Операции*.

1 Отправка сообщений

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 от Zabbix, необходимо:

- настроить способ оповещения, на который будет отправляться сообщение

Если операция выполняется вне периода времени When active, заданного для выбранного способа оповещения в настройках пользователя, сообщение не будет отправлено.

Уровень важности триггера по умолчанию (’Not classified’) **должен быть** отмечен в настройках способа оповещения пользователя, если вы хотите получать уведомления о событиях, не связанных с триггерами, таких как обнаружение, активная авторегистрация агента или внутренние события.

- настроить операцию действия, которая отправляет сообщение на один из настроенных способов оповещения

Attention:
Zabbix отправляет уведомления только тем пользователям, у которых есть как минимум права 'read' на узел сети, сгенерировавший событие. Доступен должен быть как минимум один узел сети из выражения триггера.

Вы можете настроить пользовательские сценарии отправки сообщений с помощью эскалаций.

Чтобы успешно получать и читать электронные письма от Zabbix, почтовые серверы/клиенты должны поддерживать стандартный формат 'SMTP/MIME email', поскольку Zabbix отправляет данные в UTF-8 (если тема содержит только символы ASCII, она не кодируется в UTF-8). Тема и тело сообщения кодируются в base64 в соответствии со стандартом формата 'SMTP/MIME email'.

Ограничение на размер сообщения после раскрытия всех макросов такое же, как и ограничение на размер сообщения для удалённых команд.

Отслеживание сообщений

Вы можете просматривать статус сообщений, отправленных в *Мониторинг → Проблемы*.

В столбце *Действия* вы можете видеть сводную информацию о выполненных действиях. Зеленые числа обозначают отправленные сообщения, красные — сообщения, отправка которых завершилась ошибкой. *В процессе* указывает, что действие было инициировано. *Не выполнено* сообщает, что ни одно действие не было успешно выполнено.

Если вы нажмете на время события, чтобы просмотреть сведения о событии, вы сможете увидеть подробности об отправленных (или не отправленных) из-за этого события сообщениях в блоке *Действия*.

В *Отчеты → Журнал действий* вы увидите подробности обо всех выполненных действиях для тех событий, для которых настроено действие.

2 Удаленные команды

Обзор

При помощи удаленных команд вы можете указать какие из предустановленных команд будут автоматически выполнены на наблюдаемом узле сети при соблюдении некоторых условий.

Таким образом, удаленные команды являются мощным механизмом для умного про-активного мониторинга.

Из наиболее очевидных применений функции вы можете попробовать:

- Автоматический перезапуск какого-либо приложения (веб-сервер, промежуточное ПО, CRM), если оно не отвечает
- Использовать команду IPMI 'reboot' для перезагрузки некоторого удаленного сервера, если он не отвечает на запросы
- Автоматическую очистку диска (удаление старых файлов, очистка /tmp), если закончилось место на диске
- Миграцию VM с одного физического сервера на другой, в зависимости от загрузки CPU
- Добавление новых узлов в облачную среду при недостаточных ресурсах CPU (диска, памяти, чего-угодно)

Настройка действий для удаленных команд аналогична настройке действий для отправки сообщений, с той лишь разницей, что вместо отправки сообщения Zabbix будет выполнять команду.

Удаленные команды могут выполняться на стороне Zabbix сервера, прокси или агента. Удаленные команды на Zabbix агенте могут выполняться напрямую Zabbix сервером или через Zabbix прокси. По умолчанию удаленные команды отключены как у Zabbix агента, так и у Zabbix прокси. Их можно включить:

- добавив параметр `AllowKey=system.run[*]` в конфигурацию агента;
- изменив параметр EnableRemoteCommands на значение '1' в конфигурации прокси.

Удаленные команды выполняемые Zabbix сервером запускаются как описано в разделе Выполнение команд, включая проверку кода выхода.

Удаленные команды будут выполнены даже если целевой узел сети в обслуживании.

Ограничение удалённой команды

Ограничение удалённой команды после разрешения всех макросов зависит от типа базы данных и набора символов (для хранения не-ASCII-символов требуется более одного байта):

База данных	Ограничение в символах	Ограничение в байтах
MySQL	65535	65535
PostgreSQL	65535	не ограничено

База данных	Ограничение в символах	Ограничение в байтах
SQLite (только Zabbix прокси)	65535	не ограничено

Вывод выполнения удалённой команды (возвращаемое значение) ограничен 16 МБ (включая конечные пробельные символы, которые усекаются). Ограничение удалённой команды IPMI зависит от установленной библиотеки IPMI. Обратите внимание, что ограничения базы данных применяются ко всем удалённым командам.

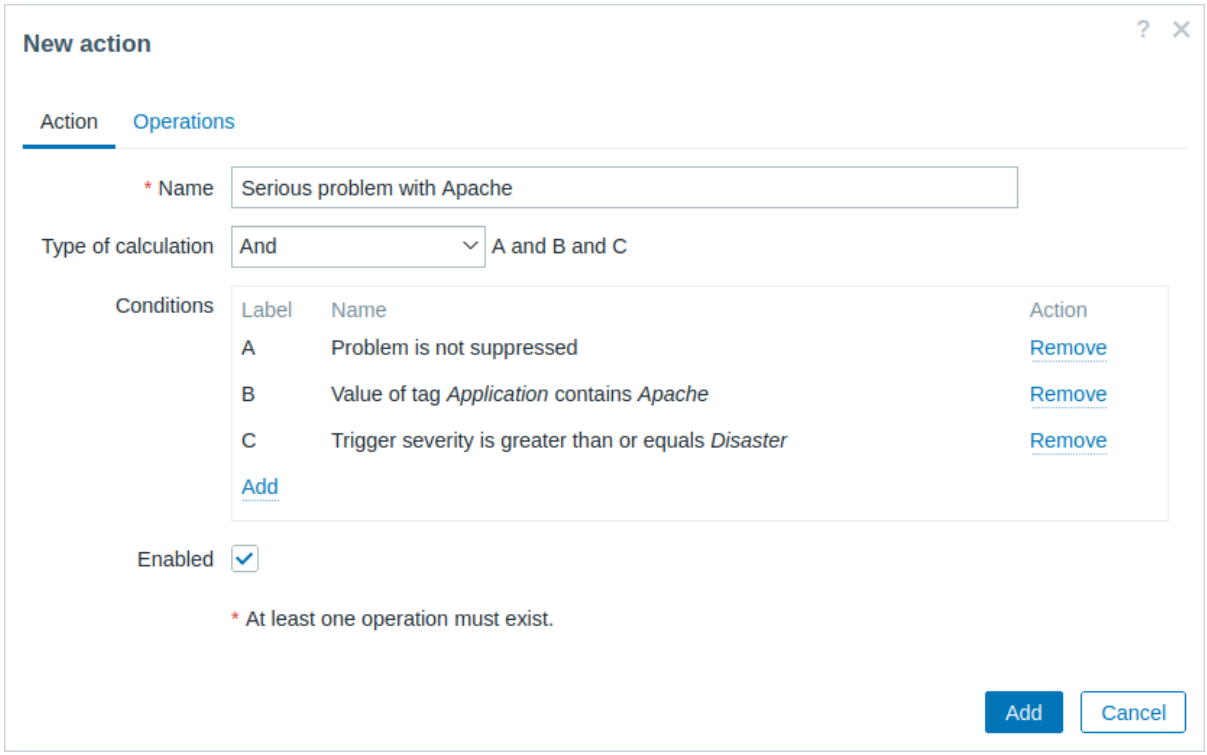
Конфигурация

Те удалённые команды, которые выполняются на Zabbix агенте (пользовательские скрипты), необходимо сначала включить в конфигурации агента.

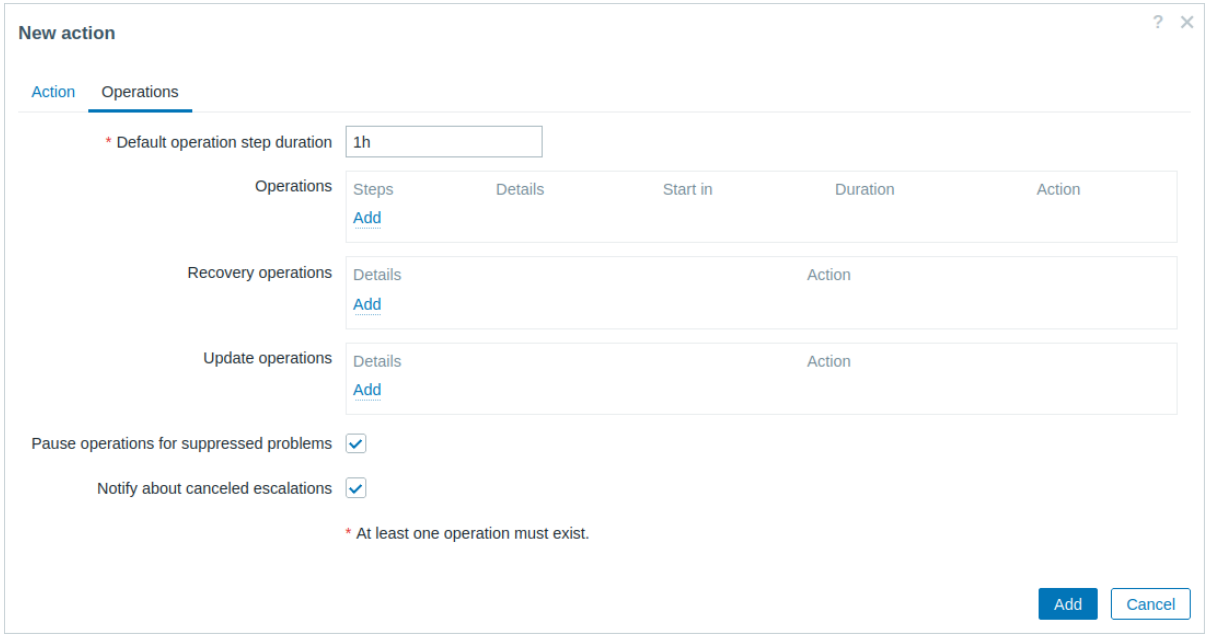
Убедитесь, что параметр AllowKey=system.run[<command>,*] добавлен для каждой разрешённой команды в конфигурации агента, чтобы разрешить выполнение конкретной команды в режиме nowait. Если вы изменили этот параметр, перезапустите процесс агента.

Затем при настройке нового действия в *Оповещения* → *Действия* → *Действия триггеров*:

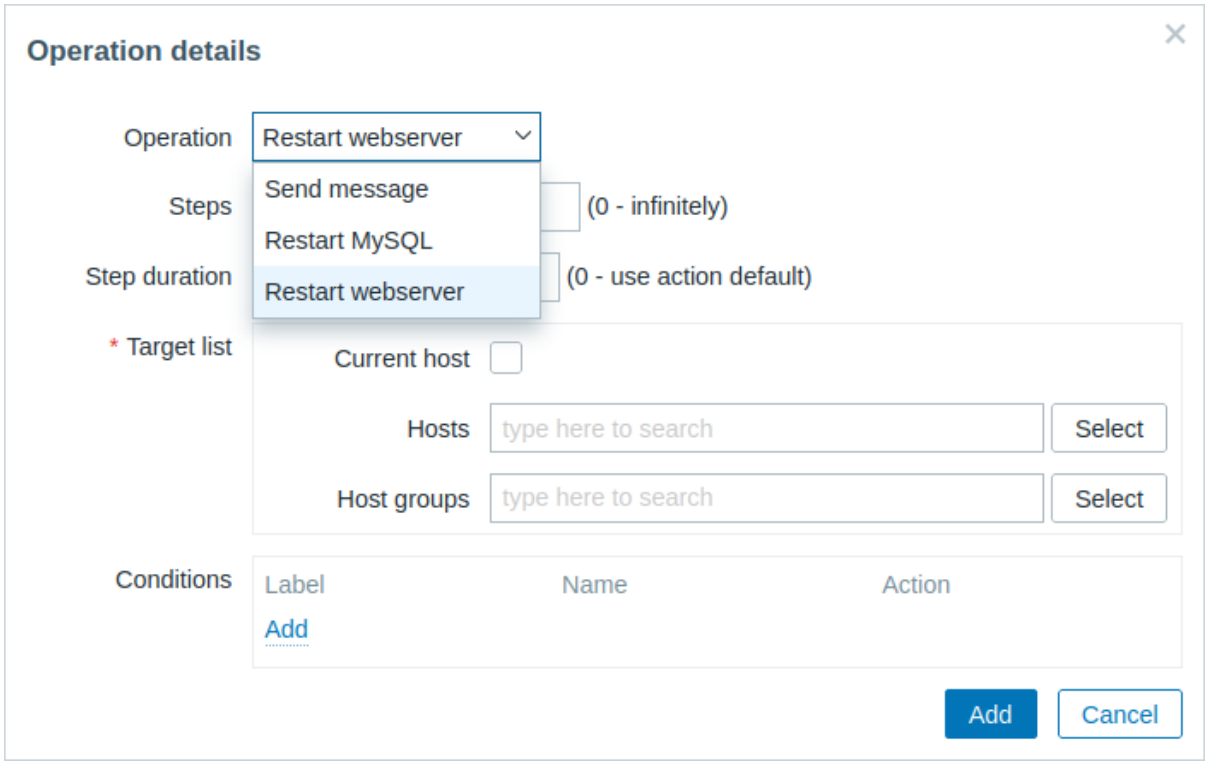
1. Задайте соответствующие условия, например, укажите, что действие активируется при любых проблемах уровня disaster для одного из приложений Apache.



2. На вкладке *Операции* нажмите *Добавить* в блоке *Операции*, *Операции восстановления* или *Операции обновления*.



3. Выберите один из предопределённых скриптов из выпадающего списка *Операция* и задайте *Целевой список* для скрипта.



Предопределенные скрипты

Скрипты, доступные для операций действий (вебхук, скрипт, SSH, Telnet, IPMI), определяются в глобальных скриптах.

Например:

```
sudo /etc/init.d/apache restart
```

В этом случае Zabbix попытается перезапустить процесс Apache. При использовании этой команды убедитесь, что команда выполняется на Zabbix агенте (нажмите кнопку *Zabbix агент* напротив поля *Выполнить на*).

Attention:
Обратите внимание на использование **sudo** — пользователь Zabbix по умолчанию не имеет прав на перезапуск системных служб. Ниже приведены рекомендации по настройке **sudo**.

Note:
Начиная с Zabbix агент 7.0, удаленные команды также могут выполняться на агенте, работающем в активном режиме. Zabbix агент — активный или пассивный — должен быть запущен на удаленном узле сети и выполняет команды в фоновом режиме.

Удаленные команды на Zabbix агенте выполняются без тайм-аута с помощью ключа system.run[,nowait] и не проверяются на результат выполнения. На Zabbix сервере и Zabbix прокси удаленные команды выполняются с тайм-аутом, заданным параметром TrapperTimeout в файле zabbix_server.conf или zabbix_proxy.conf, и проверяются на результат выполнения. Дополнительную информацию см. в разделе *Тайм-аут скрипта*.

Права доступа

Убедитесь, что пользователь 'zabbix' имеет права на выполнение для настроенных команд. Может быть полезно использовать **sudo** для предоставления доступа 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 позволяет нелокальным пользователям выполнять команды. Чтобы изменить это, закомментируйте опцию **requiretty** в */etc/sudoers*.

Удаленные команды с несколькими интерфейсами

Если целевая система имеет несколько интерфейсов выбранного типа (агент Zabbix или IPMI), удаленные команды будут выполняться через интерфейс по умолчанию.

Возможно выполнять удаленные команды через SSH и Telnet, используя интерфейс, отличный от интерфейса агента Zabbix. Доступный интерфейс для использования выбирается в следующем порядке:

- интерфейс агента Zabbix по умолчанию
- интерфейс SNMP по умолчанию
- интерфейс JMX по умолчанию
- интерфейс IPMI по умолчанию

Удаленные команды через IPMI

В случае использования удаленных команд через IPMI нужно использовать следующий синтаксис:

```
<     > [<      >]
```

где

- <команда> - одна из IPMI команд без пробелов
- <значение> - 'on', 'off' или любое положительное целое число. <значение> - опциональный параметр.

Примеры

Примеры глобальных скриптов, которые могут использоваться как удаленные команды в операциях действий.

Пример 1

Перезапуск Windows при определённом условии.

Чтобы автоматически перезапускать Windows при обнаружении проблемы Zabbix, определите следующий скрипт:

Параметр скрипта	Значение
Область	'Операция действия'
Тип	'Скрипт'
Команда	c:\windows\system32\shutdown.exe -r -f

Пример 2

Перезапустите узел сети, используя управление IPMI.

Параметр скрипта	Значение
Область	'Операция действия'
Тип	'IPMI'
Команда	reset

Пример 3

Выключение узла сети с использованием управления IPMI.

Параметр скрипта	Значение
Область	'Операция действия'
Тип	'IPMI'
Команда	power off

3 Дополнительные операции

Обзор

В этом разделе вы можете найти некоторые сведения о дополнительным операциям на события обнаружения / автоматической регистрации.

Добавление узла сети

Узлы сети добавляются в процессе обнаружения; как только узел сети обнаружен, он добавляется, а не в конце процесса обнаружения.

Note:
Поскольку сетевое обнаружение может занять некоторое время из-за большого количества недоступных узлов сети/сервисов, рекомендуется запастись терпением и использовать разумные диапазоны IP-адресов.

При добавлении узла сети его имя определяется стандартной функцией **gethostbyname**. Если узел сети может быть разрешён, используется разрешённое имя. Если нет, используется IP-адрес. Кроме того, если для имени узла сети необходимо использовать IPv6-адрес, то все ":" (двоеточия) заменяются на "_" (символы подчёркивания), поскольку двоеточия не допускаются в именах узлов сети.

Attention:
При выполнении обнаружения через прокси в настоящее время поиск имени узла сети всё равно выполняется на сервере Zabbix.

Attention:
Если в конфигурации Zabbix уже существует узел сети с тем же именем, что и у нового обнаруженного узла сети, Zabbix добавит **_N** к имени узла сети, где **N** — возрастающее число, начиная с 2.

4 Использование макросов в сообщениях

Обзор

В темах сообщений и тексте сообщений можно использовать макросы для более эффективного оповещения о проблемах.

Помимо ряда встроенных макросов, также поддерживаются пользовательские макросы и макросы выражений. Доступен полный список макросов, поддерживаемых Zabbix.

Примеры

Примеры приведенные здесь, иллюстрируют то, как вы можете использовать макросы в сообщениях.

Пример 1

Тема сообщения:

```
Problem: {TRIGGER.NAME}
```

Когда вы получите сообщение, тема сообщения будет заменена примерно на следующее:

```
Problem:                              Zabbix
```

Пример 2

Сообщение:

```
          : {?last(/zabbix.zabbix.com/system.cpu.load[,avg1])}
```

Когда вы получите сообщение, оно будет заменено примерно на следующее:

```
          : 1.45
```

Пример 3

Сообщение:

```
          : {?last(/{HOST.HOST}/{ITEM.KEY})}
MAX   15    : {?max(/{HOST.HOST}/{ITEM.KEY},15m)}
MIN   15    : {?min(/{HOST.HOST}/{ITEM.KEY},15m)}
```

Когда вы получите сообщение, оно будет заменено примерно на следующее:

```
          : 1.45
MAX   15    : 2.33
MIN   15    : 1.01
```

Пример 4

Сообщение:

```
http://<server_ip_or_name>/zabbix/tr_events.php?triggerid={TRIGGER.ID}&eventid={EVENT.ID}
```

Когда вы получите сообщение, оно будет содержать ссылку на страницу *Сведения о событии*, которая предоставляет информацию о событии, его триггере и список последних событий, сгенерированных тем же триггером.

Пример 5

Информирование о значениях с нескольких узлов сети в выражении триггера.

Сообщение:

```
       : {TRIGGER.NAME}
          : {TRIGGER.EXPRESSION}

1.                  {HOST.NAME1}: {ITEM.VALUE1} ({ITEM.NAME1})
2.                  {HOST.NAME2}: {ITEM.VALUE2} ({ITEM.NAME2})
```

Когда вы получите сообщение, оно будет заменено примерно на следующее:

```
       : Processor load is too high on a local host
          : last(/Myhost/system.cpu.load[percpu,avg1])>5 or last(/Myotherhost/system.cpu.load[percpu,avg1]

1.                  Myhost: 0.83 (Processor load (1 min average per core))
2.                  Myotherhost: 5.125 (Processor load (1 min average per core))
```

Пример 6

Получение сведений как о событии проблемы, так и о событии восстановления в сообщении восстановления:

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}
```

Когда вы получите сообщение, макросы будут заменены примерно на следующее:

```
    :

ID      : 21874
          : 1
         : PROBLEM
        : 13:04:30
       : 2018.01.02
         : 5m 0s
            :
               : 2018.01.02 13:05:51 "John Smith (Admin)"
    :        .

        :

ID      : 21896
          : 0
         : OK
        : 13:10:07
       : 2018.01.02
          :               0.83
```

3 Операции восстановления

Обзор

Операции восстановления позволяют вам получить оповещения при решении проблем.

В операциях восстановления поддерживаются как сообщения, так и удаленные команды. Хоть можно добавить несколько операций, эскалации не поддерживаются - все операции назначаются на единственный шаг, соответственно будут выполняться одновременно.

Варианты использования

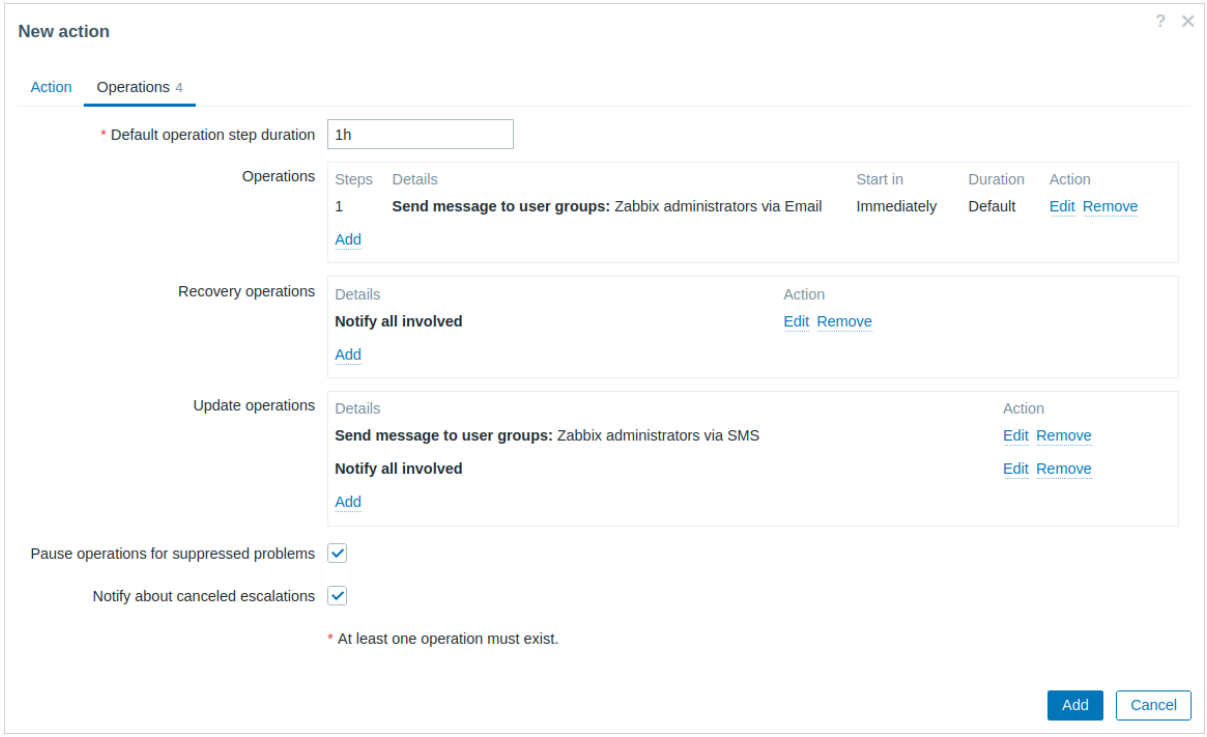
Некоторые варианты использования операций восстановления:

1. Уведомить при восстановлении всех пользователей, которые были уведомлены о проблеме:
   - Выберите *Notify all involved* в качестве типа операции.
2. Выполнить несколько операций при восстановлении: отправить уведомление и выполнить удаленную команду:
   - Добавьте типы операций для отправки сообщения и выполнения команды.
3. Открыть заявку во внешней системе helpdesk/тикетов и закрыть ее, когда проблема будет решена:
   - Создайте внешний скрипт, который взаимодействует с системой helpdesk.
   - Создайте действие с операцией, которая выполняет этот скрипт и таким образом открывает заявку.
   - Настройте операцию восстановления, которая выполняет этот скрипт с другими параметрами и закрывает заявку.
   - Используйте макрос {EVENT.ID} для ссылки на исходную проблему.

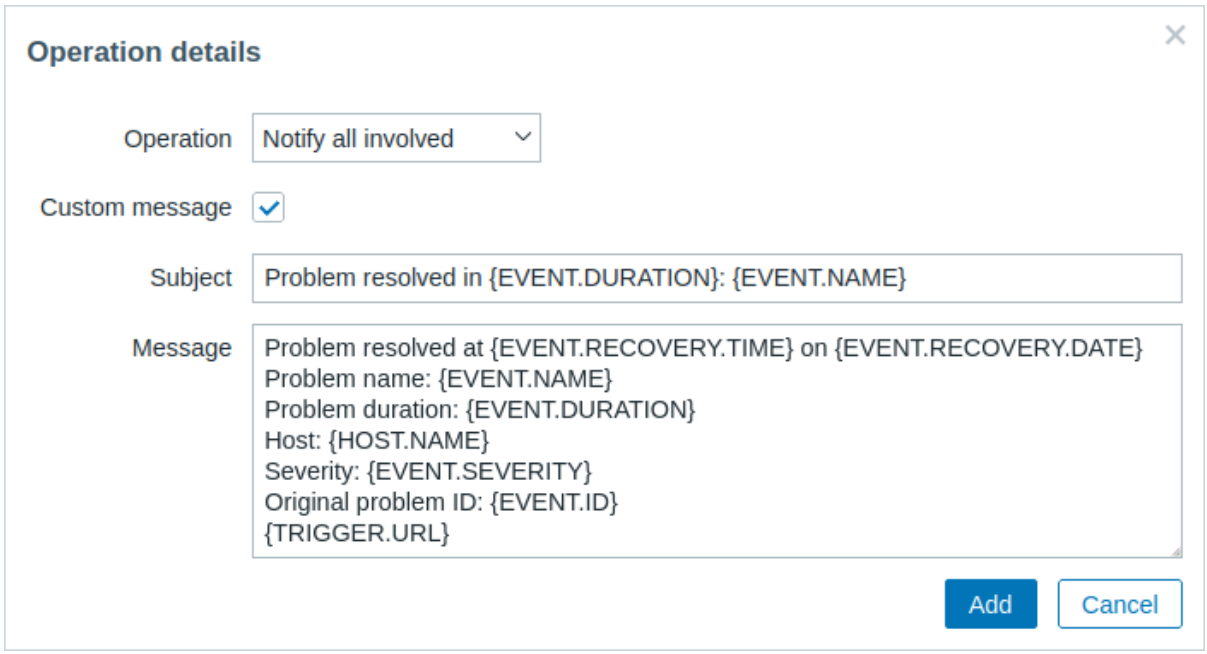
Настройка операции восстановления

Чтобы настроить операцию восстановления, выполните следующие шаги:

1. Перейдите на вкладку *Operations* в форме настройки действия.
2. Чтобы настроить новую операцию восстановления, нажмите *Add* в разделе *Recovery operations*. Чтобы изменить существующую операцию, нажмите *Edit* рядом с операцией.
3. Настройте параметры операции.



Сведения об операции восстановления



Для событий восстановления доступны три типа операций:

- **Отправить сообщение** — отправить сообщение о восстановлении указанному пользователю;
- **Уведомить всех вовлечённых** — отправить сообщение о восстановлении всем пользователям, которые были уведомлены о событии проблемы;
- **<имя удалённой команды>** — выполнить удалённую команду. Команды доступны для выполнения, если они были предварительно определены в глобальных скриптах с выбранной областью действия *Action operation*.

Параметры для каждого типа операции описаны ниже. Все обязательные поля ввода отмечены красной звёздочкой. После завершения нажмите *Добавить*, чтобы добавить операцию в список *Операции восстановления*.

Note:
Обратите внимание, что если один и тот же получатель указан в нескольких типах операций без заданного *Пользовательского сообщения*, дублирующиеся уведомления не отправляются.

Тип операции: отправить сообщение

Параметр	Описание
Отправить группам пользователей	Выберите группы пользователей, которым следует отправить сообщение о восстановлении. Чтобы получать уведомления, группа пользователей должна иметь как минимум права на чтение для узла сети.
Отправить пользователям	Выберите пользователей, которым следует отправить сообщение о восстановлении. Чтобы получать уведомления, пользователь должен иметь как минимум права на чтение для узла сети.
Отправить на способ оповещения	Отправить сообщение о восстановлении по умолчанию на все доступные (настроенные и включённые) способы оповещения или только на один конкретный.
Пользовательское сообщение	Если выбрано, можно задать пользовательское сообщение.
Тема	Тема пользовательского сообщения. Тема может содержать макросы.
Сообщение	Пользовательское сообщение. Сообщение может содержать макросы.

Тип операции: удалённая команда

Параметр	Описание
Список целей	Выберите цели для выполнения команды: **Текущий узел сети** — команда выполняется на узле сети триггера, который вызвал событие проблемы. Этот вариант не будет работать, если в триггере несколько узлов сети. **Узел сети** — выберите узел(ы) сети, на котором(ых) нужно выполнить команду. **Группа узлов сети** — выберите группу(ы) узлов сети, на которых нужно выполнить команду. Указание родительской группы узлов сети неявно выбирает все вложенные группы узлов сети. Таким образом, удалённая 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, из нескольких групп узлов сети; индивидуально и из группы узлов сети). Список целей не имеет значения, если команда выполняется на сервере Zabbix. Выбор большего количества целей в этом случае приводит только к тому, что команда будет выполнена на сервере большее число раз. Обратите внимание, что для глобальных скриптов выбор цели также зависит от настройки *Группа узлов сети* в конфигурации глобального скрипта.

Тип операции: уведомить всех участников

Параметр	Описание
Пользовательское сообщение	Если выбрано, можно определить пользовательское сообщение.
Тема	Тема пользовательского сообщения. Тема может содержать макросы.
Сообщение	Пользовательское сообщение. Сообщение может содержать макросы.

4 Операции обновления

Обзор

Операции обновления доступны в действиях со следующими источниками событий:

- *Триггеры* — когда проблемы обновляются другими пользователями, то есть к ним добавляются комментарии, они подтверждаются, для них изменяется важность, они закрываются (вручную);
- *Сервисы* — когда важность сервиса изменилась, но сервис все еще не восстановлен.

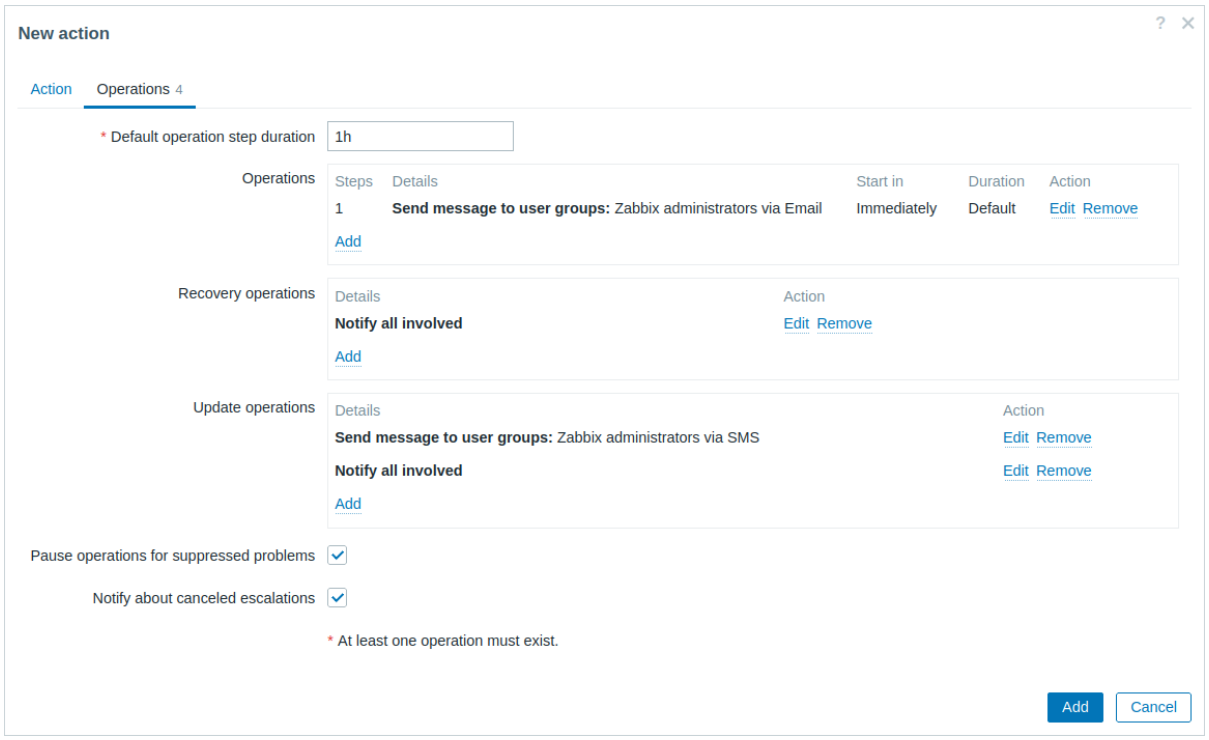
Обратите внимание, что пользователи не получают уведомления о собственных обновлениях.

В операциях обновления поддерживаются как сообщения, так и удаленные команды. Хотя можно добавить несколько операций, эскалация не поддерживается — все операции назначаются на один шаг и, следовательно, будут выполняться одновременно.

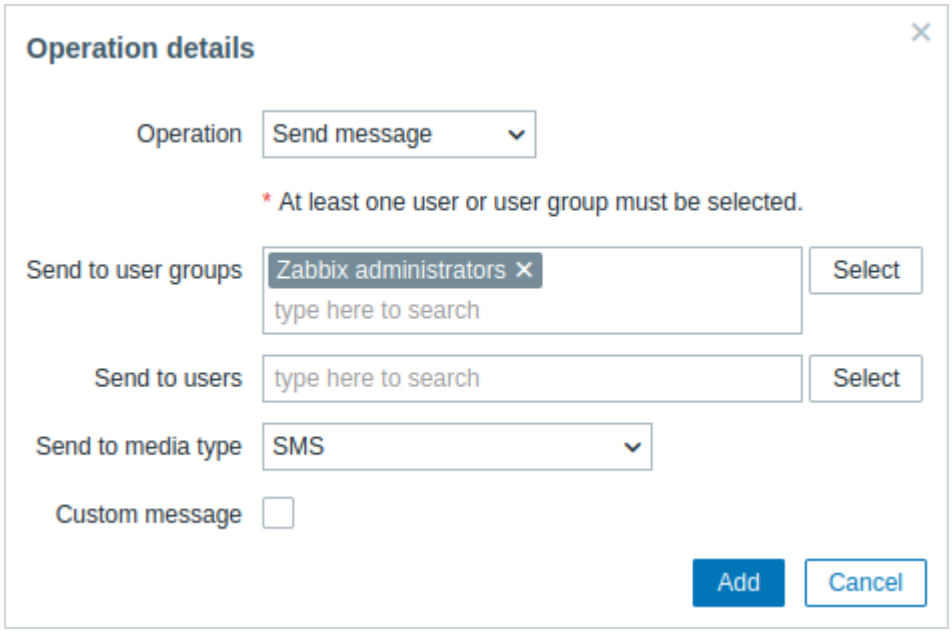
Настройка операции обновления

Чтобы настроить операцию обновления, выполните следующие шаги:

1. Перейдите на вкладку *Операции* в форме настройки действия.
2. Чтобы настроить новую операцию обновления, нажмите *Добавить* в разделе *Операции обновления*. Чтобы изменить существующую операцию, нажмите *Изменить* рядом с операцией.
3. Настройте параметры операции.



Детали операции обновления



Операции обновления предлагают тот же набор параметров, что и операции восстановления.

5 Эскалации

Обзор

Используя эскалации, вы можете создавать пользовательские сценарии для отправки оповещений или выполнения удаленных 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
- Удаленные 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 операциями на проблемы, не восстановления.

Различные аспекты поведения эскалаций

Давайте рассмотрим что произойдет при разных обстоятельствах, если действие содержит несколько шагов эскалаций.

Ситуация	Поведение
Узел сети, о котором идет речь, переходит в состояние обслуживания после отправки первоначального оповещения о проблеме	В зависимости от опции *Приостановить операции для подавленных проблем* в настройках действия, все оставшиеся шаги эскалаций выполнятся либо с задержкой, вызванной периодом обслуживания, или без задержки. Период обслуживания не отменяет операции.
*Период времени указанный в условии **Период времени** действия завершается после первоначальной отправки оповещения*	Выполнятся все оставшиеся шаги эскалации. Условие *Период времени* не может прервать операции; это условие имеет влияние в отношении действия началось/не началось, но не имеет влияния на операции.
Проблема началась в процессе обслуживания и продолжается (не исправлена) после окончания обслуживания	В зависимости от опции *Приостановить операции для подавленных проблем* в настройках действия, все шаги выполняются либо с момента завершения обслуживания или сразу же.
Проблема началась в процессе обслуживания без сбора данных и продолжается (не исправлена) после окончания обслуживания	Действие должно дождаться пока триггер не перейдет в состояние проблема, до выполнения всех шагов эскалации.
Разные эскалации следуют в тесном порядке и перекрываются	Выполнение каждой новой эскалации заменяет предыдущую эскалацию, но по крайней мере для одного шага эскалации, который всегда обязательно выполняется при предыдущей эскалации. Такое поведение тесно связано с действиями на события, которые создаются на КАЖДОЕ вычисление проблемы триггера.

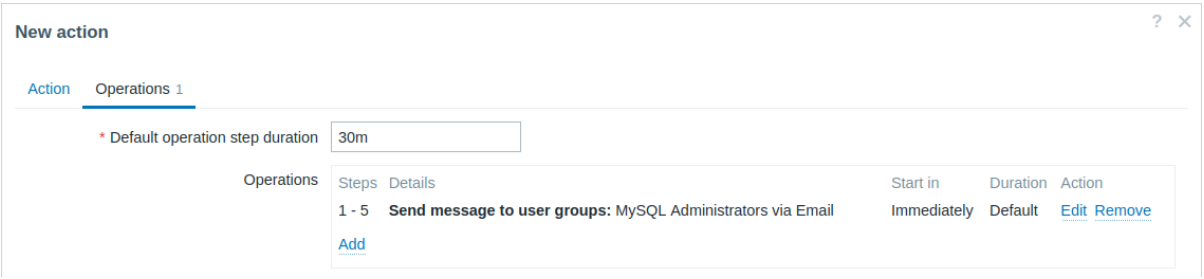
Ситуация	Поведение
В процессе выполнения эскалации (например, во время отправки сообщения), основанном на любом типе событий: - действие деактивировано - событие удалено На основе события на триггер: - триггер деэактивирован или удален - узел сети или элемент данных деактивирован На основе внутреннего события о триггере: - триггер деактивирован или удален На основе внутреннего события о элементах данных/правилах низкоуровневого обнаружения: - элемент данных деактивирован или удален - узел сети деактивирован	Будет отправлено сообщение, которое уже в процессе отправки, затем будет отправлено еще одно сообщение эскалации. Последующие сообщения будут иметь следующий текст в начале тела сообщения (*NOTE: Escalation cancelled*) с указанием причины (например, *NOTE: Escalation cancelled: action '<Имя действия>' disabled*). Таким образом получатель будет проинформирован о том, что эскалация отменена и дальнейшие шаги не будут выполнены. Это сообщение будет отправлено все кто получал оповещения ранее. Причина отмены также будет записана в файл журнала сервера (начиная с уровня отладки 3=Предупреждения). Обратите внимание, что сообщение *Escalation canceled* также будет отправлено, если все операции завершены, а также операции восстановления настроены, но еще не выполнялись.
В процессе выполнения эскалации (например, во время отправки сообщения) удалено действие	Сообщения более не отсылаются. Информация будет записана в файл журнала сервера (начиная с уровня отладки 3=Предупреждения), например: `escalation cancelled: action id:334 deleted`

Примеры эскалаций

Пример 1

Отправка повторяющегося уведомления один раз каждые 30 минут (всего 5 раз) группе "MySQL Administrators". Для настройки:

- На вкладке *Операции* установите *Стандартную длительность шага операции* в значение "30m" (30 минут).
- Установите *Шаги* эскалации от "1" до "5".
- Выберите группу "MySQL Administrators" в качестве получателей сообщения.



Уведомления будут отправлены через 0:00, 0:30, 1:00, 1:30, 2:00 часа после начала проблемы (если, конечно, проблема не будет устранена раньше).

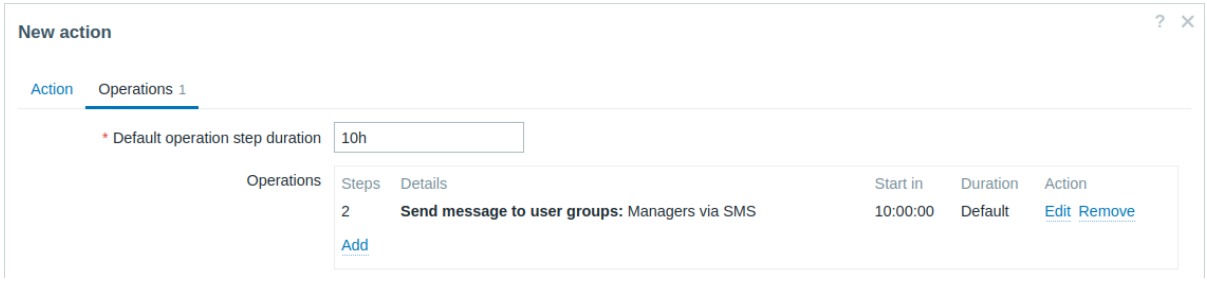
Если проблема устранена и настроено сообщение о восстановлении, оно будет отправлено тем, кто получил хотя бы одно сообщение о проблеме в рамках данного сценария эскалации.

Note:
Если триггер, создавший активную эскалацию, отключён, Zabbix отправит информационное сообщение об этом всем, кто уже получил уведомления.

Пример 2

Отправка отложенного уведомления о длительно существующей проблеме. Для настройки:

- На вкладке *Operations* установите *Default operation step duration* в значение "10h" (10 часов).
- Установите *Steps* эскалации от "2" до "2".



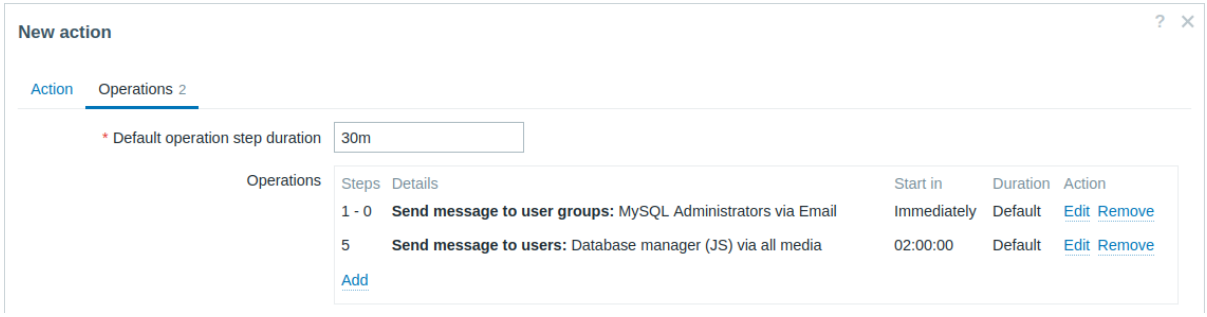
Уведомление будет отправлено только на шаге 2 сценария эскалации, или через 10 часов после возникновения проблемы.

Вы можете настроить текст сообщения, например: ”Проблеме уже более 10 часов”.

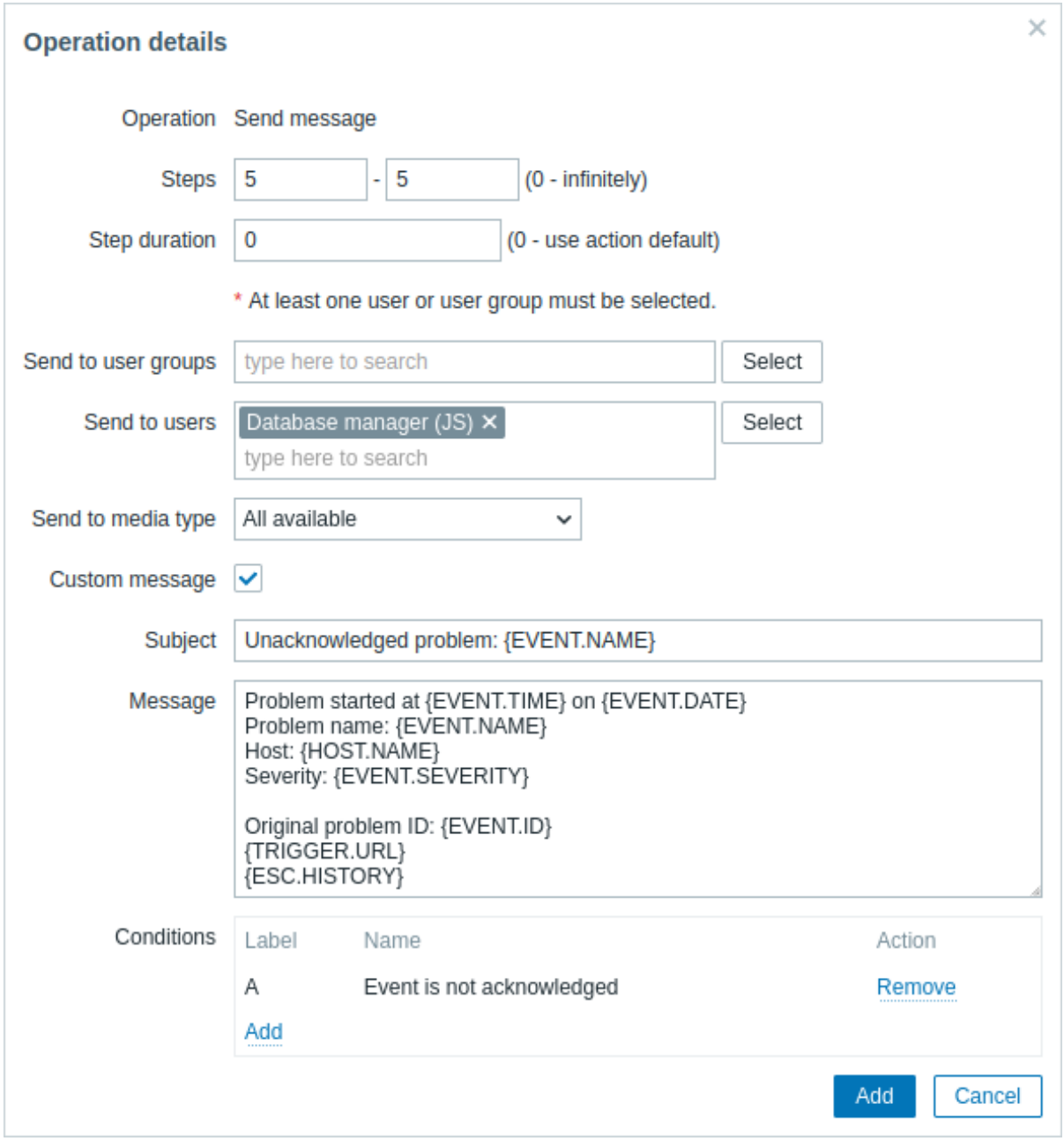
Пример 3

Эскалация проблемы руководителю.

В первом примере выше мы настроили периодическую отправку сообщений администраторам MySQL. В этом случае администраторы получат четыре сообщения, прежде чем проблема будет эскалирована менеджеру базы данных. Обратите внимание, что менеджер получит сообщение только в том случае, если проблема еще не подтверждена, то есть, предположительно, никто ею не занимается.



Подробности операции 2:



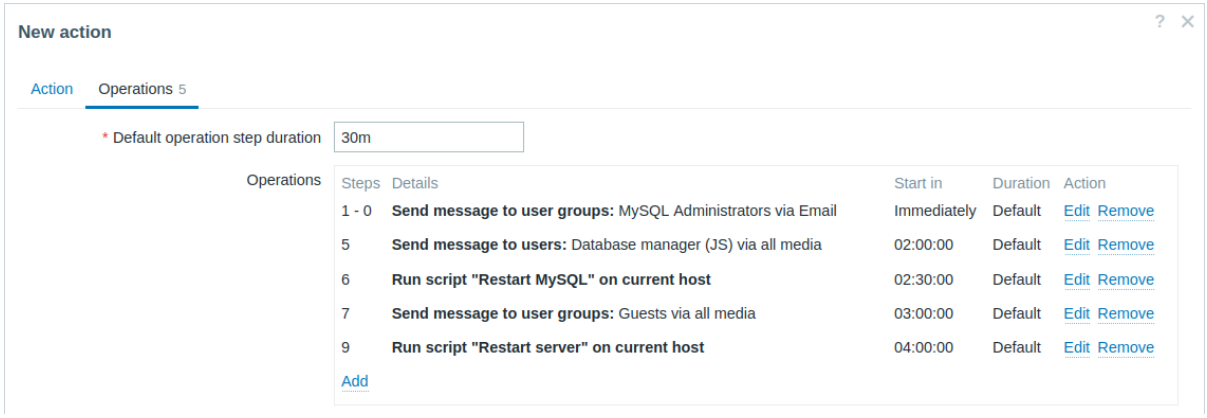
Обратите внимание на использование макроса {ESC.HISTORY} в пользовательском сообщении. Макрос будет содержать информацию обо всех ранее выполненных шагах этой эскалации, таких как отправленные уведомления и выполненные команды.

Пример 4

Более сложный сценарий. После нескольких сообщений Администраторам MySQL и эскалации менеджеру, Zabbix попытается перезапустить базу данных MySQL. Это произойдет, если проблема присутствует еще 2:30 часов и еще не была подтверждена.

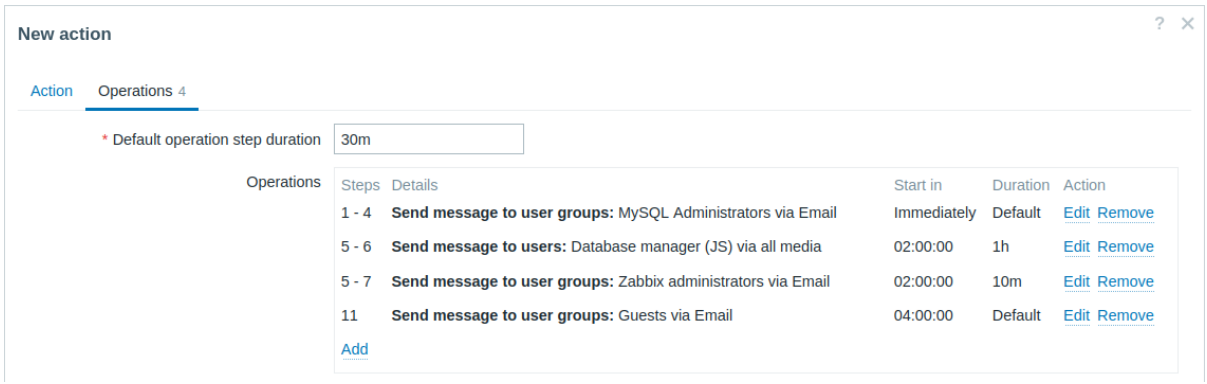
Если проблема все еще существует, спустя еще 30 минут Zabbix отправит сообщение всем гостевым пользователям.

Если и это не поможет, спустя еще час Zabbix перезагрузит сервер с этой базой данных MySQL (вторая удаленная команда), используя IPMI команды.



Пример 5

Эскалация с несколькими операциями, у которых диапазоны шагов перекрываются и заданы пользовательские интервалы. Длительность шага операции по умолчанию составляет 30 минут.



Уведомления будут отправлены следующим образом:

- Администраторам MySQL через 0:00, 0:30, 1:00 и 1:30 после начала проблемы.
- Менеджеру базы данных через 2:00 и 2:10 (более короткая пользовательская длительность шага в 10 минут, заданная в последующей операции, переопределяет более длительную длительность шага в 1 час, настроенную здесь, как описано в разделе Сведения об операции для *Длительность шага* при перекрытии шагов).
- Администраторам Zabbix через 2:00, 2:10 и 2:20 после начала проблемы (применяется пользовательская длительность шага 10 минут).
- Гостевым пользователям через 4:00 после начала проблемы (между шагами 8 и 11 действует длительность шага по умолчанию 30 минут).

3 Получение оповещений 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 состояние «unsupported» (и обратно);
- когда триггеры переходят из состояния «normal» в состояние «unknown» (и обратно);
- когда правила низкоуровневого обнаружения переходят из состояния «normal» в состояние «unsupported» (и обратно).

В этом разделе показано, как **получать уведомления**, когда элемент данных становится неподдерживаемым.

Настройка

В целом, процесс настройки оповещений должен быть знаком тем, кто раньше настраивал оповещения в Zabbix.

Шаг 1

Настройте некоторый способ оповещения, например email, SMS или скрипт, который будет использоваться для уведомлений. Чтобы выполнить эту задачу, обратитесь к соответствующим разделам руководства.

Attention:
Для уведомлений о внутренних событиях используется уровень важности по умолчанию (’Not classified’), поэтому оставьте его отмеченным при настройке способов оповещения пользователя, если вы хотите получать уведомления о внутренних событиях.

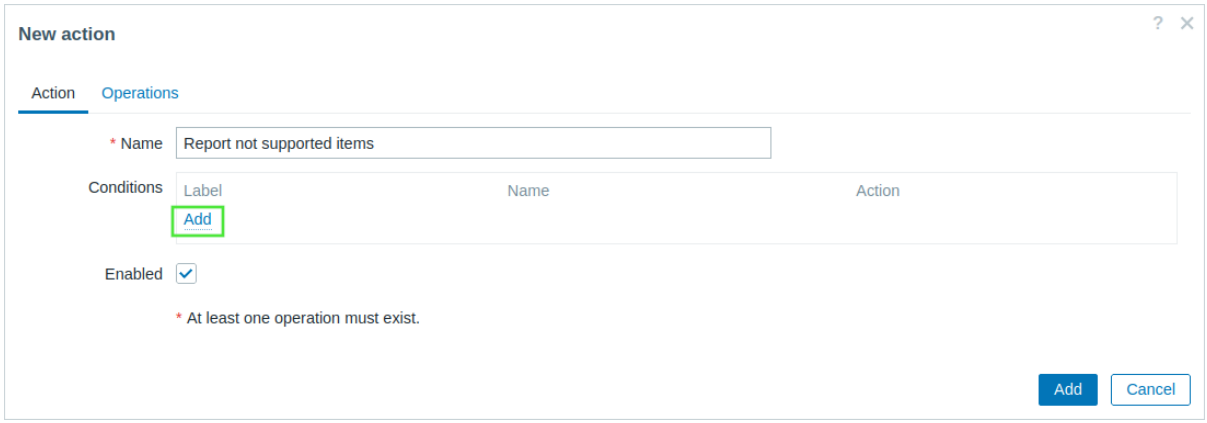
Шаг 2

Перейдите в *Оповещения* → *Действия* → *Внутренние действия*.

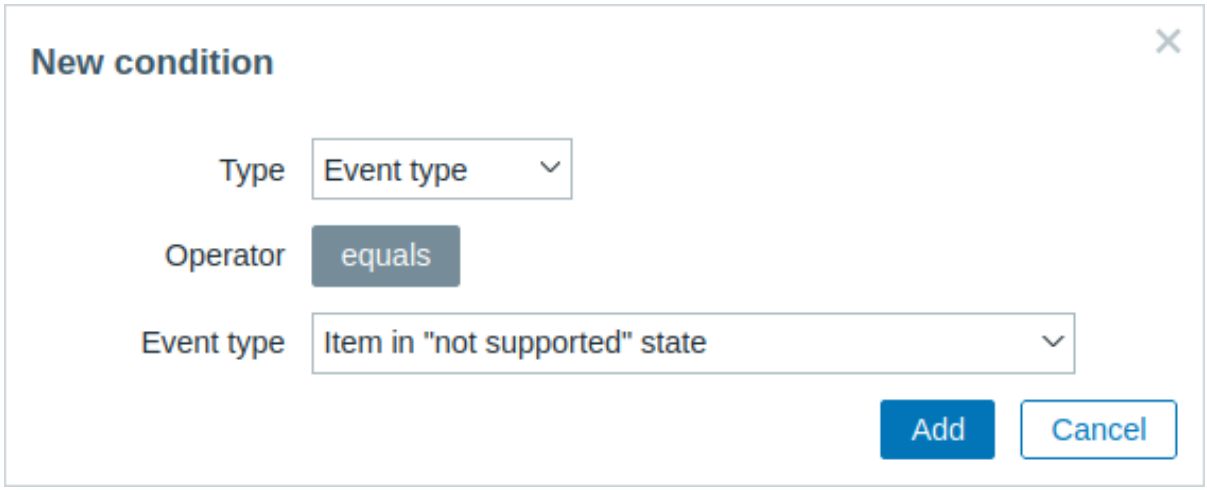
Нажмите *Создать действие* в правом верхнем углу страницы, чтобы открыть форму настройки действия.

Шаг 3

На вкладке *Действие* введите имя для действия. Затем нажмите *Добавить* в блоке *Условия*, чтобы добавить новое условие.



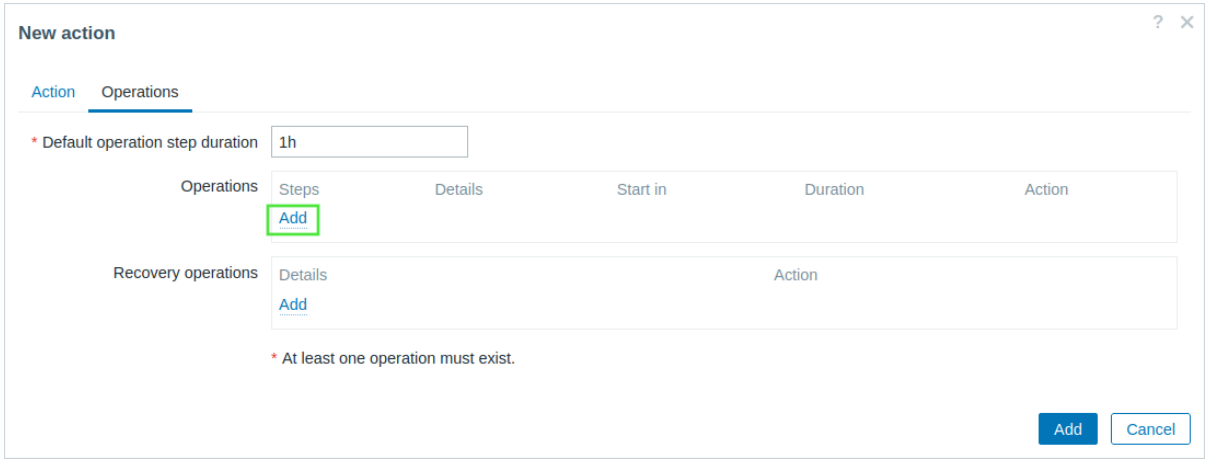
Во всплывающем окне *Новое условие* выберите «Тип события» в качестве типа условия, а затем выберите «Элемент данных в состоянии ”не поддерживается”» в качестве типа события.



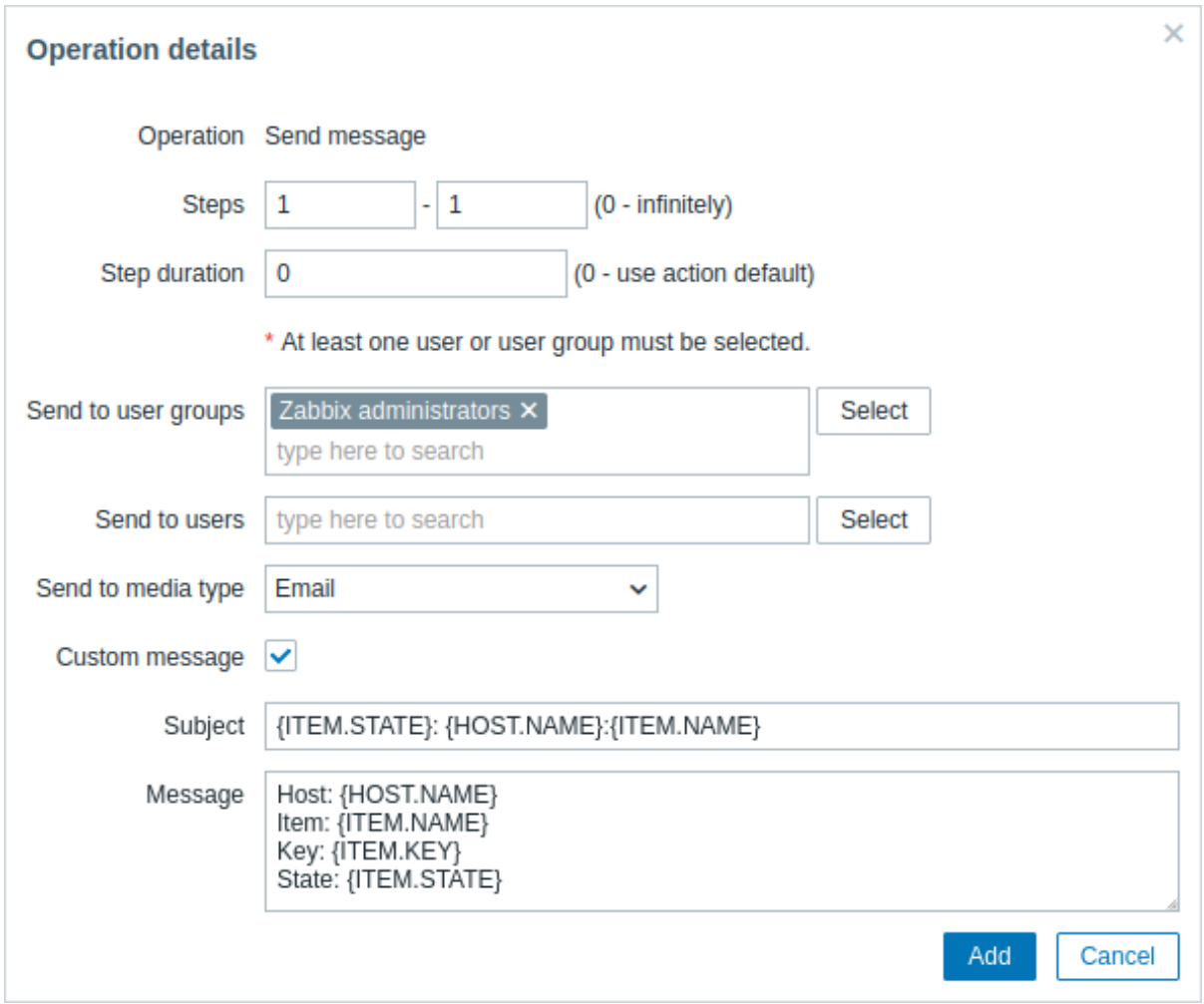
Не забудьте нажать *Добавить*, чтобы условие действительно появилось в блоке *Условия*.

Шаг 4

На вкладке *Операции* нажмите *Добавить* в блоке *Операции*, чтобы добавить новую операцию.



Выберите получателей сообщения (группы пользователей/пользователей) и тип медиа (или "Все"), который будет использоваться для доставки. Установите флажок *Пользовательское сообщение*, если хотите ввести пользовательскую тему/содержимое сообщения о проблеме.

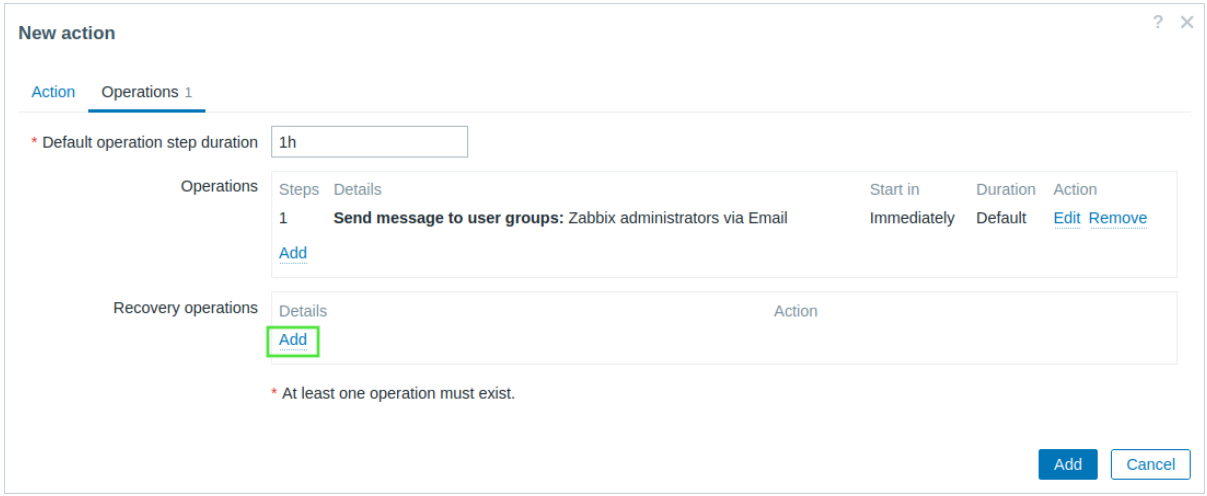


Нажмите *Добавить*, чтобы операция появилась в списке в блоке *Операции*.

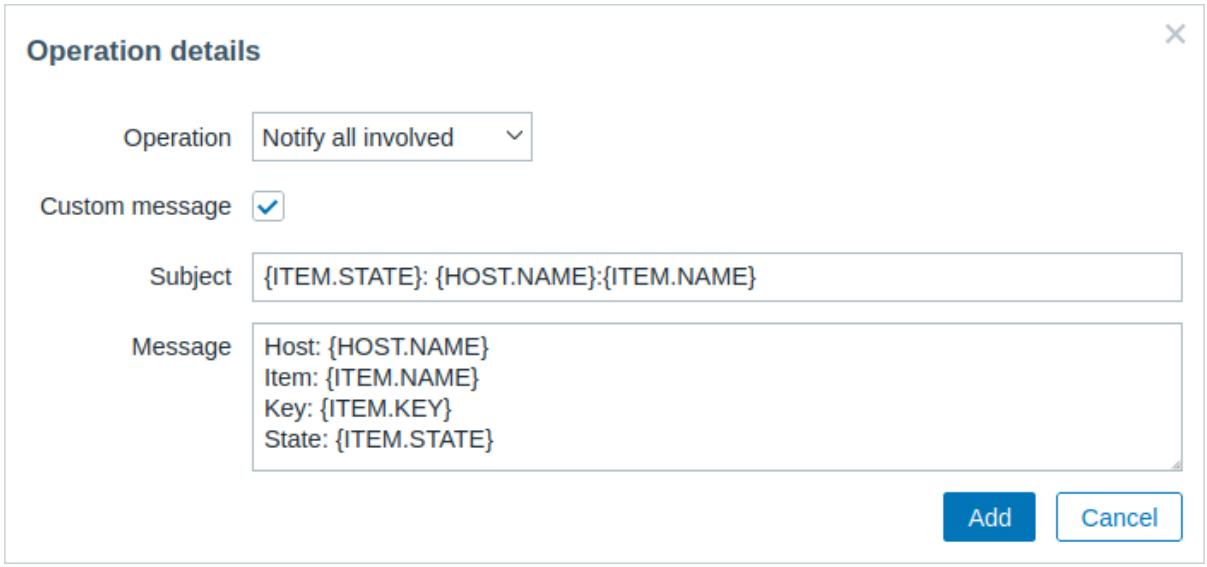
Если вы хотите получать более одного уведомления, задайте длительность шага операции (интервал между отправляемыми сообщениями) и добавьте еще один шаг.

Шаг 5

Блок *Операции восстановления* позволяет настроить уведомление о восстановлении, когда элемент данных возвращается в нормальное состояние. Нажмите *Добавить* в блоке *Операции восстановления*, чтобы добавить новую операцию восстановления.



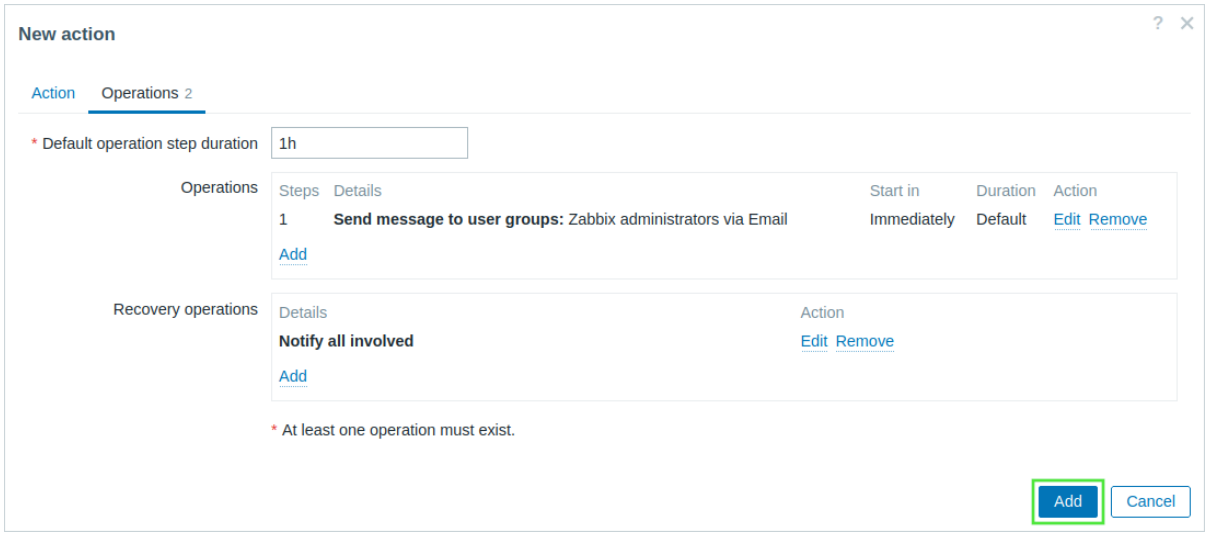
Выберите тип операции «Уведомить всех вовлечённых». Установите флажок *Пользовательское сообщение*, если хотите ввести пользовательскую тему/содержимое сообщения о восстановлении.



Нажмите *Добавить* во всплывающем окне *Сведения об операции*, чтобы операция появилась в блоке *Операции восстановления*.

Шаг 6

Когда завершите, нажмите на кнопку **Добавить** снизу диалога.



Итак, готово! Теперь вы можете ожидать получения вашего первого оповещения от Zabbix, если какой-либо элемент данных станет неподдерживаемым.

11 Макросы

Обзор

Zabbix поддерживает некоторое количество макросов, которые можно использовать в различных ситуациях. Макросами являются переменные, которые имеют специальный синтаксис:

{ }

Макросы раскрываются в требуемое значение в зависимости от контекста.

Эффективное использование макросов позволяет сохранить время и делает настройку Zabbix более прозрачной.

Самое типичное место использования макросов в шаблонах. Таким образом, триггер из шаблона можно назвать как "Слишком высокая загрузка процессора на {HOST.NAME}". Когда шаблон применяется к узлу сети, такому как Zabbix сервер, имя будет раскрыто в "Слишком высокая загрузка процессора на Zabbix сервер", когда триггер отобразится в разделе Мониторинга.

Макросы можно использовать в параметрах ключей элементов данных. Макрос можно использовать только как часть параметра, например, `item.key[server_{HOST.HOST}_local]`. Двойные кавычки не являются обязательными, так как Zabbix позаботится о неоднозначных спецсимволах, если они присутствуют в раскрытом макросе.

В Zabbix также имеются и другие типы макросов.

Zabbix поддерживает следующие макросы:

- { } - встроенные макросы (смотрите полный список)
- {< >.< >(< >)} - функции макросов
- {$ } - пользовательские макросы, опционально с контекстом
- {# } - макросы для низкоуровневого обнаружения
- {? } - макросы выражений

1 Функции макросов

Обзор

Функции макросов позволяют настраивать значения макросов (например, сокращать их или извлекать определённые подстроки), делая работу с ними более удобной.

Синтаксис функции макроса:

```
{macro.func(params)}
```

где

- **macro** - макрос для настройки;
- **func** - применяемая функция (см. поддерживаемые функции);
- **params** - список параметров функции, разделённых запятыми; они должны быть **заключены в двойные кавычки**, если:
  - начинаются с пробела или двойных кавычек;
  - содержат закрывающую круглую 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 применимо, если явно не указано, что ожидается только макрос (например, при настройке макросов узла сети или фильтров правила низкоуровневого обнаружения).

Для одного макроса поддерживается только одна функция; использование нескольких функций макросов в цепочке не поддерживается.

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 на функцию, чтобы увидеть полные сведения.

Function	Description
btoa	Кодирование значения макроса в кодировку Base64.
fmtnum	Форматирование числа для управления количеством цифр, выводимых 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).
lowercase	Преобразование символов значения макроса в нижний регистр.
regrepl	Замена символа/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.

Подробности функции

Необязательные параметры функции обозначаются с помощью < >.

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(digits)

Форматирование числа для управления количеством цифр, выводимых после десятичной точки.

Параметры:

- **digits** — количество цифр после десятичной точки. Допустимый диапазон: 0-20. Конечные нули не выводятся.

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 разрешаются в значение в одном из следующих форматов времени:

- `hh:mm:ss`
- `yyyy-mm-ddThh:mm:ss[tz]` (стандарт ISO8601)
- метка времени UNIX

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, которое 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 — 1-й день недели (понедельник), /M — 1-й 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:

```
{{TIME}.fmttime(%B)} -      "October"                "1633098961"
{{TIME}.fmttime(%d %B,-1M/M)} -      "1 September"                "1633098961"
```

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(pattern,output)

Извлечение подстроки по совпадению с регулярным выражением (без учета регистра).

Параметры:

- **pattern** — регулярное выражение для сопоставления;
- **output** — параметры вывода. Поддерживаются заполнители **\1 - \9** для групп захвата. **\0** возвращает совпавший текст.

Комментарии:

- Если совпадение с регулярным выражением отсутствует, функция возвращает пустую строку.
- Если шаблон функции является некорректным регулярным выражением, то макрос вычисляется как 'UNKNOWN' (кроме макросов низкоуровневого обнаружения, в этом случае функция будет проигнорирована, а макрос останется неразрешенным).
- Ссылки на несуществующие группы захвата в строке замены заменяются пустой строкой.

Пример:

```
{{ITEM.VALUE}.iregsub("fail|error|fault|problem","ERROR")} -              "ERROR",              "fail", "
```

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. Заполнители **\1 - \9** поддерживаются в строках замены для групп захвата.

Комментарии:

- 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(pattern,output)

Извлечение подстроки по совпадению с регулярным выражением (с учетом регистра).

Параметры:

- **pattern** - регулярное выражение для сопоставления;
- **output** - параметры вывода. Поддерживаются заполнители **\1 - \9** для групп захвата. **\0** возвращает совпавший текст.

Комментарии:

- Если для регулярного выражения нет совпадения, функция возвращает пустую строку.
- Если шаблон функции является некорректным регулярным выражением, то макрос вычисляется как 'UNKNOWN' (кроме макросов низкоуровневого обнаружения, в этом случае функция будет проигнорирована, а макрос останется неразрешенным).
- Ссылки на несуществующие группы захвата в строке замены заменяются пустой строкой.

Примеры:

```
{{ITEM.VALUE}.regsub("^([0-9]+)", Problem ID: \1)} -              "Problem ID: 123",                "123
{{ITEM.VALUE}.regsub("fail|error|fault|problem","ERROR")} -              "ERROR",              "fail", "e
```

См. больше примеров.

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"
```

Дополнительные примеры

В таблице ниже показаны дополнительные примеры использования функций макросов.

{#IFALIAS} — это LLD-макрос и он определяется только в контекстах низкоуровневого обнаружения (правила обнаружения, прототипы и созданные на их основе элементы данных/триггеры).
Использование его вне LLD оставит токен 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_11	

Функция макроса	Полученное значение	Вывод
{{#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 разрешённых макросов {ITEM.VALUE} и {ITEM.LASTVALUE} для текстовых/log элементов данных в некоторых местах веб-интерфейса усекаются до 20 символов. Чтобы увидеть полные значения этих макросов, можно использовать функции макросов, например:

```
{{ITEM.VALUE}.regsub("(.*)", \1)}
{{ITEM.LASTVALUE}.regsub("(.*)", \1)}
```

См. также: подробности о макросах {ITEM.VALUE} и {ITEM.LASTVALUE}.

2 Пользовательские макросы

Обзор

Пользовательские макросы поддерживаются в Zabbix для большей гибкости, в дополнение к макросам, поддерживаемым по умолчанию.

Пользовательские макросы могут быть определены на глобальном уровне, на уровне шаблона и узла сети. Эти макросы имеют специальный синтаксис:

{$MACRO}

Zabbix разрешает макросы в соответствии со следующим приоритетом:

1. макросы уровня узла сети (проверяются первыми)
2. макросы, определённые для шаблонов первого уровня узла сети (то есть шаблонов, связанных непосредственно с узлом сети), отсортированные по ID шаблона
3. макросы, определённые для шаблонов второго уровня узла сети, отсортированные по ID шаблона
4. макросы, определённые для шаблонов третьего уровня узла сети, отсортированные по ID шаблона и т. д.
5. глобальные макросы (проверяются последними)

Иными словами, если макрос не существует для узла сети, Zabbix попытается найти его в шаблонах узла сети с увеличением глубины. Если он всё равно не будет 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. Таким образом, наличие макросов с одинаковым именем в нескольких шаблонах является риском конфигурации.

Если Zabbix не сможет найти макрос, макрос не будет разрешён.

Attention:
Макросы (включая пользовательские макросы) намеренно оставляются неразрешёнными в разделе Configuration (например, в списке триггеров), чтобы сделать сложную конфигурацию более наглядной.

Пользовательские макросы могут использоваться в:

- имени элемента данных
- параметре ключа элемента данных
- интервалах обновления элемента данных и гибких интервалах
- имени и описании триггера
- параметрах выражения триггера и константах (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.

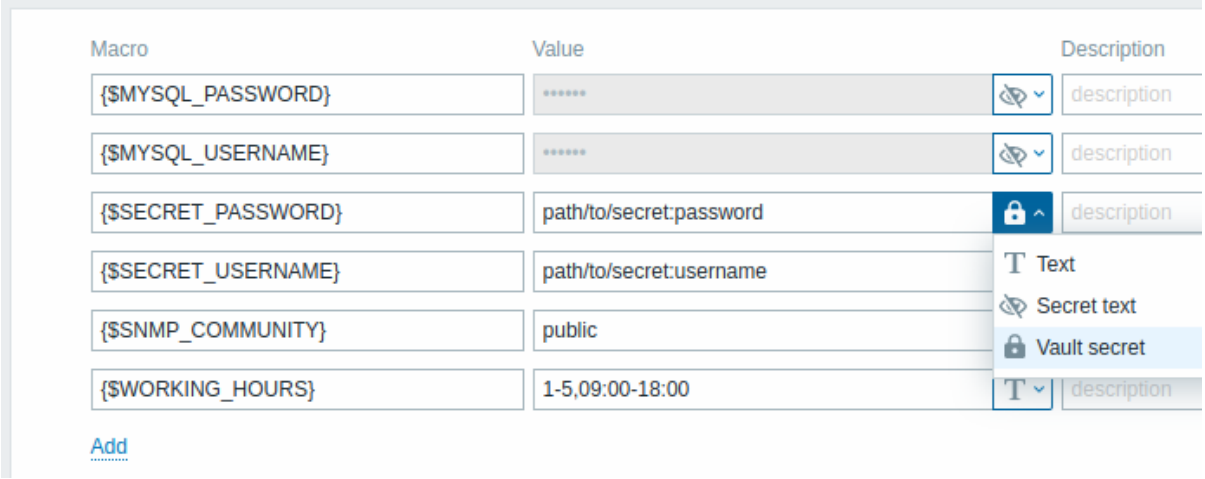
Note:
Рекомендуется использовать макросы узла сети вместо глобальных макросов, поскольку добавление, обновление или удаление глобальных макросов приводит к инкрементальному обновлению конфигурации для всех узлов сети. Дополнительную информацию смотрите в разделе Пассивные и активные проверки агента.

Настройка

Чтобы определить пользовательские макросы, перейдите в соответствующий раздел веб-интерфейса:

- для глобальных макросов откройте *Administration* → *Macros*
- для макросов уровня узла сети и шаблона откройте свойства узла сети или шаблона и найдите вкладку *Macros*

Пользовательский макрос имеет следующие атрибуты:



Параметр	Описание
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 верхний регистр), **0-9**, _, **.**
Value	Значение макроса. Поддерживаются три типа значений: **Text** (по умолчанию) — текстовое значение **Secret text** — значение маскируется 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 поведения в зависимости от контекста:

- если пользовательские макросы используются в элементах данных или триггерах шаблона, рекомендуется также добавить эти макросы 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, эти макросы будут разрешаться только при ссылке на параметр или константу; они НЕ будут разрешаться 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, у которого есть правило low-level discovery 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 присоединить к нескольким узлам сети, при условии, что значение **{$SSH_PORT}** указано на этих узлах сети.

Пример 2

Использование макроса уровня узла сети в триггере "Загрузка CPU слишком высокая":

'last(/ca_001/system.cpu.load[,avg1])>{$MAX_CPULOAD}

Такой триггер можно создать у шаблона, он не будет изменен у отдельных узлов сети.

Note:
Если вы хотите использовать количество значений параметром функции (например, **max(/узел сети/ключ,#3)**), добавьте символ решетки # непосредственно в значение макроса, например вот так: SOME_PERIOD ⇒ #3

Пример 3

Использование двух макросов в триггере "Загрузка CPU слишком высокая":

```
min(/ca_001/system.cpu.load[,avg1],{$CPULOAD_PERIOD})>{$MAX_CPULOAD}
```

Заметьте, что макрос можно использовать параметром функции триггеров, в этом примере в функции **min()**.

Пример 4

Синхронизация условия недоступности агентов с интервалом обновления элемента данных:

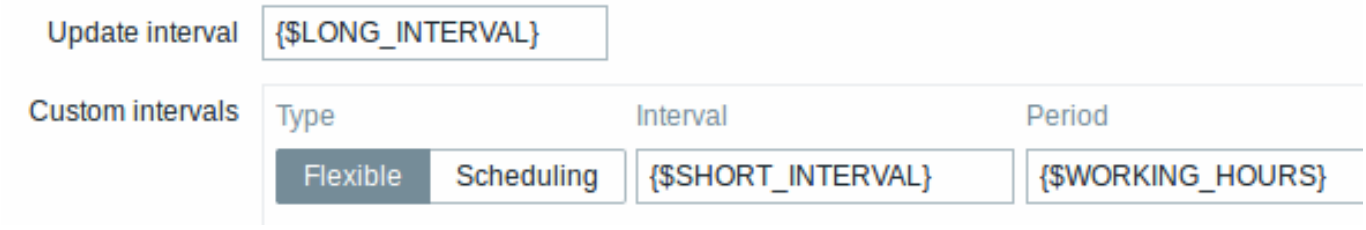
- задайте макрос {$INTERVAL} и используйте его в интервале обновления элемента данных;
- используйте {$INTERVAL} параметром в триггере на недоступность агента:

```
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 *Working time* в *Administration → General → GUI*;
- используйте его в поле *When active* в *Users → Users*, на вкладке *Media* пользователя;
- используйте его для настройки более частого опроса элементов данных в рабочие часы:

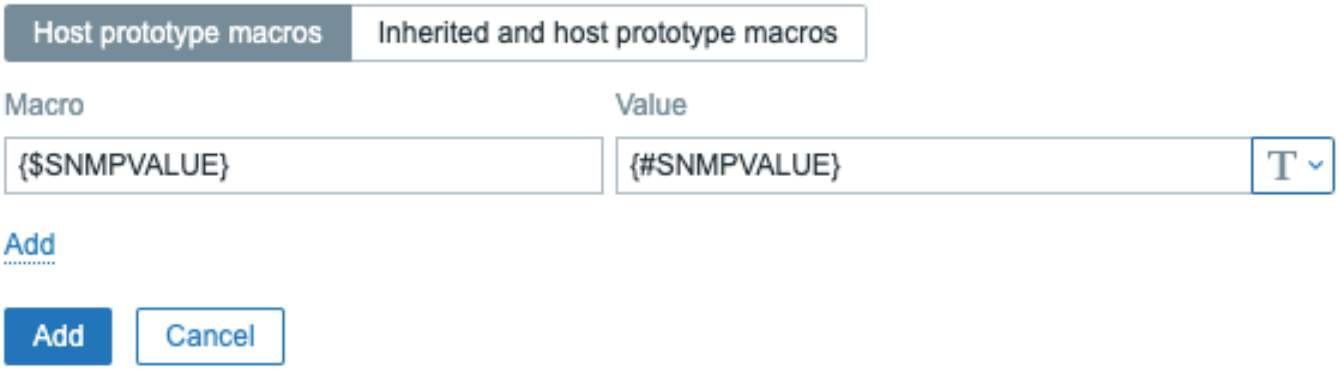


- используйте его в условии действия *Time period*;
- при необходимости скорректируйте рабочее время в *Administration → Macros*.

Пример 6

Используйте макрос прототипа узлов сети для настройки элементов данных у обнаруженных узлов сети:

- на уровне прототипа узлов сети задайте пользовательский макрос {$SNMPVALUE} с значением в виде {#SNMP-VALUE} макроса низкоуровневого обнаружения:



- назначьте *Generic SNMPv2* шаблон на прототип узлов сети;
- используйте {$SNMPVALUE} в *SNMP OID* поле элементов данных *Generic SNMPv2* шаблона.

Контекст пользовательских макросов

Смотрите пользовательские макросы с контекстом.

3 Пользовательские макросы с контекстом

Обзор

В пользовательских макросах можно использовать опциональный контекст, который позволяет переопределять значение по умолчанию значением с учётом конкретной ситуации.

Контекст добавляется к имени макроса; синтаксис зависит от того является ли контекст обычным текстовым значением:

```
{$    :"        "}
```

или регулярным выражением:

```
{$    :regex:"            "}
```

Обратите внимание, макрос с контекстом регулярного выражения можно задавать только в настройках пользовательских макросов. Если `regex:` префикс используется в каком-либо другом месте как контекст пользовательского макроса, например в выражении триггера, этот контекст будет рассматриваться как обычный контекст.

Заключение в кавычки контекста опционально (смотрите также важные заметки).

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 с префиксом /var/log/.

Варианты использования

Пользовательские макросы с контекстом можно определить для реализации более гибких порогов в выражениях триггеров (на основе значений, полученных с помощью обнаружения низкого уровня). Например, вы можете определить следующие макросы:

- {$LOW_SPACE_LIMIT} = 10
- {$LOW_SPACE_LIMIT:/home} = 20
- {$LOW_SPACE_LIMIT:regex:”^/[a-z]+$”} = 30

Затем макрос обнаружения низкого уровня можно использовать 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 нехватки свободного места в зависимости от обнаруженных точек монтирования или типов файловых систем. События о проблемах будут создаваться, если:

- в каталоге /home свободное место на диске составляет менее 20%
- в каталогах, соответствующих шаблону регулярного выражения (например, /etc, /tmp или /var), свободное место на диске составляет менее 30%
- в каталогах, которые не соответствуют шаблону регулярного выражения и не являются /home, свободное место на диске составляет менее 10%

Важные заметки

- Если существует более одного пользовательского макроса с контекстом, Zabbix сначала попытается сопоставить макросы с простым контекстом и только затем макросы с контекстом регулярного выражения в неопределенном порядке.

Warning:
Не создавайте разные контексты макросов, которые соответствуют одной строке, чтобы избежать непредвиденного поведения.

- Если макрос со своим контекстом не найден на узле сети, присоединенных шаблонах или на глобальном уровне, тогда будет произведен поиск макроса без наличия контекста.
- В контекстах макроса поддерживаются только макросы низкоуровневого обнаружения. Любые другие макросы игнорируются и обрабатываются как текстовые значения.

Технически, контекст макроса задается 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 заключенном в кавычки контексте необходимо экранировать при помощи \ символа.
- Сам символ \ не экранируется, что означает, что невозможно задать заключенный 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 кавычек в начале и после 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.

4 Секретные пользовательские макросы

Обзор

Zabbix предоставляет два варианта защиты конфиденциальной информации в значениях пользовательских макросов:

- Секретный текст
- Секрет хранилища

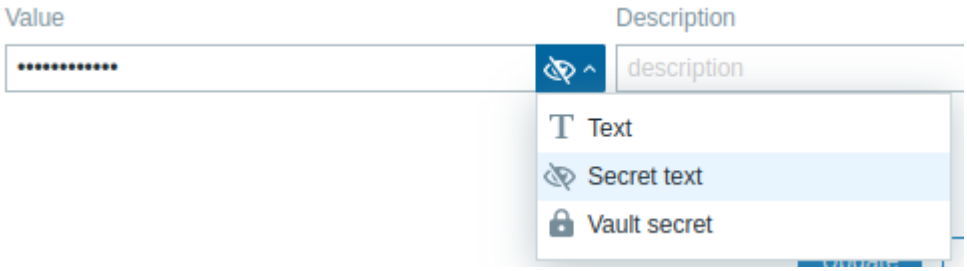
Note:
Хотя значение секретного макроса скрыто, его можно раскрыть при использовании в элементах данных. Например, во внешнем скрипте оператор echo, ссылающийся на секретный макрос, может быть использован для вывода значения макроса в веб-интерфейс, поскольку сервер Zabbix имеет доступ к реальному значению макроса. См. места, где значения секретных макросов отображаются без маскирования.

Секретные макросы нельзя использовать в выражениях триггеров.

Секретный текст

В макросах типа «Секретный текст» значение макроса маскируется звездочками.

Чтобы сделать значение макроса секретным, нажмите кнопку в конце поля *Значение* и выберите вариант *Секретный текст*:



После сохранения конфигурации просмотр значения станет невозможен.

Чтобы изменить значение макроса, наведите указатель мыши на поле *Значение* и нажмите кнопку *Установить новое значение* (появляется при наведении):



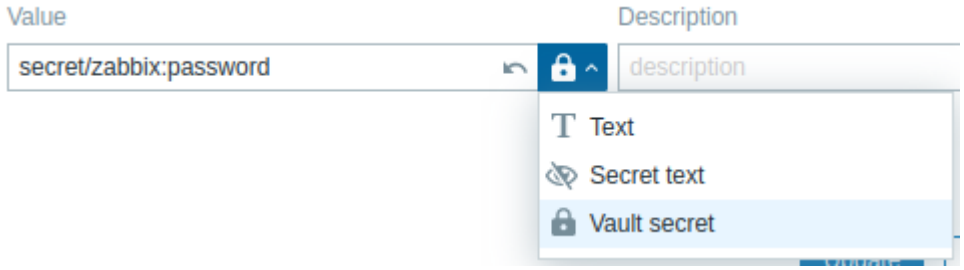
При нажатии кнопки *Установить новое значение* (или при изменении типа значения макроса) текущее значение будет удалено. Вы можете восстановить исходное значение, нажав стрелку в конце поля *Значение* (доступно только до сохранения новой конфигурации). Обратите внимание, что восстановление исходного значения не приведет к его отображению.

Note:
URL-адреса, содержащие секретный макрос, не будут работать, так как макрос в них будет разрешен как "******".

Секрет Vault

При использовании макросов секрета 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 вариант *Vault secret*:



Значение макроса должно указывать на секрет vault. Формат ввода зависит от поставщика vault. Примеры настройки для конкретных поставщиков см. здесь:

- HashiCorp
- CyberArk

Значения секретных макросов Vault извлекаются из хранилища сервером Zabbix (а также прокси Zabbix, если параметр *Resolve secret vault macros by* установлен в *Zabbix server and proxy*) при каждом обновлении данных конфигурации, а затем сохраняются в кэше конфигурации. Сервер Zabbix и прокси Zabbix могут использовать разные хранилища.

Если параметр *Resolve secret vault macros by* установлен в *Zabbix server*, то секреты хранилища извлекаются только сервером, а прокси Zabbix получает значения секретных макросов Vault от сервера Zabbix при каждой синхронизации конфигурации и сохраняет их в собственном кэше конфигурации. Это означает, что прокси Zabbix не сможет начать сбор данных после перезапуска, пока не получит обновление конфигурации от сервера Zabbix.

Чтобы вручную обновить секретные значения из хранилища, используйте параметр runtime control `secrets_reload` (только для сервера).

Между сервером Zabbix и прокси должно быть включено шифрование; в противном случае в журнал записывается предупреждающее сообщение сервера.

Warning:
Если значение макроса не удается успешно получить, соответствующий элемент данных, использующий это значение, станет неподдерживаемым.

Незамаскированные расположения

В этом списке приведены расположения параметров, в которых значения секретных макросов отображаются незамаскированными.

Note:
Значения секретных макросов останутся замаскированными в указанных ниже расположениях, если на них есть косвенная ссылка. Например, {ITEM.KEY}, {ITEM.KEY<1-9>}, {LLDRULE.KEY} встроенные макросы, используемые в типах оповещений (параметры Script или Webhook), будут преобразованы в ключи элементов данных, содержащие замаскированные секретные макросы, например `net.tcp.port[******,******]` вместо `net.tcp.port[192.0.2.0,80]`.

Контекст		Параметр
Элементы данных, прототипы элементов данных, правила LLD		
	Элемент данных	*Параметры ключа*
	Прототип элемента данных	*Параметры ключа*
	Правило низкоуровневого обнаружения	*Параметры ключа*
	SNMP агент	*SNMP community*
		Context name (SN
		Security name (SN
		Authentication pas
		Privacy passphrase
	HTTP агент	*URL*
		Поля запроса
		Тело запроса
		Заголовки
		Имя пользователя
		Пароль
		Пароль ключа SSL
	Script	*Параметры*
		Скрипт
	Browser	*Параметры*
		Скрипт
	Монитор базы данных	*SQL-запрос*
	TELNET агент	*Скрипт*
		Имя пользователя
		Пароль
	SSH агент	*Скрипт*
		Имя пользователя
		Пароль
	Простая проверка	*Имя пользователя*
		Пароль
	JMX агент	*Имя пользователя*
		Пароль
Предобработка значений элементов данных		
	Шаг предобработки JavaScript	*Скрипт*
Веб-сценарии		
	Веб-сценарий	*Значение переме*
		Значение заголов
		URL
		Значение поля за
		Значение поля PO
		Необработанные
	Аутентификация веб-сценария	*Пользователь*
		Пароль
		Пароль ключа SSL
Коннекторы		
	Коннектор	*URL*
		Имя пользователя
		Пароль
		Токен
		HTTP прокси
		Файл сертификат

Контекст		Параметр
		Файл ключа SSL
		Пароль ключа SSL
Обнаружение сети		
	SNMP	*SNMP community*
		Context name (SN
		Security name (SN
		Authentication pas
		Privacy passphrase
Глобальные скрипты		
	Webhook	*JavaScript-скрипт*
		Значение параме
	Telnet	*Имя пользовател*
		Пароль
	SSH	*Имя пользовател*
		Пароль
	Script	*Скрипт*
Типы оповещений		
	Script	*Параметры скрип*
	Webhook	*Параметры*
Управление IPMI		
	Узел сети	*Имя пользовател*
		Пароль

5 Макросы низкоуровневого обнаружения

Обзор

Существует тип макросов, используемый в функции обнаружения низкого уровня (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 могут использоваться 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 обнаружения низкого уровня поставляются "из коробки" с функцией LLD в Zabbix — {#FSNAME}, {#FSTYPE}, {#IFNAME}, {#SNMPINDEX}, {#SNMPVALUE}. Однако при создании пользовательского правила обнаружения низкого уровня придерживаться этих имен необязательно. В этом случае вы можете использовать любое другое имя макроса LLD и ссылаться на это имя.

Поддерживаемые типы данных

При определении пользовательских правил обнаружения значения свойств, возвращаемые в объектах JSON для макросов LLD, должны относиться к одному из следующих примитивных типов:

- string
- number
- boolean

Массивы, объекты и значения null не поддерживаются. Любой макрос LLD, ссылающийся на такое значение, останется неразвернутым и будет отображаться буквально (например, '{#MY_MACRO}') во время предобработки и создания элементов данных.

Поддерживаемые расположения

LLD-макросы можно использовать:

- в фильтре правила низкоуровневого обнаружения
- во вложенных правилах низкоуровневого обнаружения, в
  - параметрах предобработки 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 предобработки 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)
    - SSH-скрипте и Telnet-скрипте
    - SQL-запросе мониторинга базы данных
    - поле endpoint элемента данных JMX
    - описании
    - поле URL HTTP-агента
    - поле HTTP query fields HTTP-агента
    - поле тела запроса HTTP-агента
    - поле требуемых кодов состояния HTTP-агента
    - ключе и значении поля заголовков HTTP-агента
    - поле имени пользователя HTTP-аутентификации HTTP-агента
    - поле пароля HTTP-аутентификации HTTP-агента
    - поле HTTP-прокси HTTP-агента
    - поле файла SSL-сертификата HTTP-агента
    - поле файла SSL-ключа HTTP-агента
    - поле пароля SSL-ключа HTTP-агента
    - тегах
- для прототипов триггеров в
    - имени
    - рабочих 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, community SNMP v1/v2, контекстное имя SNMP v3, имя безопасности SNMP v3, парольная фраза аутентификации SNMP v3, парольная фраза приватности 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 пользовательского контекста макроса.

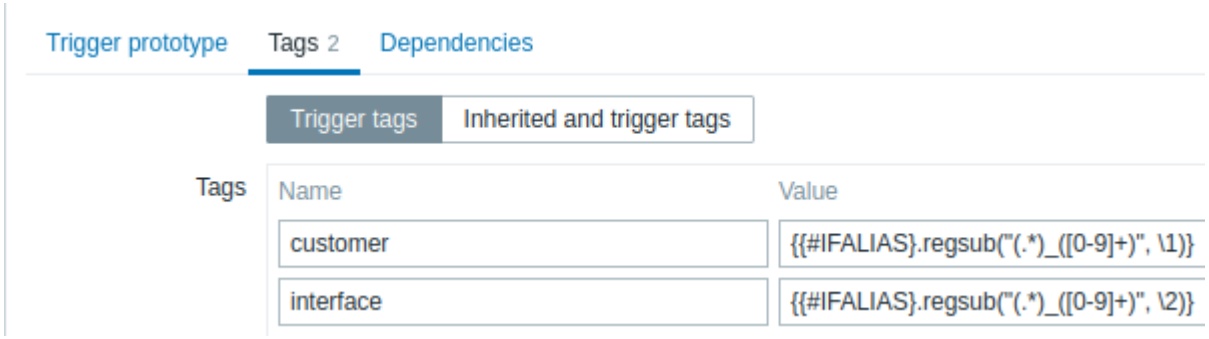
Использование функций макросов

Функции макросов поддерживаются с макросами низкоуровневого обнаружения, позволяя извлекать определённую часть значения макроса с помощью регулярного выражения (кроме фильтра правила низкоуровневого обнаружения).

Например, может потребоваться извлечь имя клиента и номер интерфейса из следующего LLD-макроса для целей присвоения тегов событиям:

```
{#IFALIAS}=customername_1
```

Для этого можно использовать функцию макроса `regsub` с макросом в поле значения тега события в прототипе триггера:



Обратите внимание, что запятые не допускаются в некавыченных параметрах ключа элемента данных, поэтому параметр, содержащий функцию макроса, должен быть заключён 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

Если параметры ключа элемента данных не используются, поместите макрос LLD внутрь скобок параметра ключа элемента данных [...], чтобы он обрабатывался как параметр и подставлялся во время обнаружения:

```
v_[{#MACRO}]
```

Сноски

[1] В полях, отмеченных [1], один макрос должен заполнять всё поле целиком. Несколько макросов в одном поле или макросы, смешанные с текстом, не поддерживаются.

6 Макросы выражений

Обзор

Макросы выражений позволяют выполнять вычисления в полях.

Их значение вычисляется путем сначала разрешения всех внутренних макросов, а затем вычисления получившегося 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.*} для ссылки на значения функций выражений триггеров/выражений восстановления.
- Используйте макросы {HOST.HOST<1-9>} и {ITEM.KEY<1-9>} для ссылки на узлы сети и элементы данных.
- В шаблонах используйте макросы {HOST.HOST<1-9>} или вообще опускайте узел сети для первого узла сети — например, {?avg(//item1,1h)} — вместо имен шаблонов, так как имена шаблонов не заменяются именами узлов сети во время привязки шаблонов.

```
{?{FUNCTION.VALUE2} - {FUNCTION.VALUE3}}
{?max(/{HOST.HOST}/{ITEM.KEY},3h)}
```

См. также Примеры выражений триггеров для примера использования макроса выражения в имени события.

12 Пользователи и группы пользователей

Обзор

Все пользователи в Zabbix работают с системой Zabbix через веб-интерфейс. Каждому пользователю в Zabbix присвоен уникальные имя пользователя и пароль.

Все пароли пользователей зашифрованы и хранятся в базе данных Zabbix. Пользователи Zabbix не могут использовать свои имя пользователя и пароль для авторизации непосредственно на сервере UNIX, за исключением, если они были заведены соответственно также и в UNIX. Соединение между Веб-сервером и браузером пользователя может быть защищено с помощью SSL.

При наличии гибкой схемы прав доступа пользователя вы можете ограничить и разграничить права доступа к:

- административным функциям веб-интерфейса Zabbix
- выполнению определенных действий в веб-интерфейсе
- доступу к наблюдаемым узлам сети в группах узлов сети
- использованию заданных API методов

1 Настройка пользователя

Обзор

В исходной установке Zabbix есть два предопределенных пользователя:

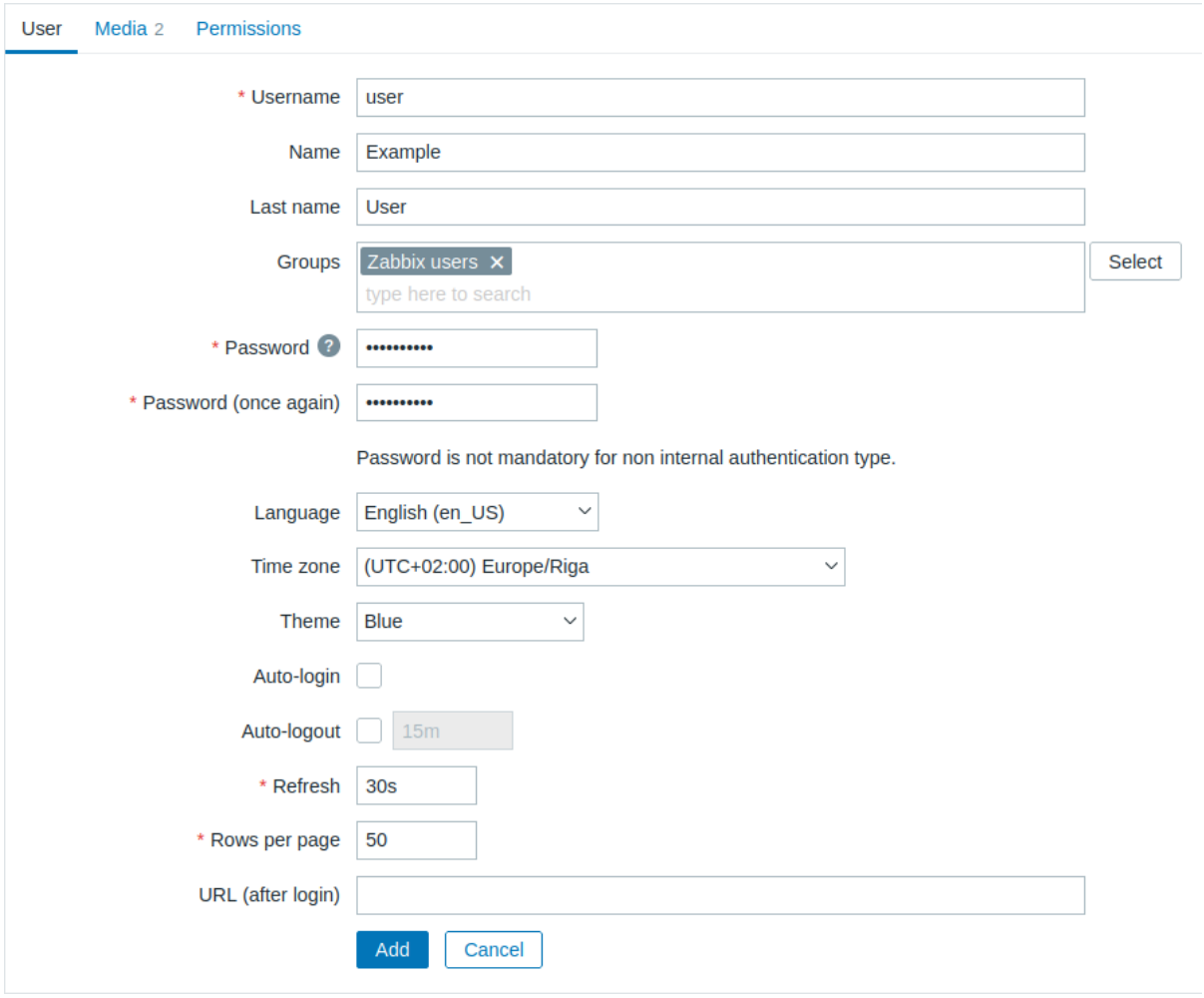
- *Admin* — Zabbix суперпользователь с полными правами.
- *guest* — специальный Zabbix пользователь. Пользователь 'guest' по умолчанию отключен. Если вы добавите его в группу пользователей Guests, вы сможете войти в систему под этим пользователем и получить доступ к страницам мониторинга в Zabbix. Обратите внимание, что по умолчанию у 'guest' нет прав на объекты Zabbix.

Чтобы настроить пользователя:

- Перейдите в *Пользователи* → *Пользователи*.
- Нажмите *Создать пользователя* (или на имя пользователя, чтобы изменить существующего пользователя).
- Измените атрибуты пользователя в форме.

Общие атрибуты

Вкладка *Пользователь* содержит общие атрибуты пользователя:



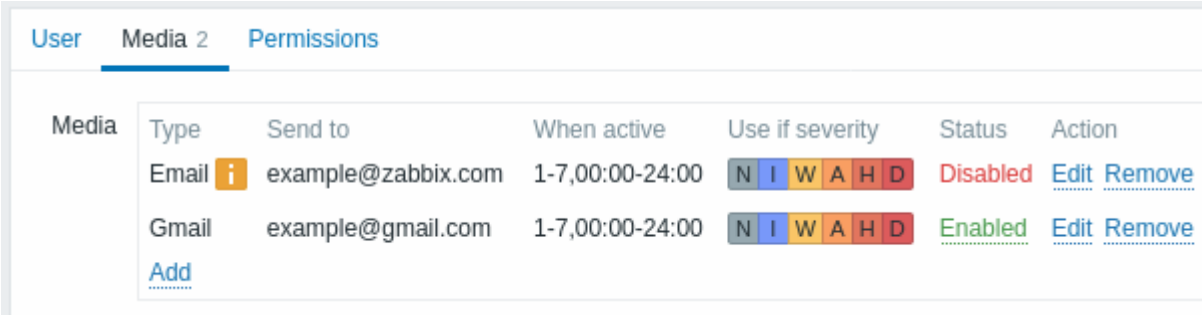
Все обязательные для заполнения поля отмечены красной звездочкой.

Параметр	Описание
Имя пользователя	Уникальное имя пользователя, используемое в качестве имени для входа.
Имя	Имя пользователя (необязательно). Если поле не пустое, отображается в информации о подтверждении и информации о получателе уведомления.
Фамилия	Фамилия пользователя (необязательно). Если поле не пустое, отображается в информации о подтверждении и информации о получателе уведомления.
Группы	Выберите группы пользователей, к которым принадлежит пользователь. Это поле поддерживает автодополнение, поэтому при вводе имени группы пользователей будет предложен выпадающий список подходящих групп. Прокрутите вниз для выбора. Либо нажмите *Выбрать*, чтобы добавить группы. Нажмите 'x', чтобы удалить выбранное. Принадлежность к группам пользователей определяет, к каким группам узлов сети и узлам сети пользователь будет иметь доступ.
Пароль	Два поля для ввода пароля пользователя или кнопка *Изменить пароль*, если пользователь уже существует. Нажатие кнопки *Изменить пароль* открывает два поля для ввода нового пароля. Для пользователя с ролью *Super admin role*, изменяющего собственный пароль, нажатие кнопки *Изменить пароль* открывает дополнительное поле для ввода текущего (старого) пароля. При успешной смене пароля пользователь, для которого был изменен пароль, будет выведен из 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 **По умолчанию для системы**, чтобы использовать глобальные настройки часового пояса.

Параметр	Описание
Тема	Определяет внешний вид веб-интерфейса: **По умолчанию для системы** — использовать системные настройки по умолчанию **Blue** — стандартная синяя тема **Dark** — альтернативная темная тема **High-contrast light** — светлая тема с высокой контрастностью **High-contrast dark** — темная тема с высокой контрастностью
Автовход	Установите этот флажок, чтобы Zabbix запомнил пользователя и автоматически выполнял вход в течение 30 дней. При входе с *Запомнить на 30 дней*: - Автовыход сбрасывается (сеанс сохраняется 30 дней). - Автовход включается для бесшовной повторной аутентификации. При входе без *Запомнить на 30 дней*: - Автовход включается для бесшовной повторной аутентификации. - Автовыход по-прежнему регулируется стандартной настройкой тайм-аута. Для этого используются cookie-файлы браузера.
Автовыход	Если этот флажок установлен, пользователь будет автоматически выходить из системы по истечении заданного количества секунд (минимум 90 секунд, максимум 1 день). Обратите внимание, что этот параметр игнорируется, если включена опция *Запомнить на 30 дней*, так как сеанс продлевается на весь этот период. Поддерживаются суффиксы времени, например 90s, 5m, 2h, 1d. Обратите внимание, что эта опция не будет работать: * Если включена глобальная опция конфигурации ”Show warning if Zabbix server is down” и веб-интерфейс Zabbix остается открытым. * Когда страницы меню Monitoring выполняют фоновое обновление информации. * При входе с установленной опцией *Remember me for 30 days*.
Обновление	Установите частоту обновления, используемую для графиков, текстовых данных и т. д. Можно установить 0, чтобы отключить. Поддерживаются суффиксы времени, например 90s, 5m, 1h.
Строк на странице	Вы можете определить, сколько строк на странице будет отображаться в списках.
URL (после входа)	Вы можете настроить перенаправление пользователя на определенный URL после успешного входа, например на страницу *Problems*.

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 имени отображается желтый значок информации.
- В столбце Status отображается *Отключено*.

Note:
Права пользователя на изменение собственных данных способов оповещения могут быть предоставлены/отозваны в зависимости от его роли пользователя (см. параметр *Create and edit own media*). Права пользователя Super admin на изменение данных способов оповещения других пользователей также могут быть предоставлены/отозваны в зависимости от его роли пользователя (см. параметр *Create and edit user media*)

Подробные сведения о настройке способов оповещения пользователя см. в разделе Типы способов оповещения.

Права доступа

Вкладка *Права доступа* содержит информацию о следующих элементах:

- Роль пользователя (обязательна для любого вновь созданного пользователя), которую может изменить только пользователь типа *Супер-администратор*.

Attention:
Пользователи не могут быть созданы без роли пользователя (кроме случаев создания через Zabbix User API). Ранее созданные пользователи, у которых нет роли, по-прежнему могут редактироваться без назначения им роли. Однако после назначения роли ее можно только изменить, но не удалить.

 Обратите внимание, что пользователи без роли могут входить в Zabbix только с использованием аутентификации LDAP или SAML, при условии, что их информация LDAP/SAML соответствует сопоставлениям групп пользователей, настроенным в Zabbix.

- Тип пользователя (*Пользователь*, *Администратор*, *Супер-администратор*), который определяется в настройках роли пользователя.
- Группы узлов сети и шаблонов, к которым у пользователя есть доступ.
  - Пользователи типа *Пользователь* и *Администратор* по умолчанию не имеют доступа ни к каким группам, шаблонам и узлам сети. Чтобы предоставить такой доступ, пользователи должны быть включены в группы пользователей, настроенные с правами доступа к соответствующим сущностям.
- Права доступа к разделам и элементам веб-интерфейса Zabbix, модулям и методам API.
  - Элементы с разрешенным доступом отображаются зеленым цветом, а элементы с запрещенным доступом — светло-серым.
- Права на выполнение определенных действий.
  - Действия, которые пользователю разрешено выполнять, отображаются зеленым цветом, а запрещенные — светло-серым.

Подробности см. на странице Права доступа.

2 Права доступа

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 зависят, прежде всего, от типа пользователя:

- *User* — имеет ограниченные права доступа к разделам меню (см. ниже) и по умолчанию не имеет доступа ни к каким ресурсам. Любые права на группы узлов сети или группы шаблонов должны быть назначены явно;
- *Admin* — имеет неполные права доступа к разделам меню (см. ниже). Пользователь по умолчанию не имеет доступа ни к каким группам узлов сети. Любые права на группы узлов сети или группы шаблонов должны быть назначены явно;
- *Super admin* — имеет доступ ко всем разделам меню. Пользователь имеет доступ на чтение и запись ко всем группам узлов сети и группам шаблонов. Права доступа не могут быть отозваны путем запрета доступа к определенным группам.

Доступ к меню

Note:
Ограничение доступа к некоторым элементам пользовательского интерфейса лишь предотвращает открытие этой страницы — оно не устраняет возможность доступа к базовым данным в других частях интерфейса.

Следующая таблица иллюстрирует доступ к разделам меню Zabbix в зависимости от типа пользователя:

Раздел меню		User	Admin	Super admin
Панели		+	+	+
Мониторинг		+	+	+
	Проблемы	+	+	+
	Узлы сети	+	+	+
	Последние данные	+	+	+
	Карты	+	+	+

Раздел меню		User	Admin	Super admin
	Обнаружение		+	+
Сервисы		+	+	+
	Сервисы	+	+	+
	SLA		+	+
	Отчет SLA	+	+	+
Инвентаризация		+	+	+
	Обзор	+	+	+
	Узлы сети	+	+	+
Отчеты		+	+	+
	Информация о системе			+
	Отчеты по расписанию		+	+
	Отчет о доступности	+	+	+
	Топ 100 триггеров	+	+	+
	Журнал аудита			+
	Журнал действий			+
	Уведомления		+	+
Сбор данных			+	+
	Группы шаблонов		+	+
	Группы узлов сети		+	+
	Шаблоны		+	+
	Узлы сети		+	+
	Обслуживание		+	+
	Корреляция событий			+
	Обнаружение		+	+
Оповещения			+	+
	Действия триггеров		+	+
	Действия сервисов		+	+
	Действия обнаружения		+	+
	Действия авторегистрации		+	+
	Внутренние действия		+	+
	Способы оповещения			+
	Скрипты			+
Пользователи				+
	Группы пользователей			+
	Роли пользователей			+
	Пользователи			+
	Токены API			+
	Аутентификация			+
Администрирование				+
	Общие			+
	Журнал аудита			+
	Очистка истории			+
	Группы прокси			+
	Прокси			+
	Макросы			+
	Очередь			+

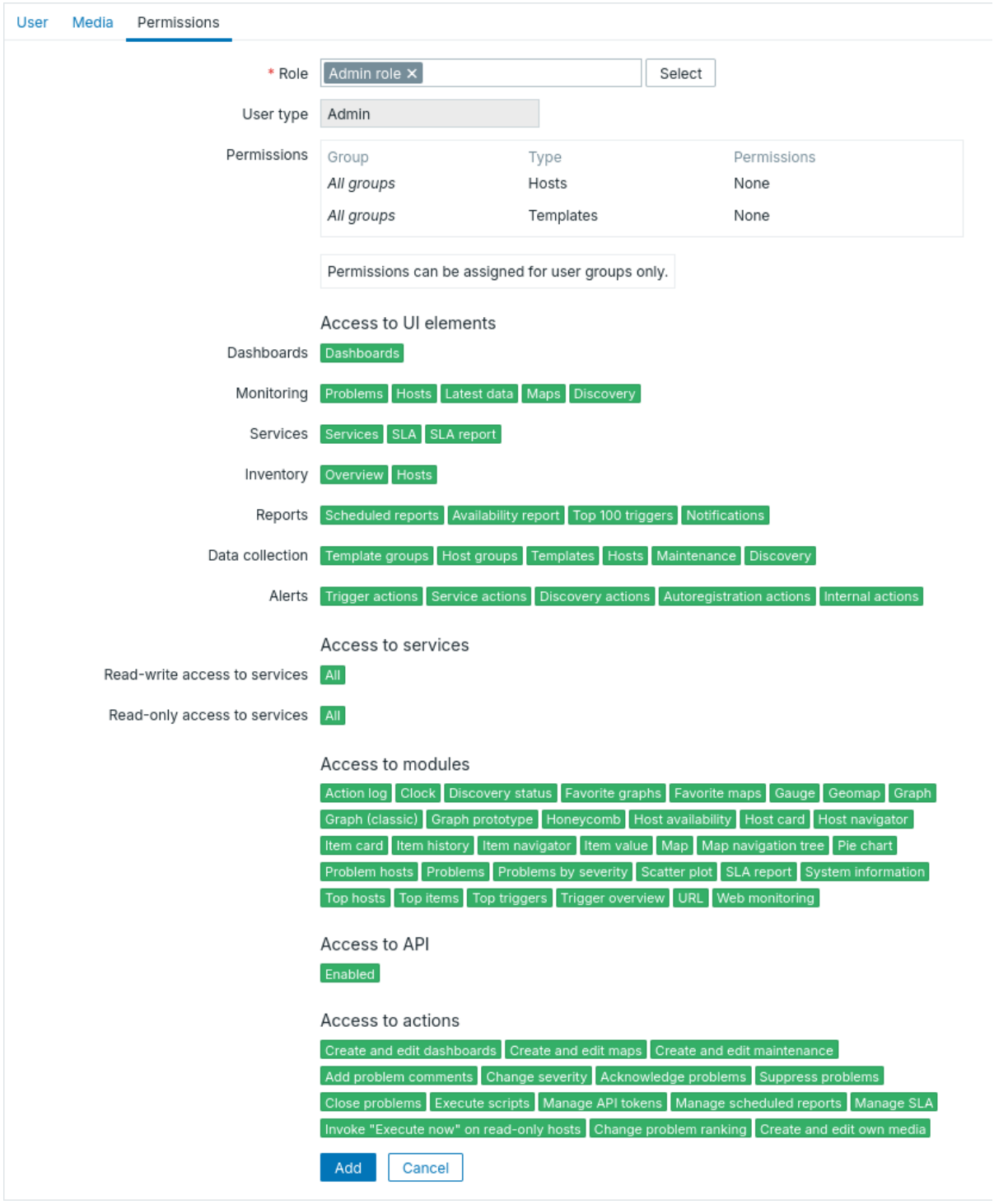
Роли пользователей

Роли пользователей позволяют выполнять пользовательскую настройку прав, определённых типом пользователя. Хотя никакие права не могут быть добавлены (так как это превысило бы права типа пользователя), некоторые права могут быть отозваны.

Кроме того, роль пользователя определяет доступ не только к разделам меню, но также к сервисам, модулям, методам API и различным действиям в веб-интерфейсе.

Роли пользователей настраиваются в разделе *Пользователи* → *Роли пользователей* пользователями с правами Super admin.

Роли пользователей назначаются пользователям в форме настройки пользователя, на вкладке *Права доступа*, пользователями с правами Super admin.



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.

3 Группы пользователей

Обзор

Группы пользователей позволяют объединять пользователей как в организационных целях, так и для назначения прав доступа к данным. Права на просмотр и настройку данных групп узлов сети и групп шаблонов назначаются группам

пользователей, а не отдельным пользователям.

Часто имеет смысл разделять, какая информация доступна одной группе пользователей, а какая — другой. Этого можно добиться, сгруппировав пользователей, а затем назначив различные права доступа группам узлов сети и шаблонов.

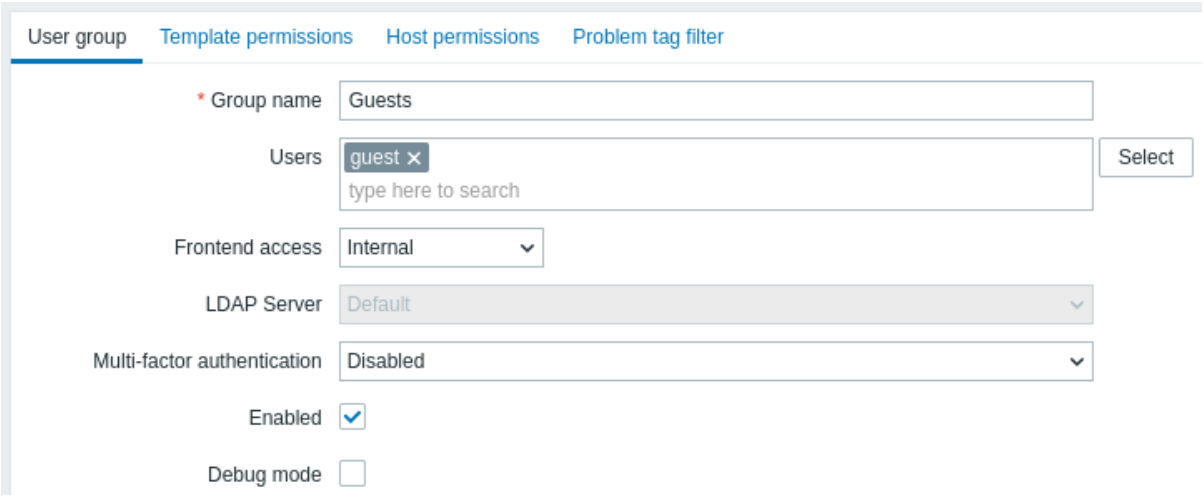
Пользователь может принадлежать к любому количеству групп.

Конфигурация

Чтобы настроить группу пользователей:

- Перейдите в *Пользователи* > *Группы пользователей*
- Нажмите *Создать группу пользователей* (или на имя группы, чтобы изменить существующую группу)
- Измените атрибуты группы в форме

Вкладка **Группа пользователей** содержит общие атрибуты группы:

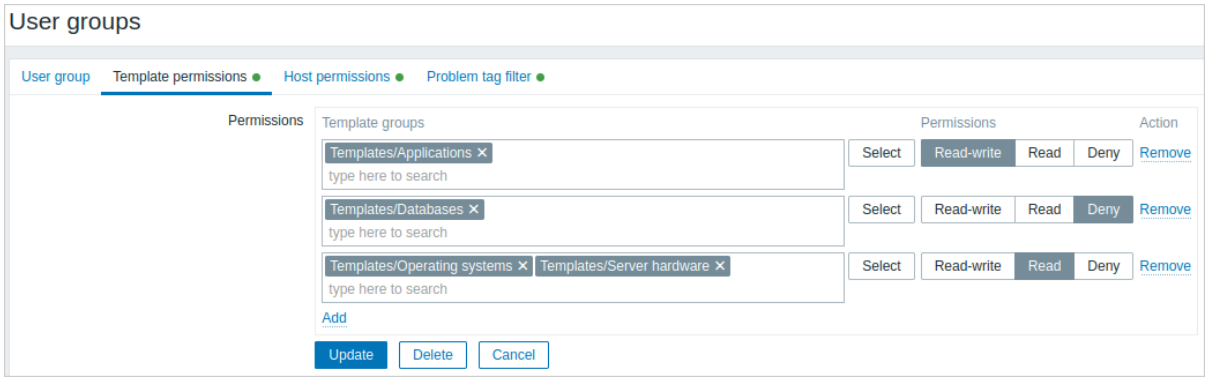


Все обязательные поля ввода отмечены красной звёздочкой.

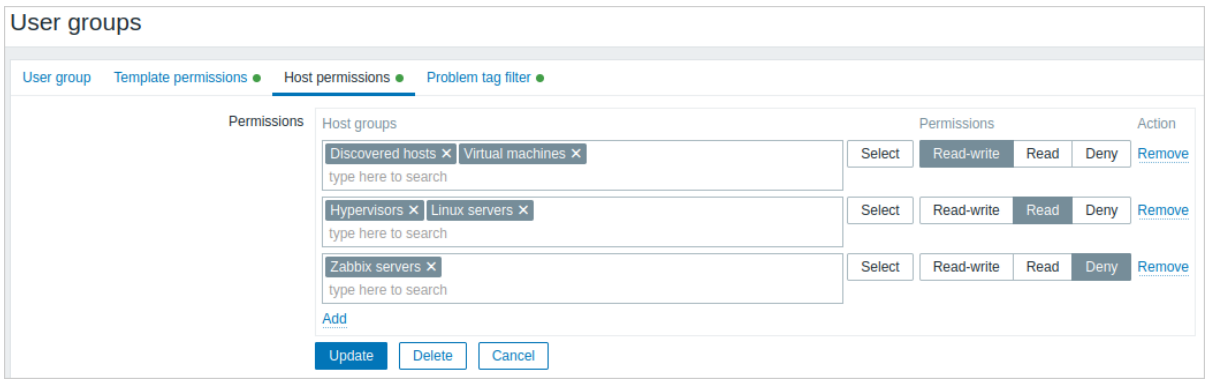
Параметр	Описание
Имя группы	Уникальное имя группы.
Пользователи	Чтобы добавить пользователей в группу, начните вводить имя существующего пользователя. Когда появится выпадающий список с подходящими именами пользователей, прокрутите его вниз и выберите нужное имя. Также можно нажать кнопку *Выбрать*, чтобы выбрать пользователей во всплывающем окне.
Доступ к веб-интерфейсу	Как выполняется аутентификация пользователей группы. **По умолчанию для системы** — использовать метод аутентификации по умолчанию (заданный глобально) **Внутренний** — использовать внутреннюю аутентификацию Zabbix (даже если глобально используется аутентификация LDAP). Игнорируется, если глобально по умолчанию используется HTTP-аутентификация. **LDAP** — использовать аутентификацию LDAP (даже если глобально используется внутренняя аутентификация). Игнорируется, если глобально по умолчанию используется HTTP-аутентификация. **Отключено** — доступ к веб-интерфейсу Zabbix для этой группы запрещён
LDAP сервер	Выберите, какой LDAP сервер использовать для аутентификации пользователя. Это поле доступно только если для *Доступ к веб-интерфейсу* установлено значение LDAP или По умолчанию для системы.

Параметр	Описание
Многофакторная аутентификация	Выберите, какой метод многофакторной аутентификации использовать для аутентификации пользователя: **По умолчанию** — использовать метод, заданный по умолчанию в конфигурации MFA; этот вариант по умолчанию выбран для новых групп пользователей, если MFA включена; **<Имя метода>** — использовать выбранный метод (например, ”Zabbix TOTP”); **Отключено** — MFA отключена для этой группы; этот вариант по умолчанию выбран для новых групп пользователей, если MFA отключена. Обратите внимание, что если пользователь входит в несколько групп пользователей с включённой 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 использоваться первый метод в алфавитном порядке.
Включено	Состояние группы пользователей и её участников. *Отмечено* — группа пользователей и пользователи включены *Не отмечено* — группа пользователей и пользователи отключены
Режим отладки	Установите этот флажок, чтобы активировать режим отладки для пользователей.

Вкладка **Права доступа к шаблонам** позволяет указать доступ группы пользователей к данным групп шаблонов (и, соответственно, шаблонов):



Вкладка **Права доступа к узлам сети** позволяет указать доступ группы пользователей к данным групп узлов сети (и, соответственно, узлов сети):



Нажмите Add, чтобы выбрать группы шаблонов/узлов сети (как родительские, так и вложенные) и назначить им права доступа. Начните вводить имя группы (появится выпадающий список подходящих групп) или нажмите *Выбрать*, чтобы открыть всплывающее окно со списком всех групп.

Затем используйте кнопки выбора, чтобы назначить права доступа выбранным группам. Возможны следующие права доступа:

- **Чтение-запись** — доступ к группе на чтение и запись
- **Чтение** — доступ к группе только на чтение
- **Запретить** — доступ к группе запрещён

Если одна и та же группа шаблонов/узлов сети добавлена в нескольких строках с разными правами доступа, будет применено самое строгое право.

Обратите внимание, что пользователь *Супер-администратор* может принудительно установить для вложенных групп тот же уровень прав доступа, что и для родительской группы; это можно сделать в форме настройки группы узлов

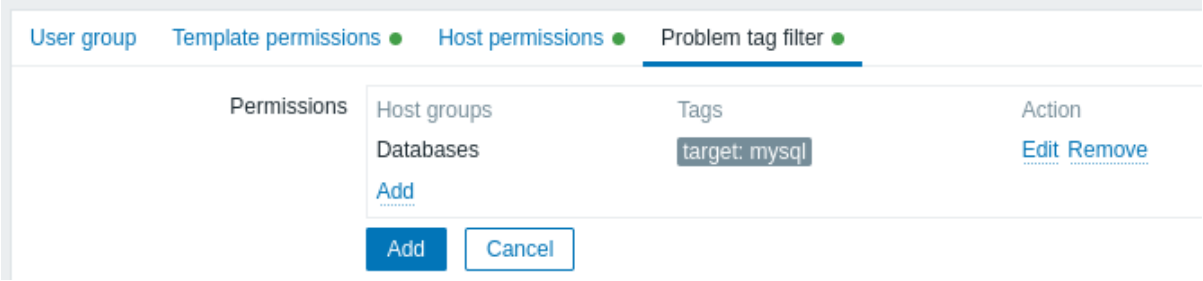
сети/шаблонов.

Вкладки **Права доступа к шаблонам** и **Права доступа к узлам сети** поддерживают одинаковый набор параметров.

Текущие права доступа к группам отображаются в блоке *Права доступа*; их можно изменить или удалить.

Note:
Если группа пользователей имеет права **Чтение-запись** на узел сети и права **Запретить** или не имеет прав на шаблон, связанный с этим узлом сети, пользователи такой группы не смогут редактировать унаследованные из шаблона элементы данных на узле сети, а имя шаблона будет отображаться как *Недоступный шаблон*.

Вкладка **Фильтр тегов проблем** позволяет задавать для групп пользователей права на основе тегов, чтобы видеть проблемы, отфильтрованные по имени и значению тега:

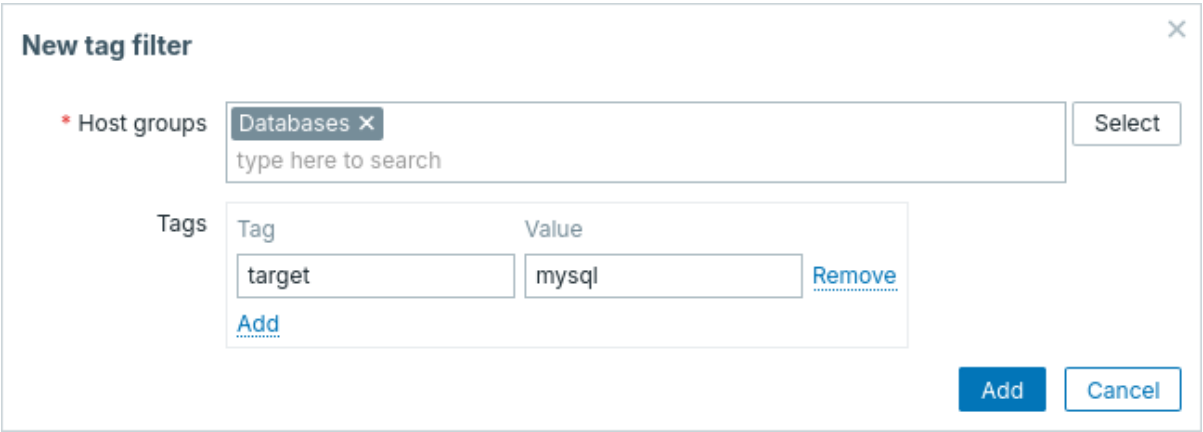


Нажмите Add, чтобы выбрать группы узлов сети. Чтобы выбрать группу узлов сети, для которой будет применён фильтр тегов, нажмите *Выбрать*, чтобы получить полный список существующих групп узлов сети, или начните вводить имя группы узлов сети, чтобы получить выпадающий список подходящих групп. Будут отображаться только группы узлов сети, поскольку фильтр тегов проблем нельзя применять к группам шаблонов.

Можно добавлять имена тегов без значений, но значения без имён добавлять нельзя. В блоке *Права доступа* отображаются только первые три тега (со значениями, если они есть); если тегов больше, их можно увидеть, нажав на значок ••• или наведя на него курсор.

Фильтр тегов позволяет разделить доступ к группе узлов сети и возможность видеть проблемы.

Например, если администратору баз данных нужно видеть только проблемы базы данных ”MySQL”, сначала необходимо 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 тега и значение тега указаны правильно, иначе группа пользователей не увидит ни одной проблемы.

Рассмотрим пример, когда пользователь является участником нескольких выбранных групп пользователей. В этом случае при фильтрации для тегов будет использоваться условие OR.

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*	
Databases	target	mysql	Databases	target	oracle	видны проблемы target:mysql или target:oracle
Не настроено в **Фильтр тегов проблем**			Databases	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 сущностям в результате сможет получить доступ непривилегированный пользователь. В следующем примере рассмотрим, как будет изменяться доступ к узлу сети **X** (в группе узлов сети 1) в различных ситуациях для пользователя, который состоит в группах пользователей A и B.

- Если у группы A есть только доступ *Чтение* к группе узлов сети 1, а у группы B — доступ *Чтение-запись* к группе узлов сети 1, пользователь получит доступ **Чтение-запись** к 'X'.

Attention:
Права доступа "Чтение-запись" имеют приоритет над правами "Чтение".

- В том же сценарии, что и выше, если 'X' одновременно также входит в группу узлов сети 2, доступ к которой **запрещён** для группы A или B, доступ к 'X' будет **недоступен**, несмотря на наличие доступа *Чтение-запись* к группе узлов сети 1.
- Если для группы A права доступа не определены, а у группы B есть доступ *Чтение-запись* к группе узлов сети 1, пользователь получит доступ **Чтение-запись** к 'X'.
- Если у группы A есть доступ *Запретить* к группе узлов сети 1, а у группы B есть доступ *Чтение-запись* к группе узлов сети 1, пользователю будет **запрещён** доступ к 'X'.

Прочие сведения

- Пользователь уровня Admin с доступом *Read-write* к узлу сети не сможет привязывать/отвязывать шаблоны, если у него нет доступа к группе шаблонов, к которой они принадлежат. При наличии доступа *Read* к группе шаблонов он сможет привязывать/отвязывать шаблоны к узлу сети, однако не будет видеть никаких шаблонов в списке шаблонов и не сможет работать 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-услуг.
- Любой пользователь, не являющийся Super Admin (включая 'guest'), может видеть карты сети, если карта пуста или содержит только изображения. Когда на карту добавляются узлы сети, группы узлов сети или триггеры, права доступа учитываются.
- сервер Zabbix не будет отправлять уведомления пользователям, указанным в качестве получателей операций действия, если доступ к соответствующему узлу сети явно "denied".

13 Хранение секретов

Обзор

Zabbix можно настроить на получение конфиденциальной информации из защищённого хранилища. Поддерживаются следующие службы управления секретами: HashiCorp Vault KV Secrets Engine - Version 2, CyberArk Vault CV12.

Секреты можно использовать для получения:

- значений пользовательских макросов
- учётных данных доступа к базе данных

Zabbix предоставляет доступ только для чтения к секретам в хранилище, предполагая, что управление секретами осуществляется кем-то другим.

Информацию о настройке конкретного поставщика хранилища секретов см. в:

- Настройка HashiCorp
- Настройка CyberArk

Кэширование значений секретов

По умолчанию значения секретных макросов хранилища извлекаются сервером Zabbix при каждом обновлении данных конфигурации, а затем сохраняются в кэше конфигурации. Прокси Zabbix получает значения секретных макросов 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 предупреждающее сообщение сервера.

Также можно настроить, чтобы значения макросов извлекались сервером Zabbix и прокси Zabbix независимо друг от друга.

Чтобы вручную запустить обновление кэшированных значений секретов из хранилища, используйте параметр командной строки option 'secrets_reload'.

Для учетных данных базы данных веб-интерфейса Zabbix кэширование по умолчанию отключено, но его можно включить, установив параметр `$DB['VAULT_CACHE'] = true` в zabbix.conf.php. Учетные данные будут храниться в локальном кэше с использованием временного каталога файловой системы. Веб-сервер должен разрешать запись в приватную временную папку (например, для Apache должен быть установлен параметр конфигурации `PrivateTmp=True`). Чтобы управлять частотой обновления/инвалидации кэша данных, используйте константу ZBX_DATA_CACHE_TTL constant .

Настройка TLS

Чтобы настроить TLS для обмена данными между компонентами Zabbix и хранилищем секретов, добавьте сертификат, подписанный центром сертификации (CA), в общесистемное хранилище доверенных сертификатов CA по умолчанию. Чтобы использовать другое расположение, укажите каталог в параметре конфигурации SSLCALocation Zabbix server/proxy, поместите файл сертификата в этот каталог, затем выполните команду CLI:

```
c_rehash .
```

1 Настройка CyberArk

В этом разделе объясняется, как настроить Zabbix для получения секретов из CyberArk Vault CV12.

Хранилище должно быть установлено и настроено, как описано в официальной документации 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 секретов для Zabbix сервера или прокси, укажите следующие параметры конфигурации в файле конфигурации:

- `Vault` — какой провайдер хранилища секретов должен использоваться;
- `VaultURL` — HTTP[S] URL сервера хранилища секретов;
- `VaultDBPath` — запрос к секрету в хранилище, содержащему учетные данные базы данных, которые будут получены по ключам "Content" и "UserName" (этот параметр можно использовать только в том случае, если не указаны DBUser и DBPassword);
- `VaultTLSCertFile`, `VaultTLSKeyFile` — имена файлов SSL-сертификата и ключа; настройка этих параметров не является обязательной, но настоятельно рекомендуется;
- `VaultPrefix` — пользовательский префикс для пути или запроса к хранилищу, в зависимости от типа 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 секретов при обработке макросов хранилища секретов сервером Zabbix (и прокси Zabbix, если настроено). Сервер Zabbix и прокси Zabbix не будут раскрывать макросы секретов хранилища, содержащие учетные данные БД из VaultDBPath.

Сервер Zabbix и прокси Zabbix считывают связанные с хранилищем параметры конфигурации 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>
}
```

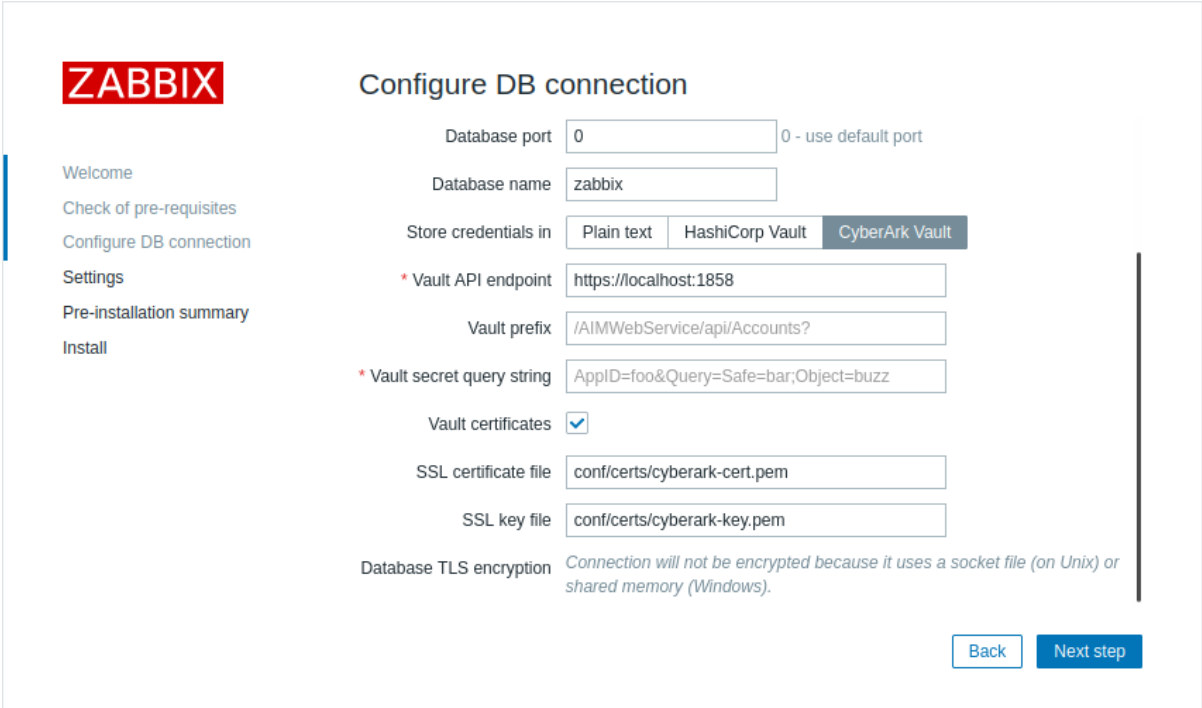
4. В результате Zabbix будет использовать следующие учетные данные для аутентификации в базе данных:

- 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 параметр *Store credentials in* в значение ”CyberArk Vault”.



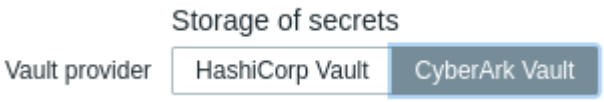
2. Затем заполните дополнительные параметры:

Параметр	Обязательный?	Значение по умолчанию	Описание
Конечная точка API хранилища	да	https://localhost:1858	Укажите URL для подключения к хранилищу в формате `scheme://host:port`
Префикс хранилища	нет	/AIMWebService/api/Accounts?	Укажите пользовательский префикс для пути или запроса хранилища. Если не указан, используется значение по умолчанию.
Строка запроса секрета хранилища	да		Запрос, который определяет, откуда должны быть получены учетные данные базы данных. Пример: `AppID=foo&Query=Safe=bar;Object=buzz`
Сертификаты хранилища	нет		После установки флажка появятся дополнительные параметры, позволяющие настроить аутентификацию клиента. Хотя этот параметр является необязательным, настоятельно рекомендуется включить его для взаимодействия с CyberArk Vault.
Файл SSL-сертификата	нет	conf/certs/cyberark-cert.pem	Путь к файлу SSL-сертификата. Файл должен быть в формате PEM. Если файл сертификата также содержит закрытый ключ, оставьте параметр файла SSL-ключа пустым.
Файл SSL-ключа	нет	conf/certs/cyberark-key.pem	Имя файла закрытого SSL-ключа, 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 разделе *Администрирование → Общие → Прочее* установлен в значение ”CyberArk Vault”.



Note:
Сервер Zabbix (и прокси Zabbix, если настроен) требует доступ к значениям макросов *Vault secret* из хранилища. Веб-интерфейс Zabbix такой доступ не требуется.

Значение макроса должно содержать запрос (в виде `query:key`).

Подробную информацию об обработке значений макросов Zabbix см. в разделе *Макросы Vault secret*.

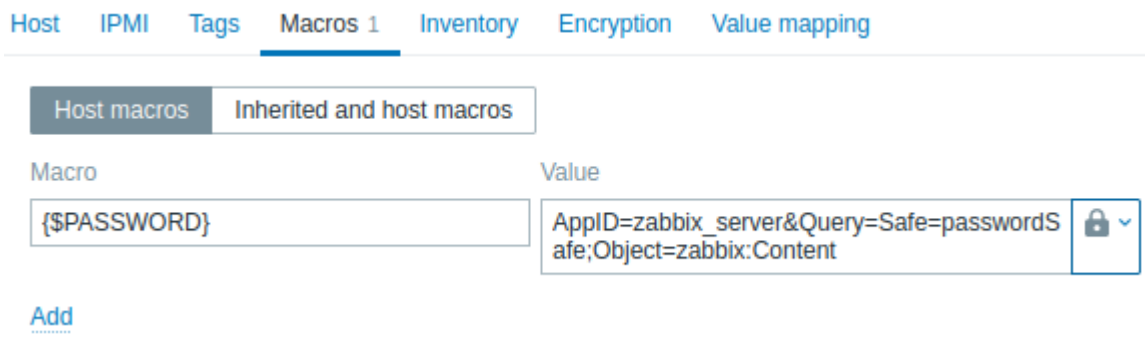
Синтаксис запроса

Символ двоеточия (”:”) зарезервирован для отделения запроса от ключа.

Если сам запрос содержит прямую косую черту или двоеточие, эти символы должны быть закодированы в формате URL (”/” кодируется как ”%2F”, ”:” кодируется как ”%3A”).

Пример

1. В Zabbix добавьте пользовательский макрос {$PASSWORD} типа *Vault secret* со значением `AppID=zabbix_server&Query=Safe`



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 ключ ”Content”:

```
{
    "Content": <password>,
    "UserName": <username>,
    "Address": <address>,
    "Database" :<Database>,
    "PasswordChangeInProcess":<PasswordChangeInProcess>
}
```

4. В результате Zabbix преобразует макрос {$PASSWORD} в значение - <password>

Обновление существующей конфигурации

Чтобы обновить существующую конфигурацию для получения секретов из CyberArk Vault:

1. Обновите параметры конфигурационного файла сервера Zabbix или прокси, как описано в разделе *Учетные данные базы данных*.
2. Обновите настройки подключения к БД, перенастроив веб-интерфейс Zabbix и указав необходимые параметры, как описано в разделе *Веб-интерфейс*. Чтобы перенастроить веб-интерфейс Zabbix, откройте URL настройки веб-интерфейса в браузере:

- для Apache: http://<server_ip_or_name>/zabbix/setup.php
- для Nginx: http://<server_ip_or_name>/setup.php

В качестве альтернативы эти параметры можно задать в конфигурационном файле веб-интерфейса (*zabbix.conf.php*):

```
$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. *Конфигурация HashiCorp*.

2 Конфигурация HashiCorp

Обзор

В этом разделе объясняется, как настроить Zabbix для получения секретов из HashiCorp Vault KV Secrets Engine — Version 2.

Хранилище должно быть развернуто и настроено, как описано в официальной документации HashiCorp.

Чтобы узнать о настройке TLS в Zabbix, см. *Хранение секретов*.

Получение учетных данных базы данных

Чтобы успешно получить секрет с учетными данными базы данных, необходимо настроить оба компонента:

- Zabbix сервер/прокси
- Zabbix веб-интерфейс

Сервер/прокси

Чтобы настроить Zabbix сервер или прокси, укажите следующие параметры конфигурации в файле конфигурации:

- `Vault` — какой провайдер хранилища секретов следует использовать;
- `VaultToken` — токен аутентификации хранилища секретов (подробности см. в файле конфигурации Zabbix сервера/прокси);
- `VaultURL` — HTTP[S] URL сервера хранилища секретов;
- `VaultDBPath` — путь к секрету в хранилище секретов, содержащему учетные данные базы данных (этот параметр можно использовать только в том случае, если не указаны DBUser и DBPassword); Zabbix сервер или прокси получит учетные данные по ключам ”password” и ”username”;
- `VaultPrefix` — пользовательский префикс для пути или запроса к хранилищу секретов, в зависимости от типа хранилища; если не указан, будет использовано наиболее подходящее значение по умолчанию.

Attention:
Параметры конфигурации `Vault`, `VaultToken`, `VaultURL` и `VaultPrefix` также используются для аутентификации в хранилище секретов при обработке макросов хранилища секретов Zabbix сервером (и Zabbix прокси, если настроено). Zabbix сервер и прокси не будут раскрывать макросы секретов хранилища, содержащие учетные данные БД из VaultDBPath.

Настоятельно рекомендуется использовать разные токены для разных прокси.

Zabbix сервер и Zabbix прокси считывают связанные с хранилищем секретов параметры конфигурации из *zabbix_server.conf* и *zabbix_proxy.conf* при запуске. Кроме того, Zabbix сервер и Zabbix прокси один раз при запуске считывают переменную окружения `VAULT_TOKEN`, а затем удаляют ее, чтобы она не была доступна через порожденные скрипты; наличие значения одновременно в параметрах `VaultToken` и `VAULT_TOKEN` является ошибкой.

Пример

1. В *zabbix_server.conf* укажите следующие параметры:

```
Vault=HashiCorp
VaultToken=hvs.CAESIIG_PILmULFYOsEyWHxkZ2mF2a8VPKNLE8eHqd4autYGGh4KHGh2cy5aeTY0NFNSaUp3ZnpWbDF1RUNjUkNTZEg
VaultURL=https://127.0.0.1:8200
VaultDBPath=database
VaultPrefix=/v1/secret/data/zabbix/
```

2. Выполните следующие CLI-команды, чтобы создать необходимый секрет в хранилище:

```
####                  "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:

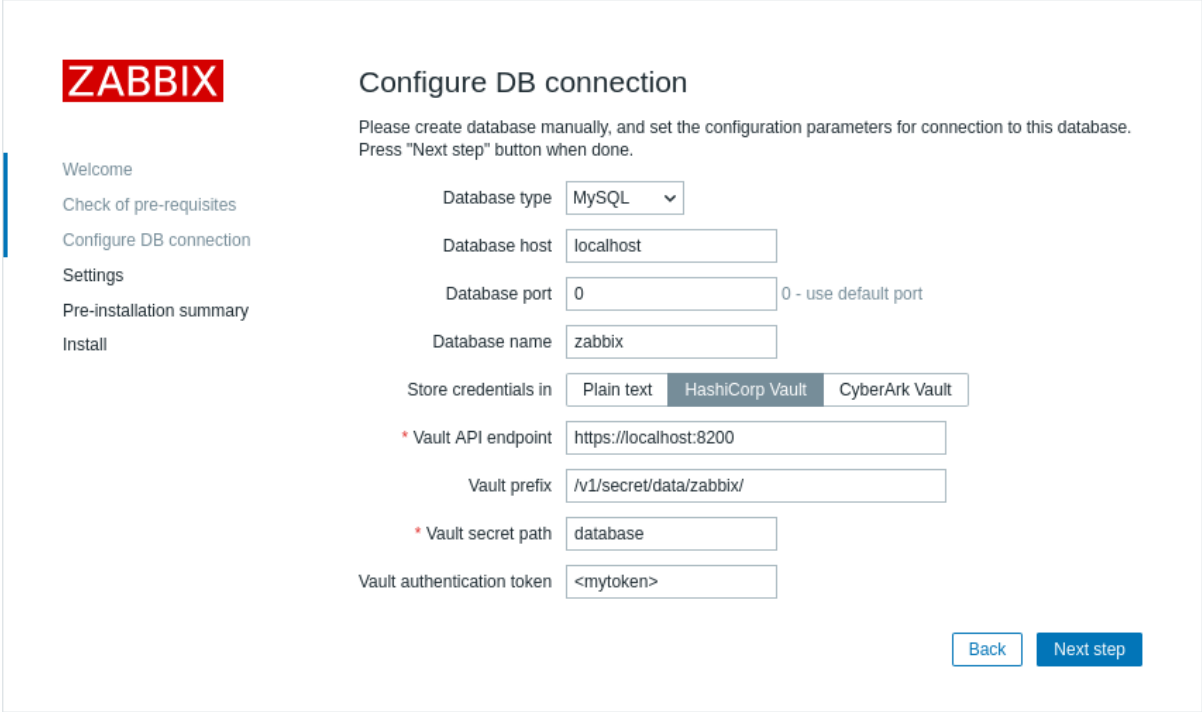
- Имя пользователя: zabbix
- Пароль: <password>

Веб-интерфейс

Веб-интерфейс Zabbix можно настроить на получение 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 были изменены после предыдущей установки веб-интерфейса, повторно запустите установку веб-интерфейса или обновите `zabbix.conf.php`. См. также: Обновление существующей конфигурации.

Во время **установки веб-интерфейса** параметры конфигурации необходимо указать на шаге *Configure DB Connection*:



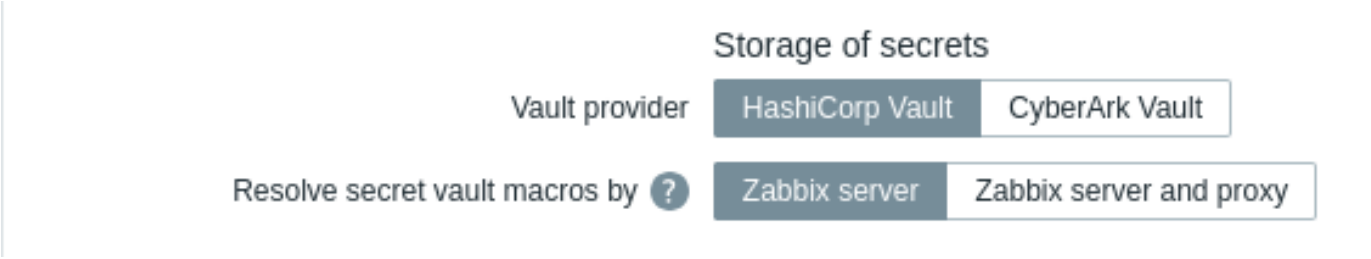
- Установите для параметра *Store credentials in* значение ”HashiCorp Vault”.
- Укажите параметры подключения:

Параметр	Обязательный	Значение по умолчанию	Описание
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 не указан, используется значение по умолчанию. Пример: `/v1/secret/data/zabbix/`
Vault secret path	нет		Путь к секрету, из которого должны быть получены учетные данные для базы данных по ключам ”password” и ”username”. Пример: `database`
Vault authentication token	нет		Укажите токен аутентификации для доступа только на чтение к пути секрета. Информацию о создании токенов и политиках хранилища см. в документации HashiCorp.

Получение значений пользовательских макросов

Чтобы использовать HashiCorp Vault для хранения значений пользовательских макросов *Vault secret*, убедитесь, что:

- сервер/прокси Zabbix настроен для работы с HashiCorp Vault;
- параметр *Vault provider* в разделе *Администрирование → Общие → Прочее* установлен в значение ”HashiCorp Vault” (по умолчанию);



Note:
Сервер Zabbix (и прокси Zabbix, если настроено) требует доступ к значениям макросов *Vault secret* из хранилища. Веб-интерфейс Zabbix такой доступ не требуется.

Значение макроса должно содержать ссылочный путь (в виде `path:key`, например, `macros:password`). Токен аутентификации, указанный при настройке сервера/прокси Zabbix (параметром `VaultToken`), должен предоставлять доступ только на чтение к этому пути.

Подробную информацию об обработке значений макросов в Zabbix см. в разделе *Макросы Vault secret*.

Синтаксис пути

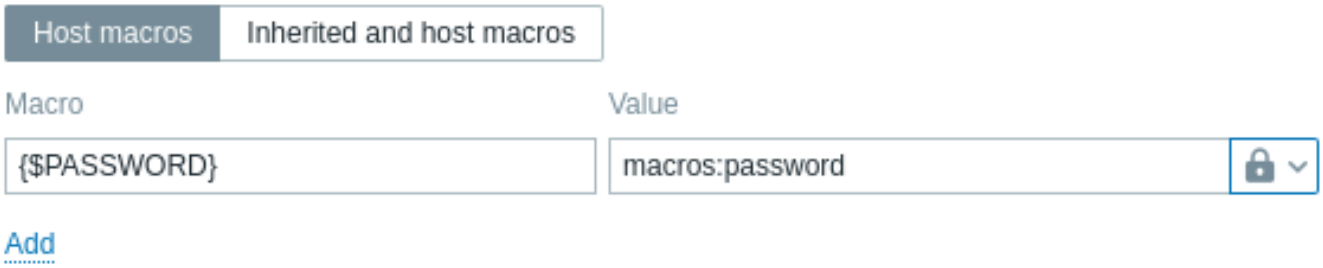
Символы косой черты (”/”) и двоеточия (”:”) являются зарезервированными.

Косая черта может использоваться только для отделения точки монтирования от пути (например, *secret/zabbix*, где точка монтирования — ”secret”, а путь — ”zabbix”). В случае макросов Vault двоеточие может использоваться только для отделения пути/запроса от ключа.

При необходимости можно выполнить URL-кодирование символов косой черты и двоеточия, если требуется создать точку монтирования с именем, разделённым косой чертой (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.

Пример

1. В Zabbix добавьте пользовательский макрос {$PASSWORD} типа ”Vault secret” со значением `macros:password`



2. Выполните следующие CLI-команды, чтобы создать необходимый секрет в хранилище:

```
#### Enable "secret/" mount point if not already enabled; note that "kv-v2" must be used.
vault secrets enable -path=secret/ kv-v2

#### Put new secret with key "password" under mount point "secret/" and path "zabbix/macros".
vault kv put -mount=secret zabbix/macros password=<password>

#### Test that secret is successfully added.
vault kv get secret/zabbix/macros

#### Finally test with Curl; note that "data" need to be manually added after mount point and "/v1" before
curl --header "X-Vault-Token: <VaultToken>" https://127.0.0.1:8200/v1/secret/data/zabbix/macros
```

3. В результате Zabbix преобразует макрос {$PASSWORD} в значение: <password>

Обновление существующей конфигурации

Чтобы обновить существующую конфигурацию для получения секретов из HashiCorp Vault:

1. Обновите параметры конфигурационного файла сервера Zabbix или прокси, как описано в разделе *Учетные данные базы данных*.

2. Обновите настройки подключения к БД, перенастроив веб-интерфейс Zabbix и указав необходимые параметры, как описано в разделе *Веб-интерфейс*. Чтобы перенастроить веб-интерфейс Zabbix, откройте URL настройки веб-интерфейса в браузере:

- для Apache: http://<server_ip_or_name>/zabbix/setup.php
- для Nginx: http://<server_ip_or_name>/setup.php

В качестве альтернативы эти параметры можно задать в конфигурационном файле веб-интерфейса (*zabbix.conf.php*):

```
$DB['VAULT']                    = 'HashiCorp';
$DB['VAULT_URL']                = 'https://localhost:8200';
$DB['VAULT_DB_PATH']            = 'database';
$DB['VAULT_TOKEN']              = '<mytoken>';
$DB['VAULT_CERT_FILE']          = '';
$DB['VAULT_KEY_FILE']           = '';
$DB['VAULT_PREFIX']             = '/v1/secret/data/zabbix/';
```

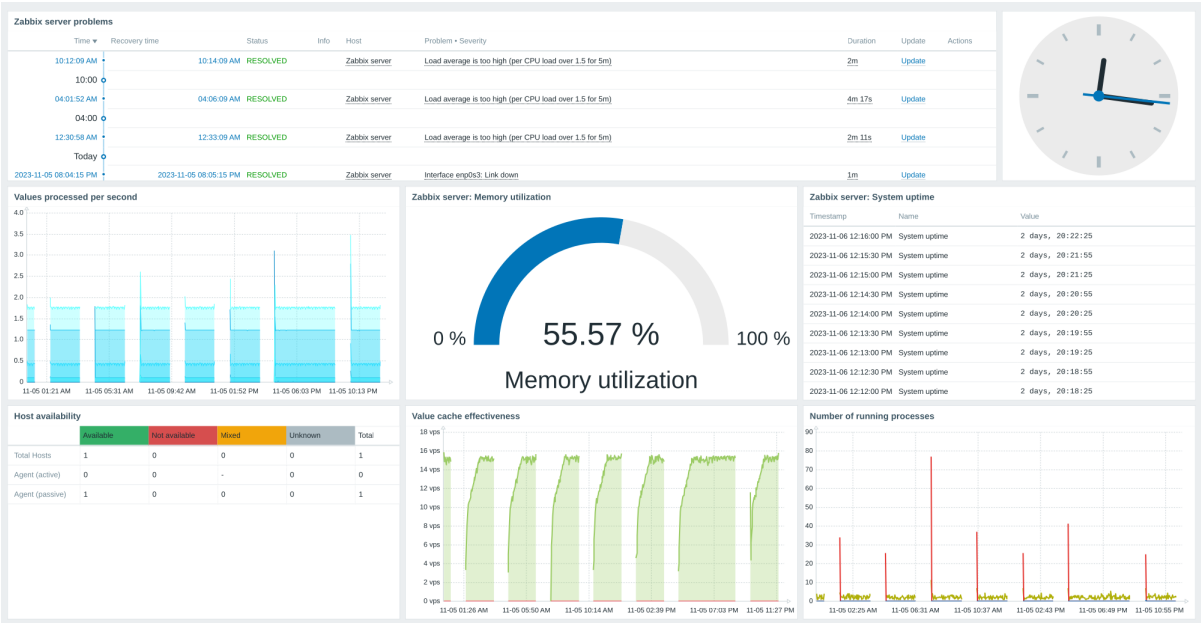
3. При необходимости настройте пользовательские макросы, как описано в разделе *Значения пользовательских макросов*.

Чтобы обновить существующую конфигурацию для получения секретов из CyberArk Vault, см. *Конфигурация CyberArk*.

14 Регулярные отчеты

Обзор

С помощью функции *Запланированные отчеты* вы можете настроить отправку PDF-версии выбранной панели указанным получателям через повторяющиеся интервалы времени.



Предварительные требования:

- Для включения генерации запланированных отчетов необходимо установить и правильно настроить Zabbix web service — инструкции см. в разделе Настройка запланированных отчетов.
- Пользователь должен иметь роль пользователя типа *Admin* или *Super admin* со следующими разрешениями:
  - *Scheduled reports* в блоке *Access to UI elements* (для просмотра настроек отчетов)
  - *Manage scheduled reports* в блоке *Access to actions* (для создания/редактирования отчетов)

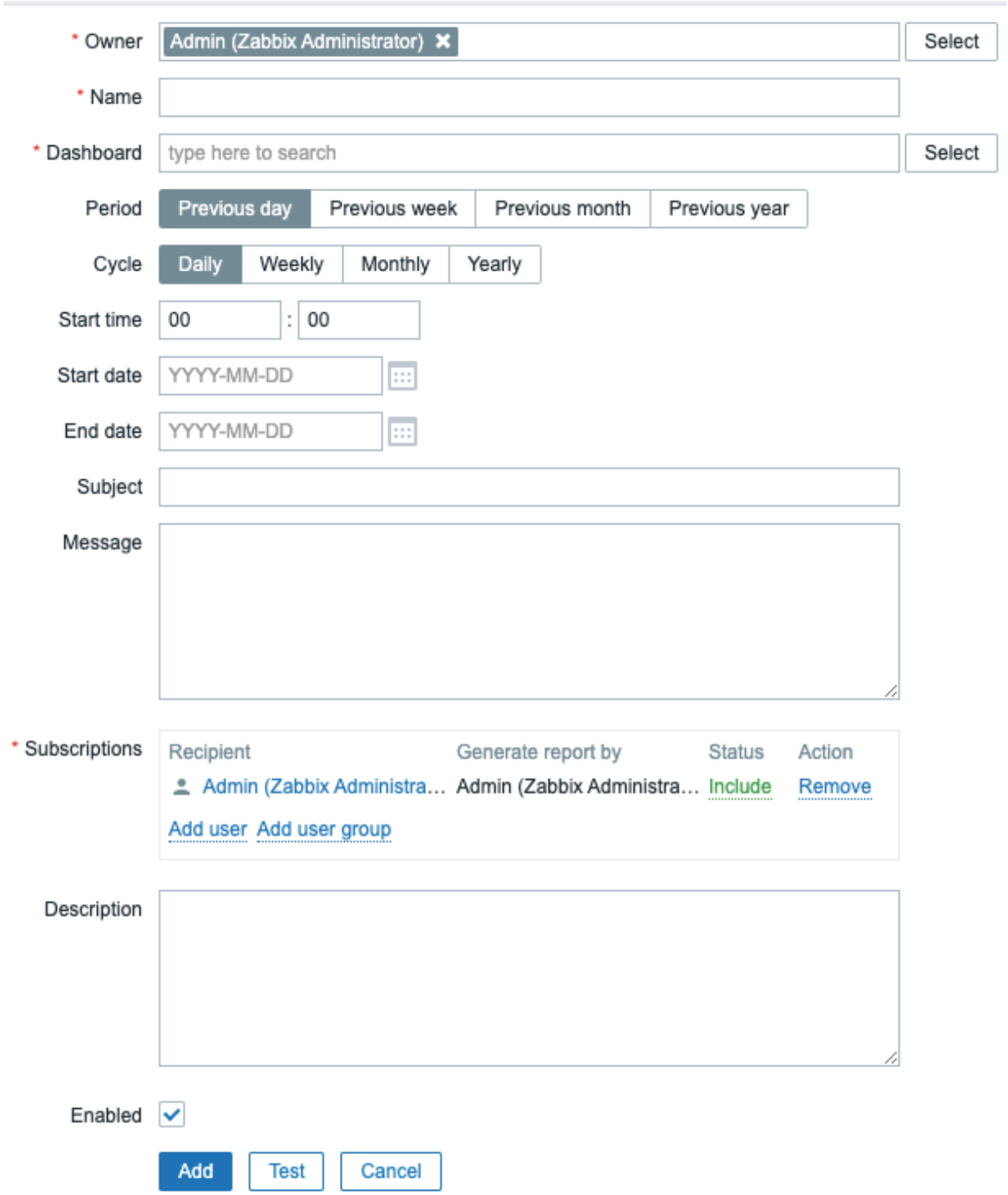
Чтобы создать запланированный отчет в веб-интерфейсе Zabbix, выполните следующее:

- Перейдите в: *Reports* > *Scheduled reports*.
- Нажмите *Create report* в правом верхнем углу экрана.
- Введите параметры отчета в форме.

Вы также можете создать отчет, открыв существующий, нажав кнопку *Clone*, а затем сохранив его под другим именем.

Конфигурация

Вкладка *Запланированные отчеты* содержит общие атрибуты отчета.



Все обязательные поля ввода отмечены красной звездочкой.

Параметр	Описание
Владелец	Пользователь, который создает отчет. Пользователи уровня *Super admin* могут изменять владельца. Для пользователей уровня *Admin* это поле доступно только для чтения.
Имя	Имя отчета; должно быть уникальным.
Панель	Панель, на основе которой формируется отчет; одновременно можно выбрать только одну панель. Чтобы выбрать панель, начните вводить имя — появится список подходящих панелей; прокрутите вниз для выбора. Либо нажмите *Выбрать* рядом с полем и выберите панель из отображаемого списка.
Период	Период времени, за который будет подготовлен отчет. Выберите предыдущий день, неделю, месяц или год.
Цикл	Частота формирования отчета. Отчеты можно отправлять ежедневно, еженедельно, ежемесячно или ежегодно. Режим ”Еженедельно” позволяет выбрать дни недели, в которые будет отправляться отчет.

Параметр	Описание
Время начала	Время суток в формате hh:mm, когда будет подготовлен отчет. Обратите внимание, что будет использоваться часовой пояс сервера Zabbix.
Повторять по	Дни недели, в которые будет отправляться отчет. Это поле доступно только если для *Цикл* установлено значение ”Еженедельно”.
Дата начала	Дата, с которой должно начаться регулярное формирование отчета.
Дата окончания	Дата, когда регулярное формирование отчета должно быть остановлено.
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. В качестве получателя отчета можно указать любого пользователя Zabbix с настроенным email-каналом оповещений. Нажмите *Добавить пользователя* или *Добавить группу пользователей*, чтобы добавить больше получателей. Нажмите на имя пользователя, чтобы изменить настройки: *Формировать отчет от имени* — следует ли формировать данные отчета на основе прав доступа к панели текущего пользователя или получателя. *Статус* — выберите ”Включить”, чтобы отправлять отчет пользователю, или ”Исключить”, чтобы запретить отправку отчета этому пользователю. Как минимум один пользователь должен иметь статус ”Включить”. Статус ”Исключить” можно использовать, чтобы исключить отдельных пользователей из включенной группы пользователей. Обратите внимание, что пользователи с недостаточными правами (то есть пользователи с ролью на основе типа пользователя *Admin*, которые не являются членами той же группы пользователей, что и получатель или владелец отчета) будут видеть ”Недоступный пользователь” или ”Недоступная группа пользователей” вместо фактических имен в полях *Получатель* и *Формировать отчет от имени*; поля *Статус* и *Действие* будут отображаться только для чтения.
Включено	Статус отчета. Снятие этого флажка отключит отчет.
Описание	Необязательное описание отчета. Это описание предназначено для внутреннего использования и не будет отправлено получателям отчета.

Кнопки диалога

Кнопки в нижней части диалога позволяют выполнить несколько видов операций.

Add	Добавление отчета. Эта кнопка доступна только для новых отчетов.
Update	Обновление свойств отчета.
Clone	Создание другого отчета на основе свойств текущего отчета.
Test	Проверка корректности настройки отчета методом отправки отчета текущему пользователю.
Delete	Удаление отчета.
Cancel	Отмена изменения свойств отчета.

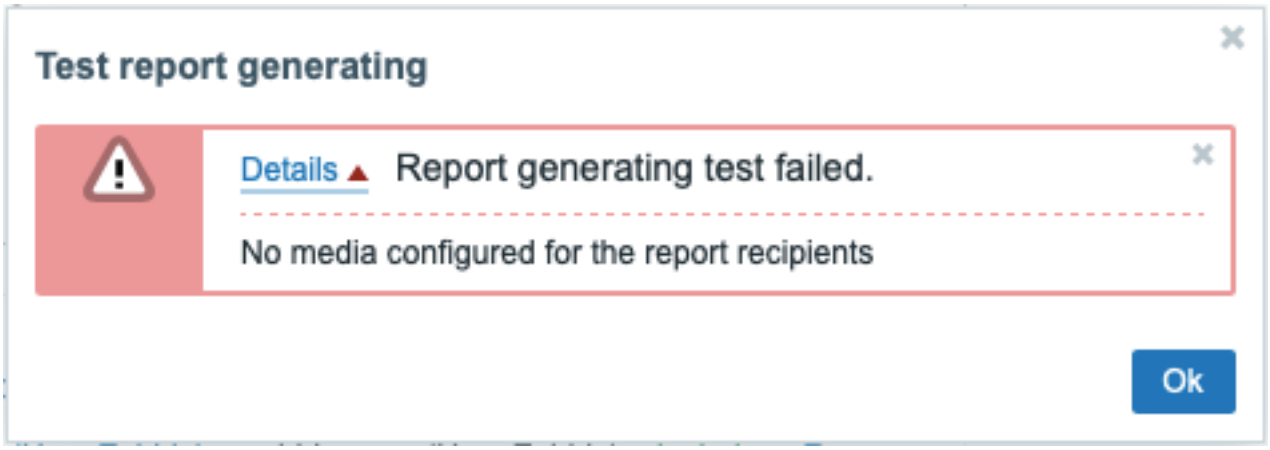
Тестирование

Чтобы протестировать отчет, нажмите кнопку *Тест* в нижней части формы настройки отчета.

> **Note:**
> Кнопка *Тест* недоступна, если форма настройки отчета была открыта из меню действий панели.

Если настройка выполнена правильно, тестовый отчет немедленно отправляется текущему пользователю. Для тестовых отчетов подписчики и настройки пользователя *Generate report by* игнорируются.

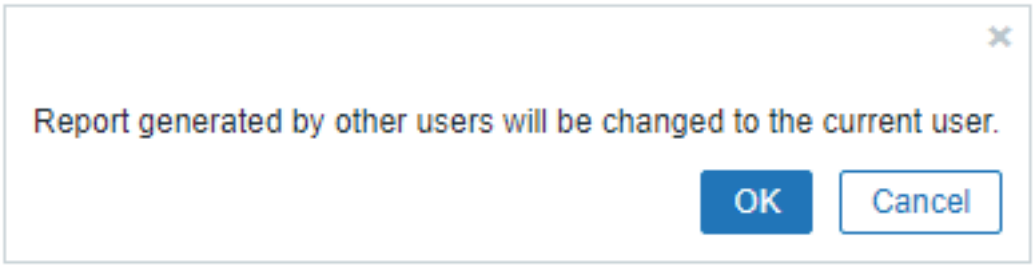
Если настройка выполнена неправильно, отображается сообщение об ошибке с описанием возможной причины.



Обновление отчета

Чтобы обновить существующий отчет, нажмите на имя отчета, внесите необходимые изменения в конфигурацию, а затем нажмите кнопку *Update*.

Если существующий отчет обновляется другим пользователем и этот пользователь изменяет панель, то при нажатии кнопки *Update* будет отображено предупреждающее сообщение: ”Report generated by other users will be changed to the current user”.



Нажатие *OK* на этом этапе приведет к следующим изменениям:

- Настройки *Generate report by* будут обновлены так, чтобы отображать пользователя, который последним редактировал отчет (если только в *Generate report by* не указан получатель).
- Пользователи, отображаемые как ”Inaccessible user” или ”Inaccessible user group”, будут удалены из списка подписчиков отчета.

Нажатие *Cancel* закроет форму конфигурации и отменит обновление отчета.

Клонирование отчета

Чтобы быстро клонировать существующий отчет, нажмите кнопку *Клонировать* в нижней части формы настройки существующего отчета. При клонировании отчета, созданного другим пользователем, текущий пользователь становится владельцем нового отчета.

Настройки отчета будут скопированы в новую форму настройки отчета с учетом прав пользователя:

- Если пользователь, клонирующий отчет, не имеет прав к панели, поле *Панель* будет очищено.
- Если пользователь, клонирующий отчет, не имеет прав к некоторым пользователям или группам пользователей в списке *Подписки*, недоступные получатели не будут клонированы.
- Настройки *Генерировать отчет от имени* будут обновлены так, чтобы отображать текущего пользователя (если только в *Генерировать отчет от имени* не указан получатель).

Измените необходимые настройки и имя отчета, затем нажмите *Добавить*.

###15 Экспорт данных {#manual-config-export}

Обзор

Zabbix поддерживает экспорт данных в режиме реального времени двумя способами:

- экспорт в файлы
- потоковая передача во внешние системы

Следующие объекты могут быть экспортированы:

- триггерные события
- значения элементов данных
- тенденции (только экспорт в файлы)

####1 Экспорт в файлы {#manual-config-export-files}

Обзор

Можно настроить экспорт в реальном времени событий триггеров, значений элементов данных и трендов в формате JSON с разделением строками.

Экспорт выполняется в файлы, где каждая строка файла экспорта представляет собой объект JSON. Преобразования значений не применяются.

В случае ошибок (если данные не могут быть записаны в файл экспорта, или файл экспорта не может быть переименован, или после его переименования не удаётся создать 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 записи устранена.

Файл экспорта создаётся с правами на чтение и запись только для владельца файла. Дополнительно файл доступен для чтения группе владельца. Все остальные права запрещены.

Подробную информацию о том, какие именно данные экспортируются, смотрите на странице протокола экспорта.

Обратите внимание, что у узла сети/элемента данных могут отсутствовать метаданные (группы узлов сети, имя узла сети, имя элемента данных), если узел сети/элемент данных был удалён после получения данных, но до того, как сервер экспортировал данные.

Конфигурация

Экспорт в реальном времени событий триггеров, значений элементов данных и трендов настраивается указанием каталога для файлов экспорта — см. параметр ExportDir в конфигурации сервера.

Доступны ещё два параметра:

- `ExportFileSize` может использоваться для задания максимально допустимого размера отдельного файла экспорта. Когда процессу требуется записать данные в файл, он сначала проверяет размер файла. Если он превышает настроенное ограничение размера, файл переименовывается с добавлением .old к его имени, а затем создаётся новый файл с исходным именем.

> **Attention:**
> Для каждого процесса, который будет записывать данные, создаётся отдельный файл (то есть примерно 4–30 файлов). Так как размер файла экспорта по умолчанию составляет 1G, хранение больших файлов экспорта может быстро исчерпать дисковое пространство.

- `ExportType` позволяет указать, какие типы сущностей (события, история, тренды) будут экспортироваться.

2 Потоковая передача во внешние системы

Обзор

Можно передавать значения элементов данных и события из Zabbix во внешние системы по HTTP (см. подробности протокола).

Фильтр тегов можно использовать для передачи подмножеств значений элементов данных или событий.

За потоковую передачу данных отвечают два процесса сервера Zabbix: `connector manager` и `connector worker`. Внутренний элемент данных Zabbix `zabbix[connector_queue]` позволяет отслеживать количество значений, поставленных в очередь коннектора.

Конфигурация

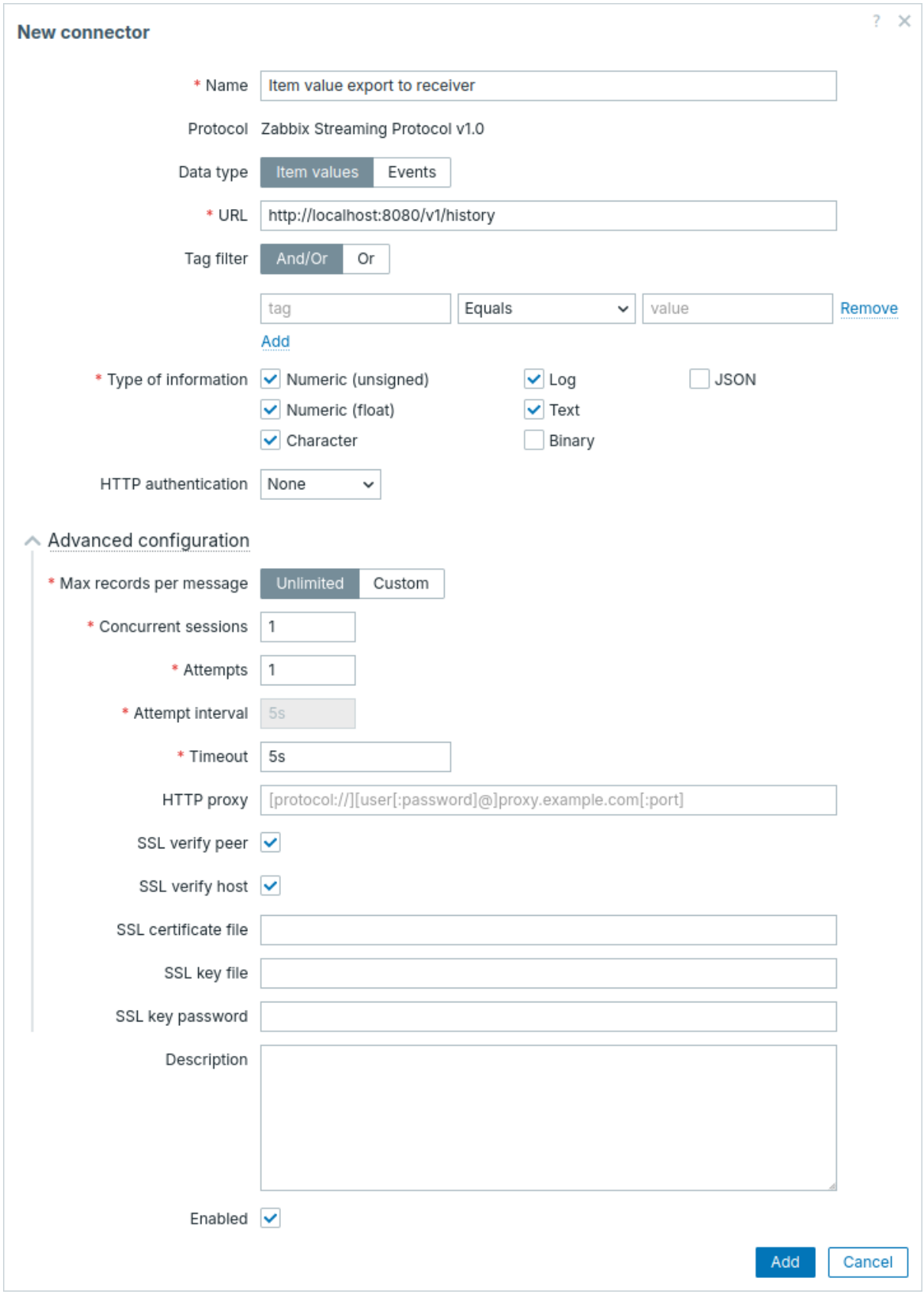
Для настройки потоковой передачи данных во внешнюю систему необходимо выполнить следующие шаги:

1. Настроить удалённую систему для приёма данных от Zabbix. Для этой цели доступны следующие инструменты:

- Пример простого приёмника, который записывает полученную информацию в файлы `events.ndjson` и `history.ndjson`.
- Kafka connector for Zabbix server — лёгкий сервер, написанный на Go, предназначенный для пересылки значений элементов данных и событий с сервера Zabbix в брокер Kafka.

2. Установить необходимое количество процессов connector worker в Zabbix, изменив параметр `StartConnectors` в `zabbix_server.conf`. Количество процессов connector worker должно соответствовать (или превышать, если число одновременных сессий больше 1) настроенному количеству коннекторов в веб-интерфейсе Zabbix. Затем перезапустите сервер Zabbix.

3. Настроить новый коннектор в веб-интерфейсе Zabbix (*Administration* > *General* > *Connectors*) и перезагрузить кэш сервера командой `zabbix_server -R config_cache_reload`.



Обязательные поля отмечены звёздочкой.

Параметр	Описание
Name	Введите имя коннектора.
Data type	Выберите тип данных для потоковой передачи: **Item values** — передавать значения элементов данных из Zabbix во внешние системы; **Events** — передавать события из Zabbix во внешние системы.
URL	Введите URL приёмника. Поддерживаются пользовательские макросы.
Tag filter	Экспортировать только значения элементов данных или события, соответствующие фильтру тегов. Если не задано, экспортируется всё. 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: **Exists** — включить указанные имена тегов; **Equals** — включить указанные имена и значения тегов (с учётом регистра); **Contains** — включить указанные имена тегов, если значения тегов содержат введённую строку (поиск подстроки, без учёта регистра); **Does not exist** — исключить указанные имена тегов; **Does not equal** — исключить указанные имена и значения тегов (с учётом регистра); **Does not contain** — исключить указанные имена тегов, если значения тегов содержат введённую строку (поиск подстроки, без учёта 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.
Type of information	Выберите тип информации (numeric (unsigned), numeric (float), character и т. д.), по которому будут фильтроваться значения элементов данных, передаваемые коннектором. Это поле доступно, если *Data type* установлено в ”Item values”.
HTTP authentication	Выберите вариант аутентификации: **None** — аутентификация не используется; **Basic** — используется базовая аутентификация; **NTLM** — используется аутентификация NTLM (Windows NT LAN Manager); **Kerberos** — используется аутентификация Kerberos (см. также: Configuring Kerberos with Zabbix); **Digest** — используется аутентификация Digest; **Bearer** — используется аутентификация Bearer.
Username	Введите имя пользователя (до 255 символов). Поддерживаются пользовательские макросы. Это поле доступно, если *HTTP authentication* установлено в ”Basic”, ”NTLM”, ”Kerberos” или ”Digest”.
Password	Введите пароль пользователя (до 255 символов). Поддерживаются пользовательские макросы. Это поле доступно, если *HTTP authentication* установлено в ”Basic”, ”NTLM”, ”Kerberos” или ”Digest”.
Bearer token	Введите Bearer token. Поддерживаются пользовательские макросы. Это поле доступно и обязательно, если *HTTP authentication* установлено в ”Bearer”.
Advanced configuration	Щёлкните по метке *Advanced configuration*, чтобы отобразить дополнительные параметры конфигурации (см. ниже).
Max records per message	Укажите максимальное количество значений или событий, которое может быть передано в одном сообщении.
Concurrent sessions	Выберите количество процессов отправки, которые будут запущены для этого коннектора. Можно указать до 100 сессий; значение по умолчанию — ”1”.
Attempts	Количество попыток потоковой передачи данных. Можно указать до 5 попыток; значение по умолчанию — ”1”.
Attempt interval	Укажите, сколько времени коннектор должен ждать после неудачной попытки передачи данных. Можно указать до 10s; значение по умолчанию — ”5s”. Это поле доступно, если *Attempts* установлено в ”2” или больше. Неудачными считаются попытки, при которых не удалось установить соединение, либо код HTTP-ответа не равен 200, 201, 202, 203, 204. Повторные попытки **запускаются** в случае ошибок связи или если код HTTP-ответа не равен 200, 201, 202, 203, 204, 400, 401, 403, 404, 405, 415, 422. Перенаправления обрабатываются, поэтому 302 -> 200 считается успешным ответом; тогда как 302 -> 503 приведёт к повторной попытке.

Параметр	Описание
Timeout	Укажите тайм-аут сообщения (1-60 секунд, по умолчанию — 5 секунд). Поддерживаются суффиксы времени, например 30s, 1m. Поддерживаются пользовательские макросы.
HTTP proxy	Можно указать HTTP-прокси в следующем формате: `[protocol://][username[:password]@]proxy.example.com[:port]` Поддерживаются пользовательские макросы. Необязательный префикс `protocol://` может использоваться для указания альтернативных протоколов прокси (поддержка префикса протокола добавлена в cURL 7.21.7). Если протокол не указан, прокси будет рассматриваться как HTTP-прокси. По умолчанию будет использоваться порт 1080. Если указан *HTTP proxy*, прокси переопределит связанные с прокси переменные окружения, такие как `http_proxy`, `HTTPS_PROXY`. Если не указан, прокси не будет переопределять связанные с прокси переменные окружения. Введённое значение передаётся как есть, без какой-либо проверки корректности. Также можно указать адрес SOCKS-прокси. Если указан неверный протокол, соединение завершится ошибкой, и элемент данных станет неподдерживаемым. Обратите внимание, что для HTTP-прокси поддерживается только простая аутентификация.
SSL verify peer	Установите флажок, чтобы проверять SSL-сертификат веб-сервера. Сертификат сервера будет автоматически взят из системного расположения центра сертификации (CA). Вы можете переопределить расположение файлов CA с помощью параметра конфигурации сервера Zabbix или прокси `SSLCALocation`.
SSL verify host	Установите флажок, чтобы проверять соответствие поля *Common Name* или поля *Subject Alternate Name* сертификата веб-сервера. Это задаёт параметр cURL `CURLOPT_SSL_VERIFYHOST`.
SSL certificate file	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 *SSL key file* пустым. Если ключ зашифрован, укажите пароль в поле *SSL key password*. Каталог, содержащий этот файл, задаётся параметром конфигурации сервера Zabbix или прокси `SSLCertLocation`.
SSL key file	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`.
SSL key password	Пароль файла закрытого SSL-ключа. Поддерживаются пользовательские макросы.
Description	Введите описание коннектора.
Enabled	Установите флажок, чтобы включить коннектор.

Когда Kafka connector настроен со списком адресов bootstrap-брокеров, разделённых запятыми (например, `Kafka.URL=kafka1.exam` клиент Kafka подключается к брокеру(ам), ответившему(им) первым, и использует их метаданные 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 при запуске могут появляться предупреждения, 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 (частные сети, сети контейнеров или нестандартные настройки DNS/hosts) имена узлов 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 путаницу, убедитесь, что все адреса `Kafka.URL` принадлежат одному и тому же кластеру Kafka, проверьте разрешение DNS с узла, на котором работает коннектор, а также параметр `advertised.listeners` брокеров, и по возможности используйте адреса, которые разрешаются в advertised-адрес брокера.

Протокол

Обмен данными между сервером и получателем выполняется по HTTP с использованием REST API, NDJSON, "Content-Type: application/x-ndjson".

Подробнее см. в разделе Протокол экспорта JSON с разделением по символу новой строки.

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 HTTP-ответа и полезной нагрузки JSON. Код состояния HTTP-ответа должен быть ”200”, ”201”, ”202”, ”203” или ”204” для запросов, которые были успешно обработаны, в противном случае — для неуспешных запросов.

Пример успешного ответа:

```
localhost:8080/v1/history": HTTP/1.1 200 OK
Date: Wed, 11 Jan 2023 16:40:30 GMT
Content-Length: 0
```

Пример с ошибками:

```
localhost:8080/v1/history": HTTP/1.1 422 Unprocessable Entity
Content-Type: application/json
Date: Wed, 11 Jan 2023 17:07:36 GMT
Content-Length: 55

{"error":"invalid character '{' after top-level value"}
```

3 SNMP-шлюз

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

Инструкции по следующим действиям см. в файле readme SNMP gateway:

- установка и настройка snmpd;
- включение поддержки AgentX;
- настройка Zabbix SNMP gateway;
- настройка SNMP-ловушек для изменений состояния триггера.

Получение данных

После правильной настройки 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 проблемных триггерах в конфигурации SNMP gateway:

- по важности (по умолчанию `ProblemMinSeverity=-1`)
- скрывая подтверждённые 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 подтверждёнными/неподтверждёнными/всеми проблемами (по умолчанию `CountAcknowledgeStatus=all`)

6 Мониторинг сервисов

Обзор Мониторинг сервисов — это мониторинг на уровне бизнеса, который можно использовать для получения общего представления обо всём дереве сервисов ИТ-инфраструктуры, выявления слабых мест инфраструктуры, расчёта 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 места на диске, высокая загрузка процессора и т. д. Мониторинг сервисов также предоставляет возможность находить первопричину проблемы, если сервис работает не так, как ожидается.

Мониторинг сервисов позволяет создавать иерархическое представление отслеживаемых данных.

Очень простая структура сервисов может выглядеть так:

```
|
|-
| |
| |-          1
```

| |
| |– 2
|
|–

Каждый узел структуры имеет атрибут состояния. Состояние вычисляется и распространяется 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, должен ли родительский сервис переходить в «состояние проблемы» на основе состояний дочерних сервисов. Данные о проблемах сервисов затем можно использовать для расчёта SLA и отправки отчётов SLA на основе гибкого набора условий.

Мониторинг сервисов настраивается в меню Services, которое состоит из следующих разделов:

- Services

Раздел Services позволяет построить 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 выявлять проблемы, которые привели к изменению состояния сервиса.

- SLA

В этом разделе можно определить соглашения об уровне сервиса и задать целевые показатели уровня сервиса для конкретных сервисов.

- SLA report

В этом разделе можно просматривать отчёты SLA.

Действия сервиса

Вы также можете настроить действия сервиса.

Действия сервиса являются необязательными и позволяют:

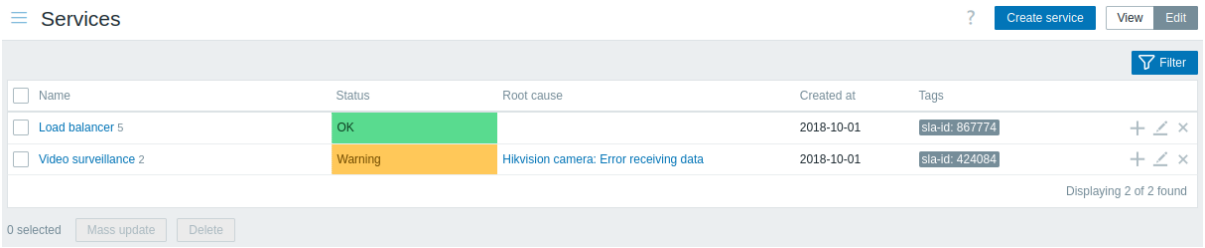
- отправлять уведомление о недоступности сервиса
- выполнять удалённую команду на сервере Zabbix при изменении состояния сервиса
- отправлять уведомление о восстановлении, когда сервис снова доступен.

См. также:

- Пример настройки мониторинга SLA
- Примечания об обновлении сервисов с версий Zabbix ниже 6.0

1 Дерево сервисов

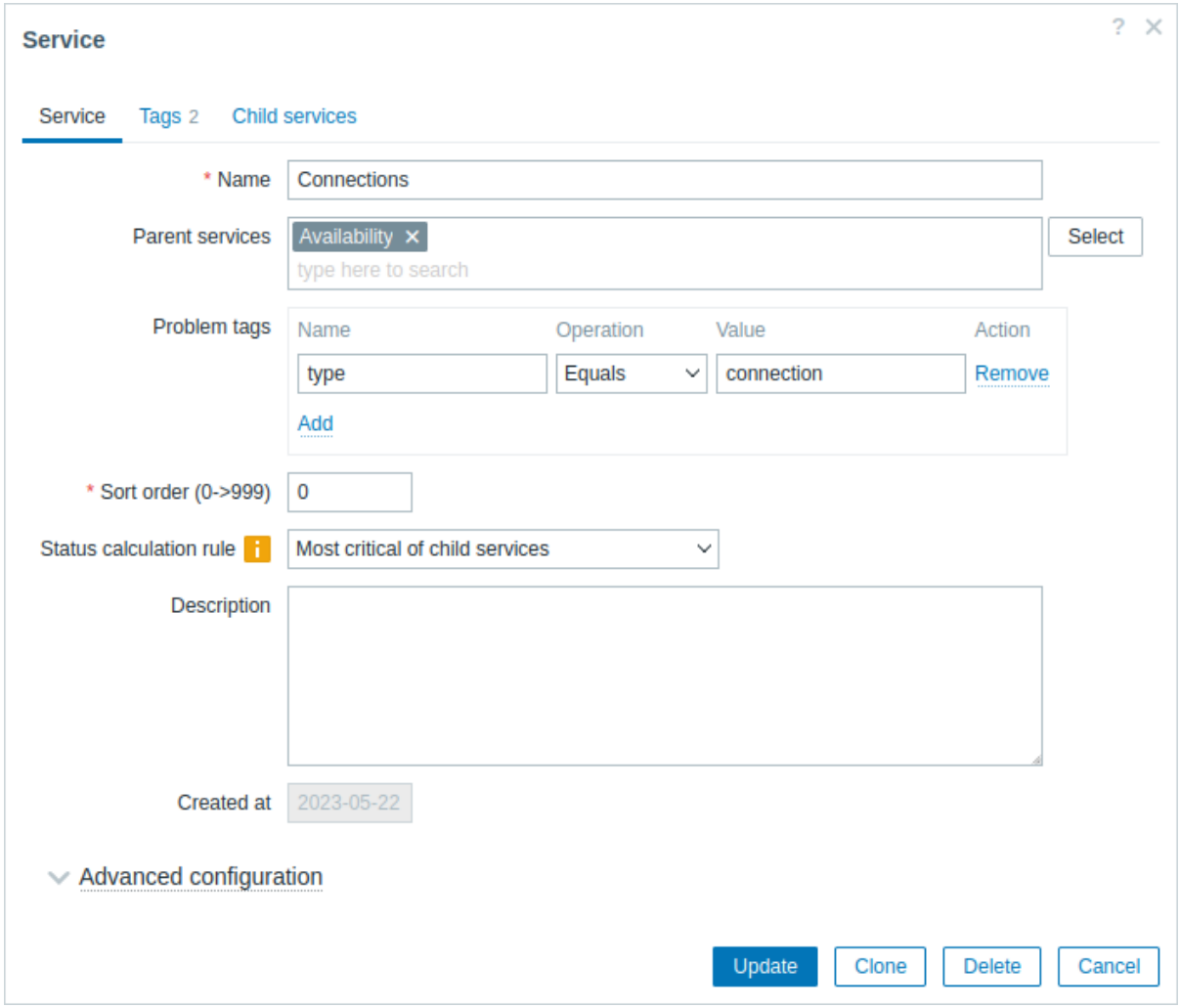
Дерево сервисов настраивается в разделе меню *Services -> Services*. В правом верхнем углу переключитесь из режима просмотра в режим редактирования.



Чтобы настроить новый сервис, нажмите кнопку *Create service* в правом верхнем углу.

Чтобы быстро добавить дочерний сервис, вы также можете нажать значок плюса рядом с родительским сервисом. Откроется та же форма настройки сервиса, но параметр *Parent services* будет уже заполнен.

Настройка сервиса На вкладке **Service** укажите необходимые параметры сервиса:

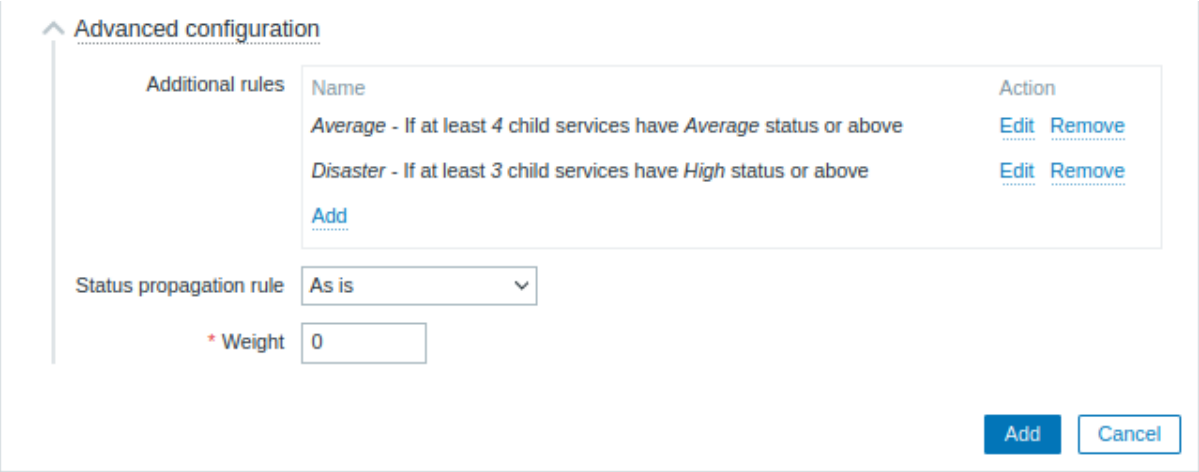


Все обязательные поля ввода отмечены красной звёздочкой.

Параметр	Описание
Name	Имя сервиса.
Parent services	Родительские сервисы, к которым относится данный сервис. Оставьте это поле пустым, если вы добавляете сервис самого высокого 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 имён тегов всегда чувствительно к регистру. Для каждого условия доступны два оператора: **Equals** — включить указанные имена и значения тегов (с учётом регистра) **Contains** — включить указанные имена тегов, значения которых содержат введённую строку (поиск подстроки, с учётом регистра)
Sort order	Порядок сортировки для отображения, сначала меньшие значения.
Status calculation rule	Правило расчёта состояния сервиса: **Most critical if all children have problems** — наиболее критичная проблема в дочерних сервисах используется для окрашивания состояния сервиса, если проблемы есть у всех дочерних сервисов **Most critical of child services** — наиболее критичная проблема в дочерних сервисах используется для окрашивания состояния сервиса **Set status to OK** — не рассчитывать состояние сервиса Дополнительные правила расчёта состояния можно настроить в параметрах расширенной конфигурации.
Description	Описание сервиса.
Created at	Время создания сервиса; отображается при редактировании существующего сервиса.

Параметр	Описание
Advanced configuration	Щёлкните по метке *Advanced configuration*, чтобы отобразить параметры расширенной конфигурации.

Расширенная настройка



Параметр	Описание
Дополнительные правила	Нажмите *Добавить*, чтобы настроить дополнительные правила вычисления статуса.
Установить статус в	Установите статус сервиса в *OK* (по умолчанию), *Не классифицировано*, *Информация*, *Предупреждение*, *Средняя*, *Высокая* или *Катастрофа* в случае совпадения условия.
Условие	Выберите условие для дочерних сервисов прямого уровня: **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)	Установите значение N или W (1-100000), либо N% (1-100) в условии.
Status	Выберите значение *Status* в условии: *OK* (по умолчанию), *Не классифицировано*, *Информация*, *Предупреждение*, *Средняя*, *Высокая* или *Катастрофа*.
Правило распространения статуса	Правило распространения статуса сервиса на родительский сервис: **Как есть** - статус распространяется без изменений **Повысить на** - вы можете повысить распространяемый статус на 1–5 уровней серьезности **Понизить на** - вы можете понизить распространяемый статус на 1–5 уровней серьезности **Игнорировать этот сервис** - статус вообще не распространяется на родительский сервис **Фиксированный статус** - статус распространяется статически, то есть всегда как один и тот же
Вес	Вес сервиса (целое число в диапазоне от 0 (по умолчанию) до 1000000).

Note:
Дополнительные правила вычисления статуса могут использоваться только для повышения уровня серьезности относительно уровня, вычисленного согласно основному параметру *Правило вычисления статуса*. Если согласно дополнительным правилам статус должен быть Предупреждение, а согласно *Правилу вычисления статуса* статус — Катастрофа, то сервис будет иметь статус Катастрофа.

Вкладка **Теги** подержит теги уровня услуг. Теги уровня услуг используются для идентификации услуги. Теги этого типа не используются для сопоставления проблем к услуге (для этого используйте *Теги проблем* с первой вкладки).

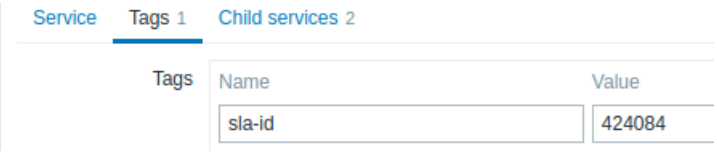
Вкладка **Дочерние услуги** позволяет задать зависимые услуги. Нажмите на *Добавить* для добавления услуги из всего списка существующих услуг. Если вы хотите добавить новую дочернюю услугу, сначала сохраните эту услугу, затем нажмите на иконку плюса рядом с услугой, которую вы только что создали.

Теги В сервисах есть два разных типа тегов:

- Теги сервиса
- Теги проблем

Теги сервиса

Теги сервиса используются для сопоставления сервисов с действиями сервиса и SLA. Эти теги указываются на вкладке настройки сервиса *Теги*. Для сопоставления SLA используется логика *ИЛИ*: сервис будет сопоставлен с SLA, если у него есть хотя бы один совпадающий тег. В действиях сервиса правила сопоставления можно настраивать, используя логику *И*, *ИЛИ* или *И/ИЛИ*.

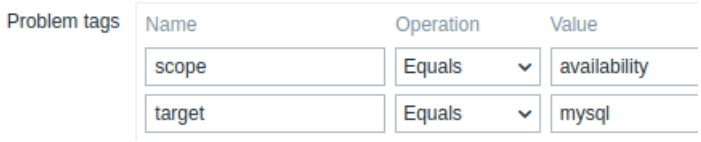


Теги проблем

Теги проблем используются для сопоставления проблем и сервисов. Эти теги указываются на вкладке настройки основного сервиса.

Только дочерние сервисы самого нижнего уровня иерархии могут иметь заданные теги проблем и напрямую соотноситься с проблемами. Если теги проблем совпадают, статус сервиса изменится на тот же статус, что и у проблемы. При наличии нескольких проблем сервис будет иметь статус наиболее серьёзной из них. Статус родительского сервиса затем рассчитывается на основе статусов дочерних сервисов в соответствии с правилами расчёта статуса.

Если указано несколько тегов, используется логика *AND*: проблема должна иметь все теги, указанные в настройке сервиса, чтобы быть сопоставленной с сервисом.



Note:
Проблема в Zabbix наследует теги по всей цепочке шаблонов, узлов сети, элементов данных, веб-сценариев и триггеров. Любой из этих тегов может использоваться для сопоставления проблем с сервисами.

Пример:

Проблема *Web camera 3 is down* имеет теги `type:video-surveillance`, `floor:1` и `name:webcam-3`, а также статус *Warning*

У сервиса **Web camera 3** указан только один 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:
Описанные ниже кнопки видимы только тогда, когда раздел *Сервисы* находится в рех

Чтобы изменить существующий сервис, нажмите на значок карандаша рядом с сервисом.

Чтобы клонировать существующий сервис, нажмите на 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 на значок x рядом с ним. При удалении родительского сервиса его дочерние сервисы не будут удалены и переместятся на один уровень выше в дереве сервисов (дочерние сервисы 1-го уровня получат тот же уровень, что и удалённый родительский сервис).

Две кнопки под списком сервисов предлагают несколько вариантов массового редактирования:

- *Mass update* - массовое обновление свойств сервисов
- *Delete* - удаление сервисов

Чтобы использовать эти параметры, установите флажки перед соответствующими сервисами, затем нажмите нужную кнопку.

2 SLA

Обзор После создания услуг вы можете начать мониторинг их производительности в соответствии с соглашением об уровне услуг (SLA).

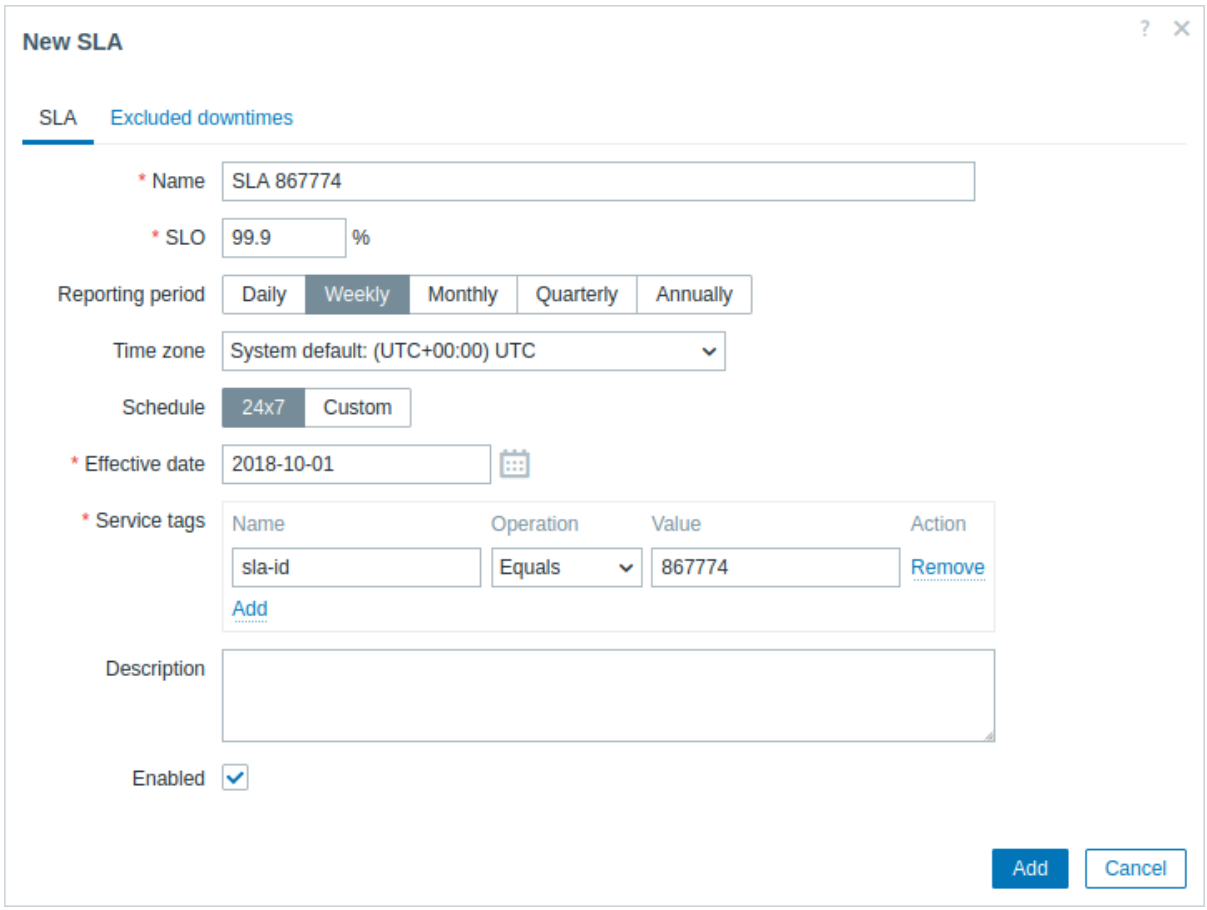
Раздел меню *Услуги->SLA* позволяет настроить SLA по различным услугам. SLA в Zabbix определяет цель уровня обслуживания (SLO), расписание ожидаемого времени работы и запланированного времени простоя.

SLA и услуги сопоставляются при помощи тегов услуг. К нескольким услугам можно применить один и тот же SLA - производительность будет измеряться по каждой соответствующей услуге отдельно. Одна услуга может иметь несколько назначенных SLA - данные по каждому SLA будут отображаться отдельно.

В отчетах SLA Zabbix отображает данные Индикатора уровня услуги (SLI), который замеряет действительную доступность услуги. Соответствие услуги целевому показателю SLA определяется сравнением SLO (ожидаемая доступность в %) с SLI (действительная доступность в %).

Конфигурация Чтобы создать новый SLA, нажмите кнопку *Create SLA*.

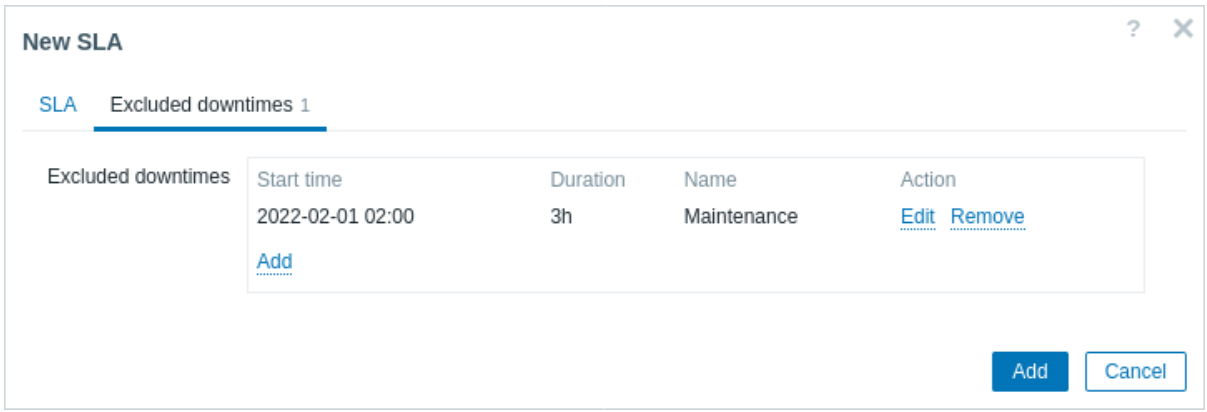
Вкладка **SLA** позволяет указать общие параметры SLA.



Параметр	Описание
Name	Введите имя SLA.

Параметр	Описание
SLO	Введите целевой уровень сервиса (SLO) в процентах.
Reporting period	Выберите отчетный период SLI, используемый для расчета данных SLI в отчете 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 (янв–мар), Q2 (апр–июн), Q3 (июл–сен), Q4 (окт–дек). **Ежегодно** — каждый календарный год, с 1 января 00:00:00 до 31 декабря 23:59:59.
Time zone	Выберите часовой пояс SLA.
Schedule	Выберите расписание SLA — 24x7 или пользовательское.
Effective date	Выберите дату начала расчета SLA.
Service tags	Добавьте теги сервиса для определения сервисов, к которым должен применяться этот SLA. **Name** — имя тега сервиса, должно точно совпадать, с учетом регистра. **Operation** — выберите *Equals*, если значение тега должно точно совпадать (с учетом регистра), или *Contains*, если должна совпадать часть значения тега (без учета регистра). **Value** — значение тега сервиса для поиска в соответствии с выбранной операцией. SLA применяется к сервису, если совпадает хотя бы один тег сервиса.
Description	Добавьте описание для SLA.
Enabled	Установите флажок, чтобы включить расчет SLA.

Вкладка **Excluded downtimes** позволяет указать периоды недоступности, которые исключаются из расчета SLA.



Нажмите *Add*, чтобы настроить исключаемые периоды недоступности, затем введите имя периода, дату начала и длительность.

Отчеты SLA Насколько хорошо услуга работает по сравнению с SLA, видно в отчете SLA. Отчеты SLA можно увидеть:

- из раздела *SLA* при нажатии на гиперссылку отчета SLA;
- из раздела *Услуги* при нажатии на имя SLA на вкладке инфо;
- в виджете панели *Отчет SLA*.

Как только SLA будет настроен, на вкладке *Инфо* в разделе с услугами также будет отображаться некоторая информация о производительности услуги.

4 Пример настройки

Обзор В этом разделе описана простая настройка мониторинга кластера высокой доступности Zabbix как услуги.

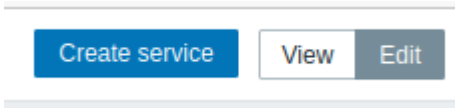
Предварительные требования Перед настройкой мониторинга сервиса необходимо, чтобы были настроены узлы сети:

- *HA node 1* как минимум с одним триггером и тегом (предпочтительно заданным на уровне триггера) `component:ha-node-1`
- *HA node 2* как минимум с одним триггером и тегом (предпочтительно заданным на уровне триггера) `component:ha-node-2`

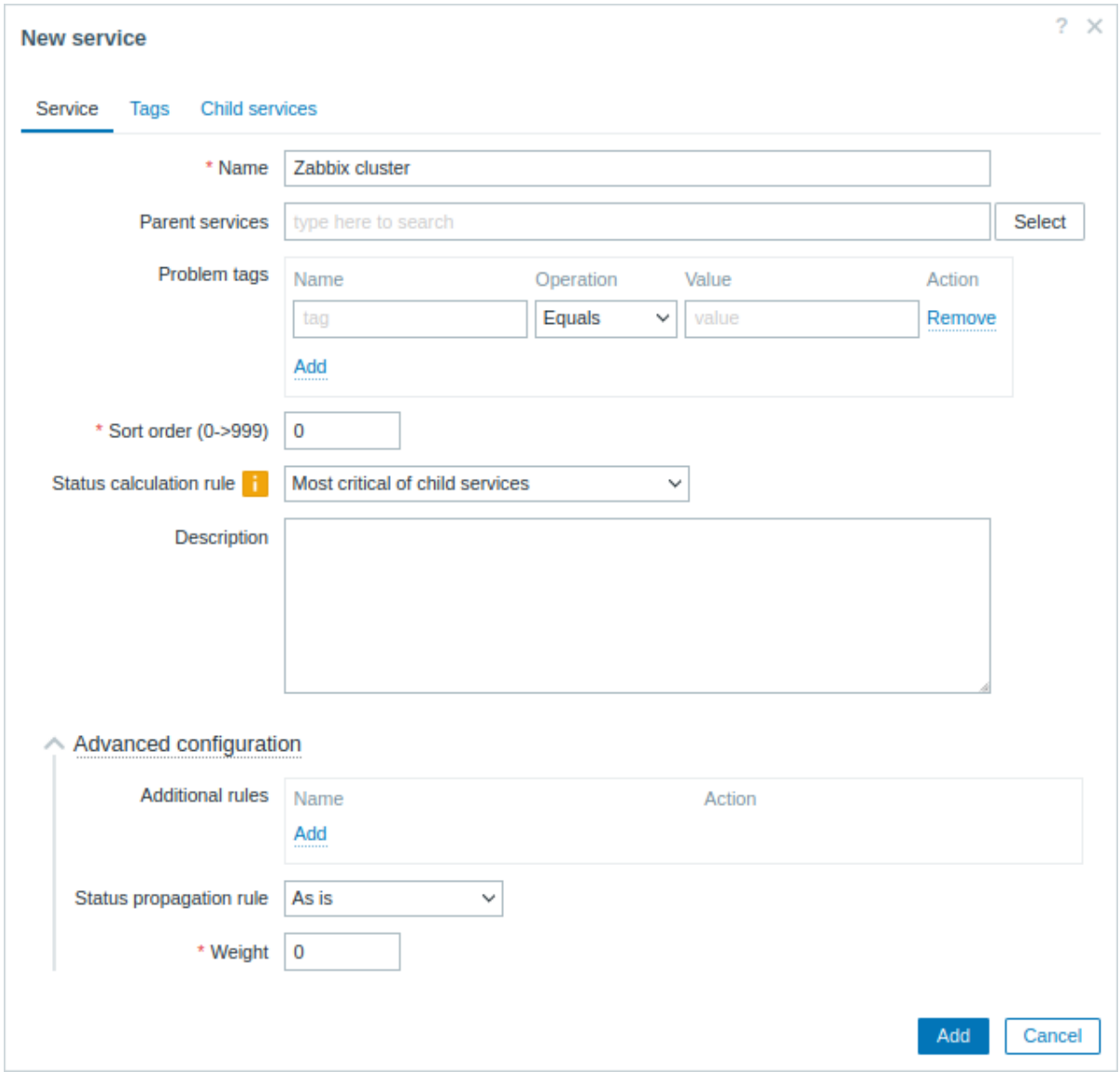
Дерево сервисов Следующий шаг — построить дерево сервисов. В этом примере инфраструктура очень простая и состоит из трёх сервисов: *Zabbix cluster* (родительский) и двух дочерних сервисов *Zabbix server node 1* и *Zabbix server node 2*.

```
Zabbix cluster
|
|- Zabbix server node 1
|- Zabbix server node 2
```

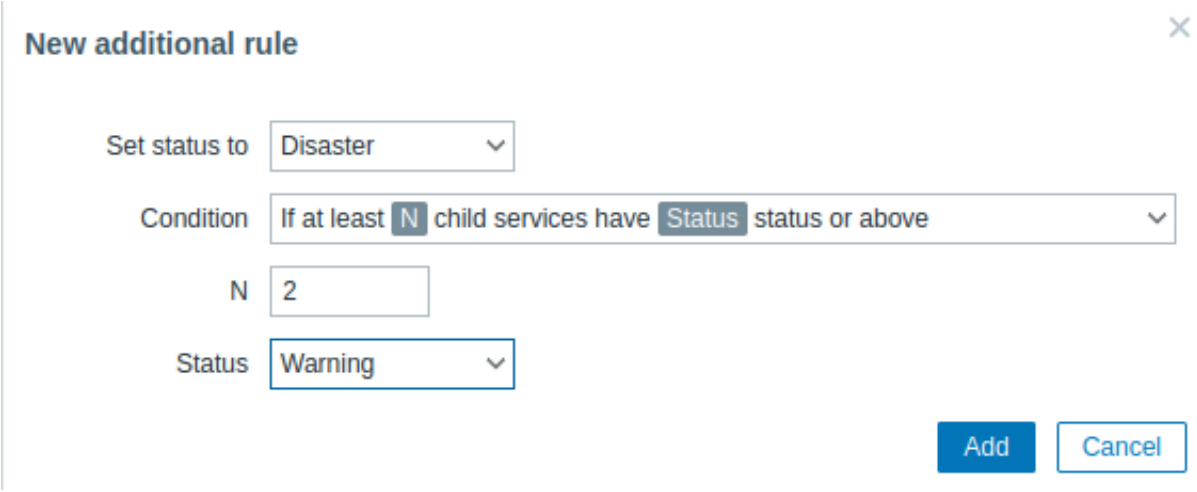
На странице Services включите режим *Edit* и нажмите Create service:



В окне настройки сервиса введите имя *Zabbix cluster* и нажмите на метку *Advanced configuration*, чтобы отобразить дополнительные параметры настройки.



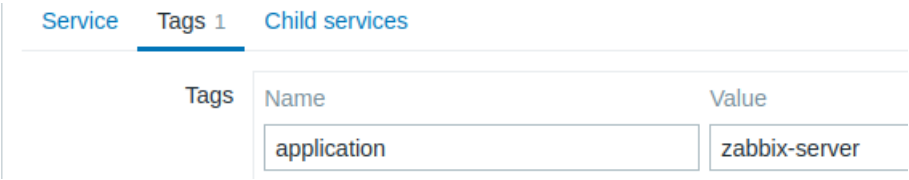
Настройте дополнительное правило:



У сервиса Zabbix cluster будет два дочерних сервиса — по одному для каждого HA-узла. Если у обоих HA-узлов есть проблемы как минимум со статусом *Warning*, статус родительского сервиса должен быть установлен в *Disaster*. Для этого дополнительное правило должно быть настроено следующим образом:

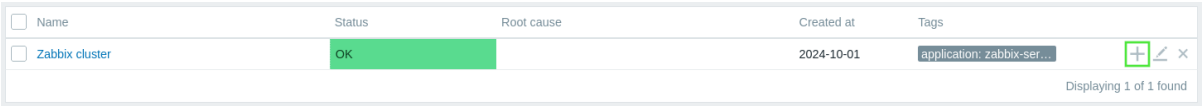
- Set status to: Disaster
- Condition: If at least `N` child services have `Status` status or above
- N: 2
- Status: Warning

Перейдите на вкладку *Tags* и добавьте тег `application:zabbix-server`. Этот тег будет использоваться позже для действий сервиса и отчётов SLA.



Сохраните новый сервис.

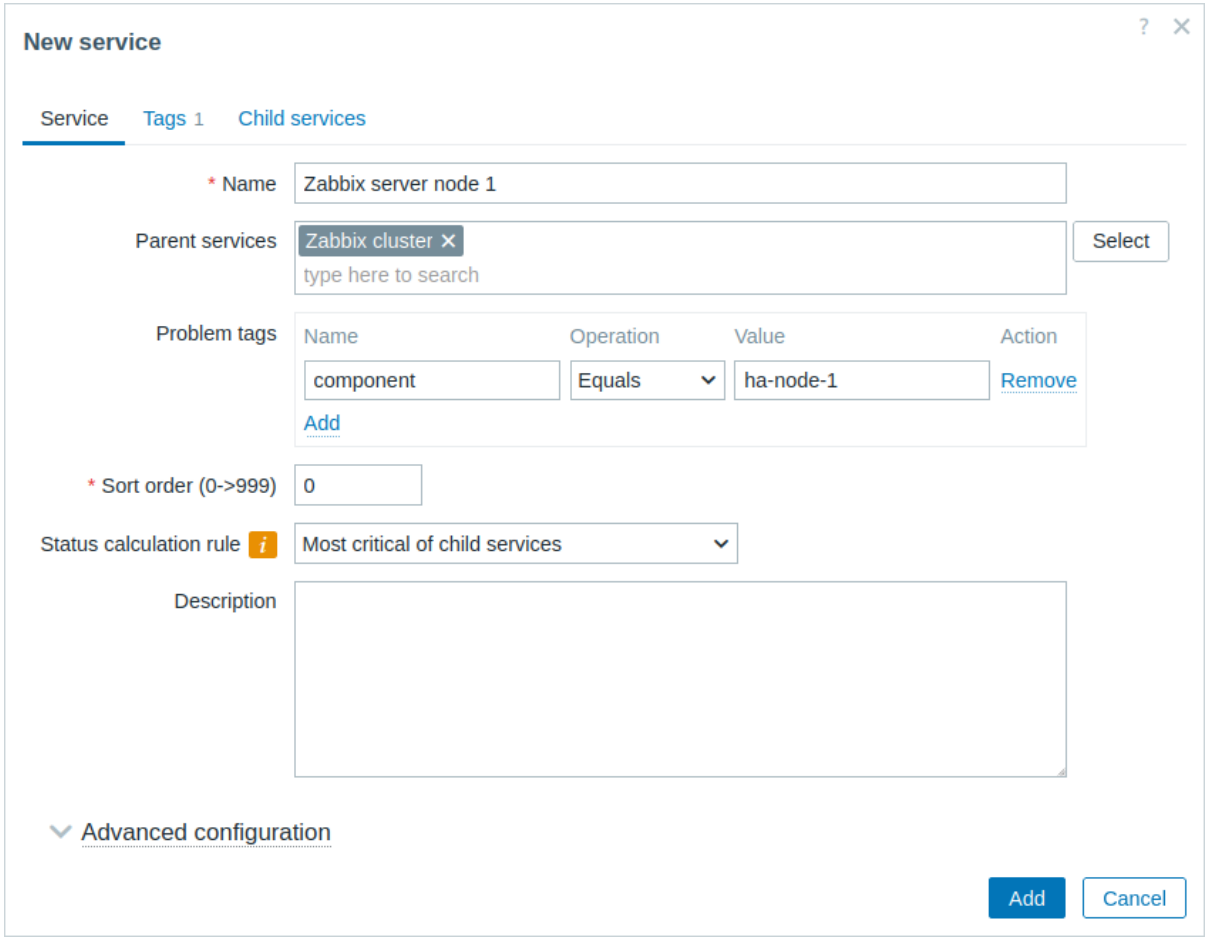
Чтобы добавить дочерний сервис, нажмите на значок плюса рядом с сервисом *Zabbix cluster* (значок виден только в режиме *Edit*).



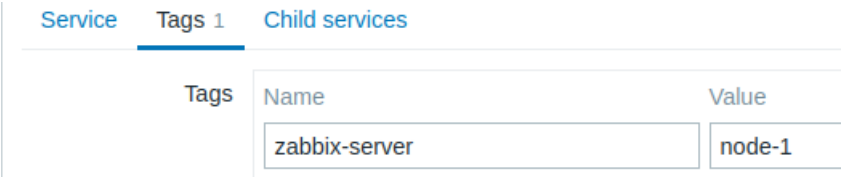
В окне настройки сервиса введите имя *Zabbix server node 1*. Обратите внимание, что параметр Parent services уже предварительно заполнен значением *Zabbix cluster*.

Доступность этого сервиса зависит от проблем на узле сети *HA node 1*, помеченных тегом проблемы `component:ha-node-1`. В параметре Problem tags введите:

- Name: component
- Operation: Equals
- Value: ha-node-1



Переключитесь на вкладку *Tags* и добавьте тег сервиса: `zabbix-server:node-1`. Этот тег будет использоваться позже для действий сервиса и отчетов SLA.



Сохраните новый сервис.

Создайте ещё один дочерний сервис кластера Zabbix с именем ”Zabbix server node 2”.

Установите теги проблем следующим образом:

- Имя: component
- Операция: Равно
- Значение: ha-node-2

Перейдите на вкладку *Теги* и добавьте тег сервиса: `zabbix-server:node-2`.

Сохраните новый сервис.

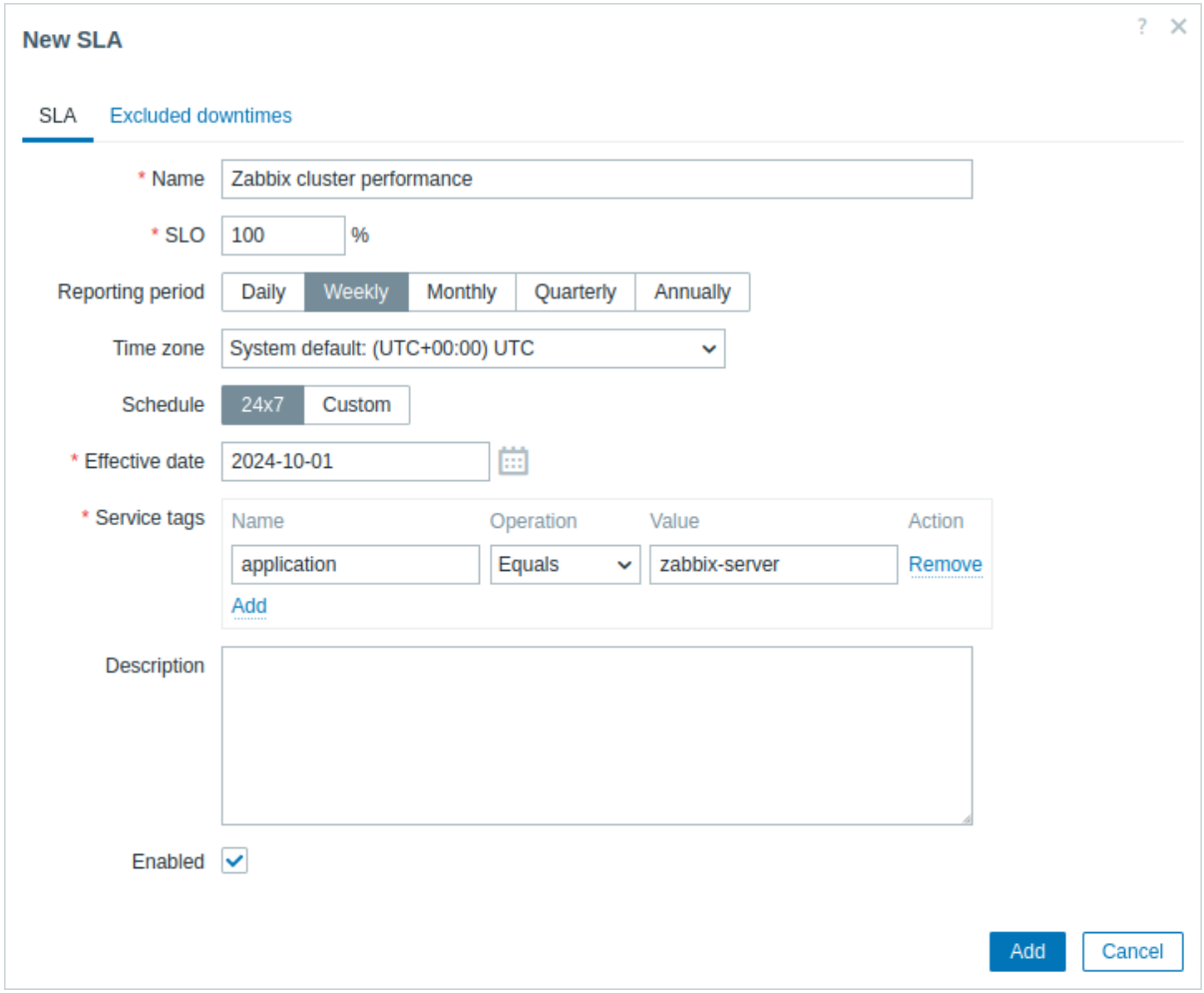
SLA В этом примере ожидаемая производительность кластера Zabbix составляет 100%, исключая полугодовой период обслуживания продолжительностью один час.

Сначала необходимо добавить новое соглашение об уровне обслуживания.

Перейдите в раздел меню *Services->SLA* и нажмите Create SLA. Введите имя *Zabbix cluster performance* и установите SLO равным 100%.

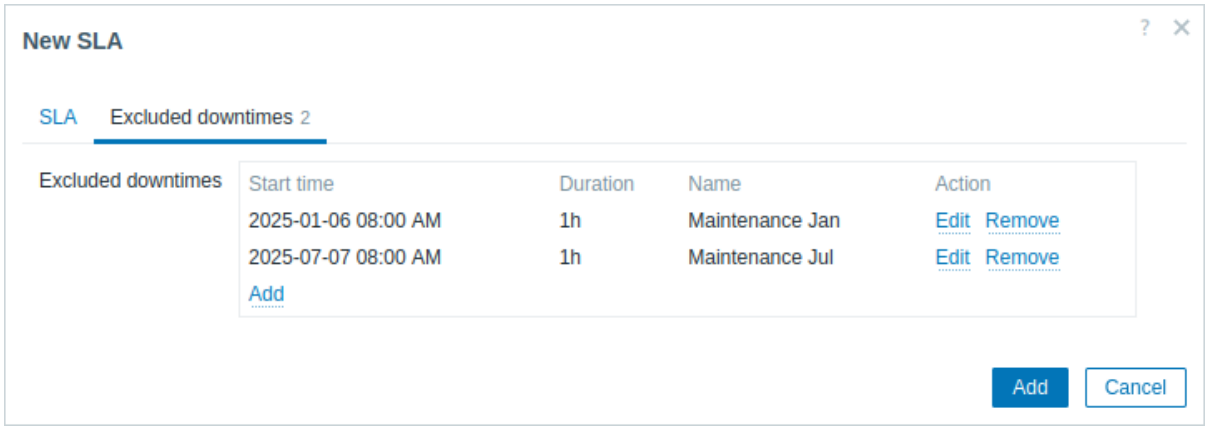
Сервис Zabbix cluster имеет тег сервиса `application:zabbix-server`. Чтобы использовать это SLA для измерения производительности кластера Zabbix, в параметре *Service tags* укажите:

- Имя: application
- Операция: Equals
- Значение: zabbix-server



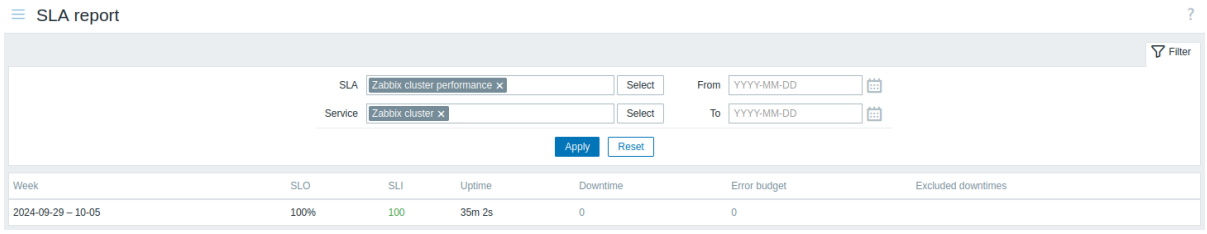
В реальной среде вы также можете изменить желаемый отчетный период, часовой пояс и дату начала или изменить расписание с 24/7 на пользовательское. Для этого примера достаточно настроек по умолчанию.

Перейдите на вкладку *Excluded downtimes* и добавьте периоды недоступности для запланированных периодов обслуживания, чтобы исключить эти периоды из расчета SLA. В разделе Excluded downtimes нажмите ссылку Add, введите имя периода недоступности, планируемое время начала и длительность.

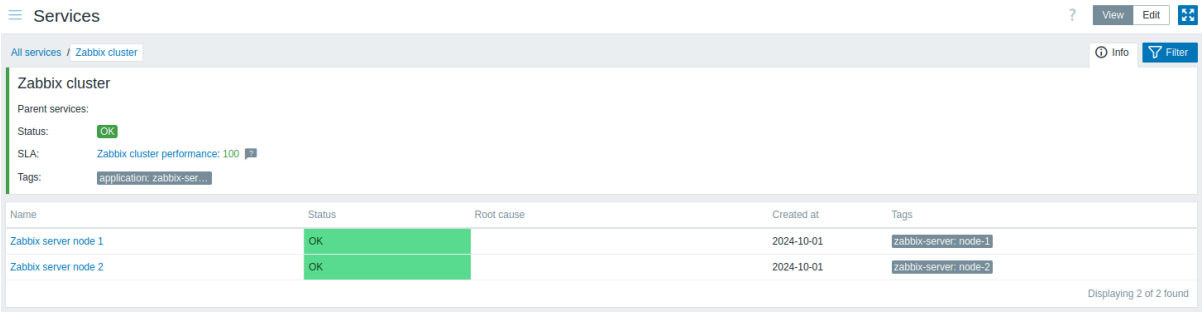


Нажмите Add, чтобы сохранить новое SLA.

Перейдите в раздел отчетов SLA, чтобы просмотреть отчет SLA для Zabbix cluster.



Информацию SLA также можно проверить в разделе *Services*.



7 Веб-мониторинг

Обзор С помощью Zabbix можно проверять несколько аспектов доступности веб-сайтов.

> **Attention:**
> Для выполнения веб-мониторинга сервер Zabbix должен быть предварительно настроен с поддержкой cURL (libcurl).

Чтобы активировать веб-мониторинг, необходимо определить веб-сценарии. Веб-сценарий состоит из одного или нескольких HTTP-запросов, или «шагов». Шаги периодически выполняются сервером Zabbix в заранее определённом порядке. Если узел сети мониторится 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 могут быть созданы на уровне шаблона, а затем применены сразу к нескольким узлам сети.

Следующая информация собирается в любом веб-сценарии:

- средняя скорость загрузки в секунду для всех шагов всего сценария
- номер шага, на котором произошёл сбой
- последнее сообщение об ошибке

Следующая информация собирается на каждом шаге веб-сценария:

- скорость загрузки в секунду
- время отклика
- код ответа

Подробнее см. в разделе элементы данных веб-мониторинга.

Данные, собранные при выполнении веб-сценариев, хранятся в базе данных. Эти данные автоматически используются для графиков, триггеров и уведомлений.

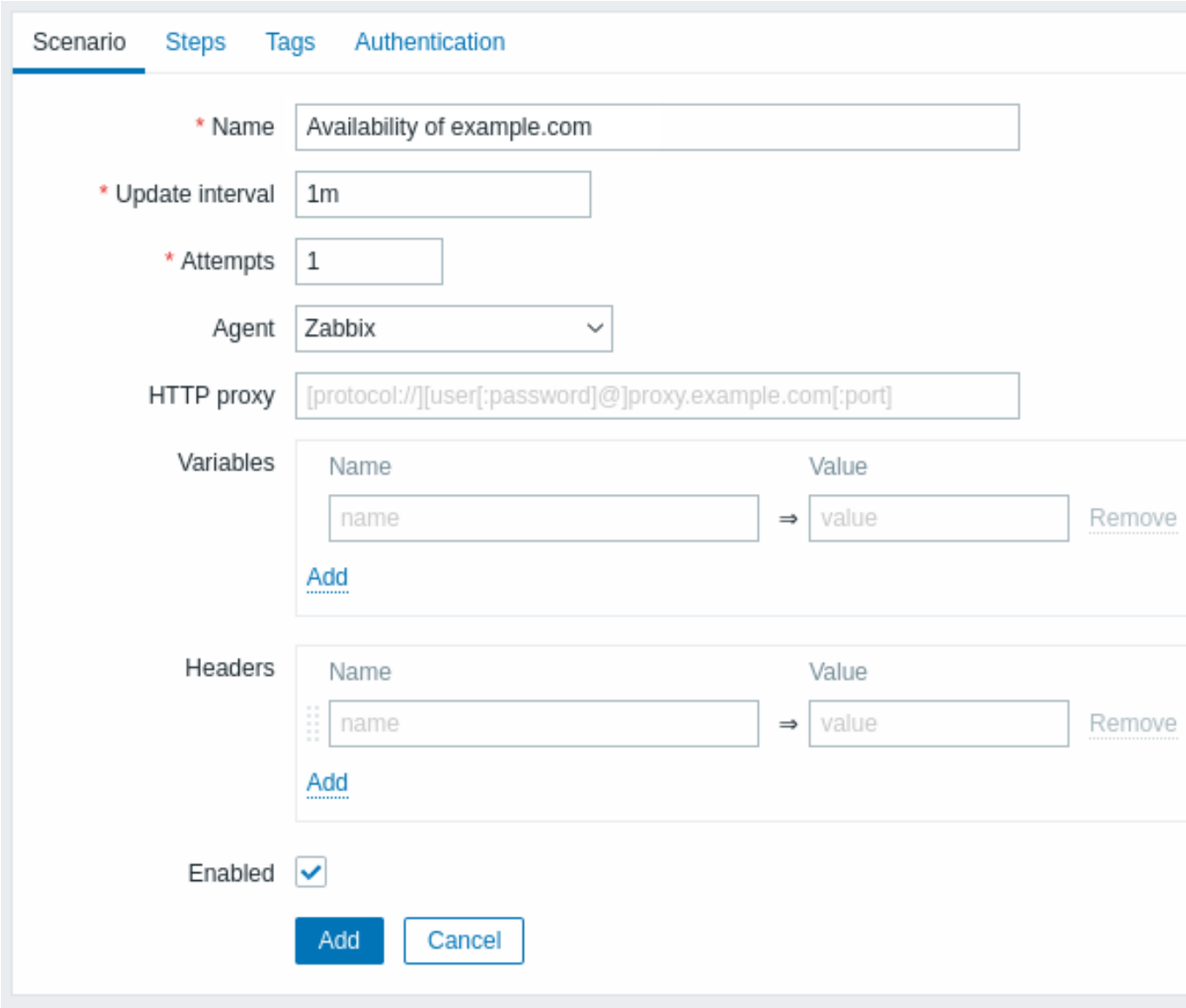
Zabbix также может проверять, содержит ли полученная HTML-страница заранее определённую строку. Он может выполнять имитацию входа в систему и следовать по пути имитируемых щелчков мышью на странице.

Веб-мониторинг Zabbix поддерживает как HTTP, так и HTTPS. При выполнении веб-сценария Zabbix при необходимости будет следовать перенаправлениям (см. параметр *Follow redirects* ниже). Максимальное количество перенаправлений жёстко задано и равно 10 (с использованием параметра cURL CURLOPT_MAXREDIRS). Все cookie-файлы сохраняются в течение выполнения одного сценария.

Настройка веб-сценария Чтобы настроить веб-сценарий:

- Перейдите в: *Сбор данных* → *Узлы сети* (или *Шаблоны*)
- Нажмите *Web* в строке узла сети/шаблона
- Нажмите *Создать веб-сценарий* справа (или на имя сценария, чтобы изменить существующий сценарий)
- Введите параметры сценария в форме

Вкладка **Сценарий** позволяет настроить общие параметры веб-сценария.



Все обязательные поля ввода отмечены красной звёздочкой.

Параметры сценария:

Параметр	Описание
Имя	Уникальное имя сценария. Поддерживаются пользовательские макросы. *Обратите внимание*, что если используются пользовательские макросы, эти макросы останутся неразрешёнными 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 приведённом примере — позже). Новые веб-сценарии будут проверены в течение 60 секунд после их создания.
Попытки	Количество попыток выполнения шагов веб-сценария. В случае проблем с сетью (тайм-аут, отсутствие соединения и т. д.) Zabbix может повторить выполнение шага несколько раз. Указанное значение одинаково влияет на каждый шаг сценария. Можно указать до 10 попыток, значение по умолчанию — 1. *Обратите внимание*: Zabbix не будет повторять шаг из-за неверного кода ответа или несоответствия требуемой строки.
Агент	Выберите клиентский агент. Zabbix будет представляться выбранным браузером. Это полезно, когда веб-сайт возвращает разное содержимое для разных браузеров. В этом поле можно использовать пользовательские макросы.

Параметр	Описание
HTTP proxy	Можно указать HTTP-прокси для использования в формате `[protocol://][username[:password]@]proxy.example.com[:port]`. Это задаёт параметр cURL CURLOPT_PROXY. Необязательный префикс `protocol://` можно использовать для указания альтернативных протоколов прокси (поддержка префикса протокола была добавлена в cURL 7.21.7). Если протокол не указан, прокси будет рассматриваться как HTTP-прокси. По умолчанию будет использоваться порт 1080. Если прокси указан, он переопределит связанные с прокси переменные окружения, такие как http_proxy, HTTPS_PROXY. Если не указан, прокси не будет переопределять связанные с прокси переменные окружения. Введённое значение передаётся «как есть», без какой-либо проверки корректности. Также можно указать адрес SOCKS-прокси. Если указан неверный протокол, соединение завершится ошибкой, и элемент данных станет неподдерживаемым. *Обратите внимание*, что для HTTP-прокси поддерживается только простая аутентификация. В этом поле можно использовать пользовательские макросы.
Переменные	Переменные, которые могут использоваться в шагах сценария (URL, post-переменные). 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 переменным с `regex:` требуется один шаг для получения значения регулярного выражения, поэтому извлечённое значение можно применить только на следующем шаге. Если часть значения начинается с `regex:`, то следующая за ней часть рассматривается как регулярное выражение, которое ищет совпадение на веб-странице и, если оно найдено, сохраняет его в переменной. Должна присутствовать как минимум одна подгруппа, чтобы можно было извлечь совпавшее значение. Поддерживаются пользовательские макросы и макросы {HOST.*}. Переменные автоматически URL-кодируются при использовании в полях запроса или данных формы для post-переменных, но должны быть URL-кодированы вручную при использовании в raw post или непосредственно в URL.

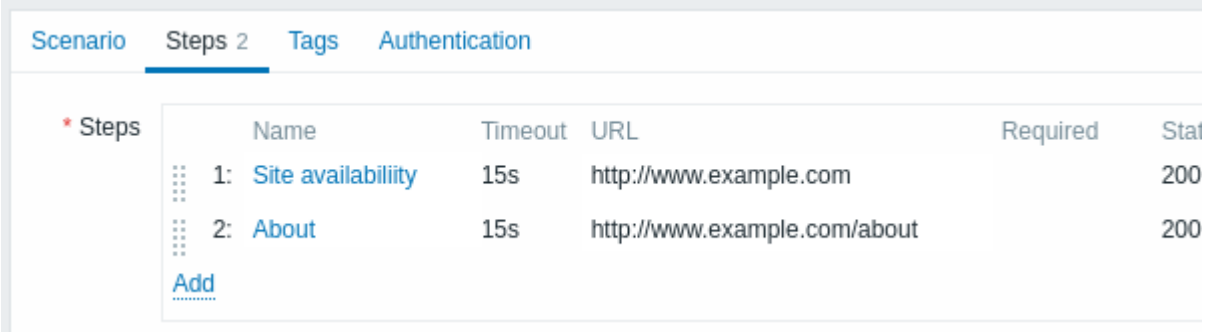
Параметр	Описание
Заголовки	HTTP-заголовки используются при выполнении запроса. Можно использовать заголовки по умолчанию и пользовательские заголовки. Заголовки будут назначаться с использованием настроек по умолчанию в зависимости от типа агента, выбранного из выпадающего списка на уровне сценария, и будут применяться ко всем шагам, если только они не определены как пользовательские на уровне шага. **Следует отметить, что определение заголовка на уровне шага автоматически отбрасывает все ранее определённые заголовки, кроме заголовка по умолчанию, который назначается выбором 'User-Agent' из выпадающего списка на уровне сценария.** Однако даже заголовок по умолчанию 'User-Agent' можно переопределить, указав его на уровне шага. Чтобы отменить заголовок на уровне сценария, заголовок должен быть указан по имени и без значения на уровне шага. Заголовки должны быть перечислены с использованием того же синтаксиса, в котором они появляются в HTTP-протоколе, при необходимости с использованием некоторых дополнительных возможностей, поддерживаемых параметром cURL CURLOPT_HTTPHEADER. Например: Accept-Charset=utf-8 Accept-Language=en-US Content-Type=application/xml; charset=utf-8 Поддерживаются пользовательские макросы и макросы {HOST.*}.
Включён	Сценарий активен, если этот флажок установлен, в противном случае — отключён.

Обратите внимание, что при редактировании существующего сценария в форме доступны две дополнительные кнопки:

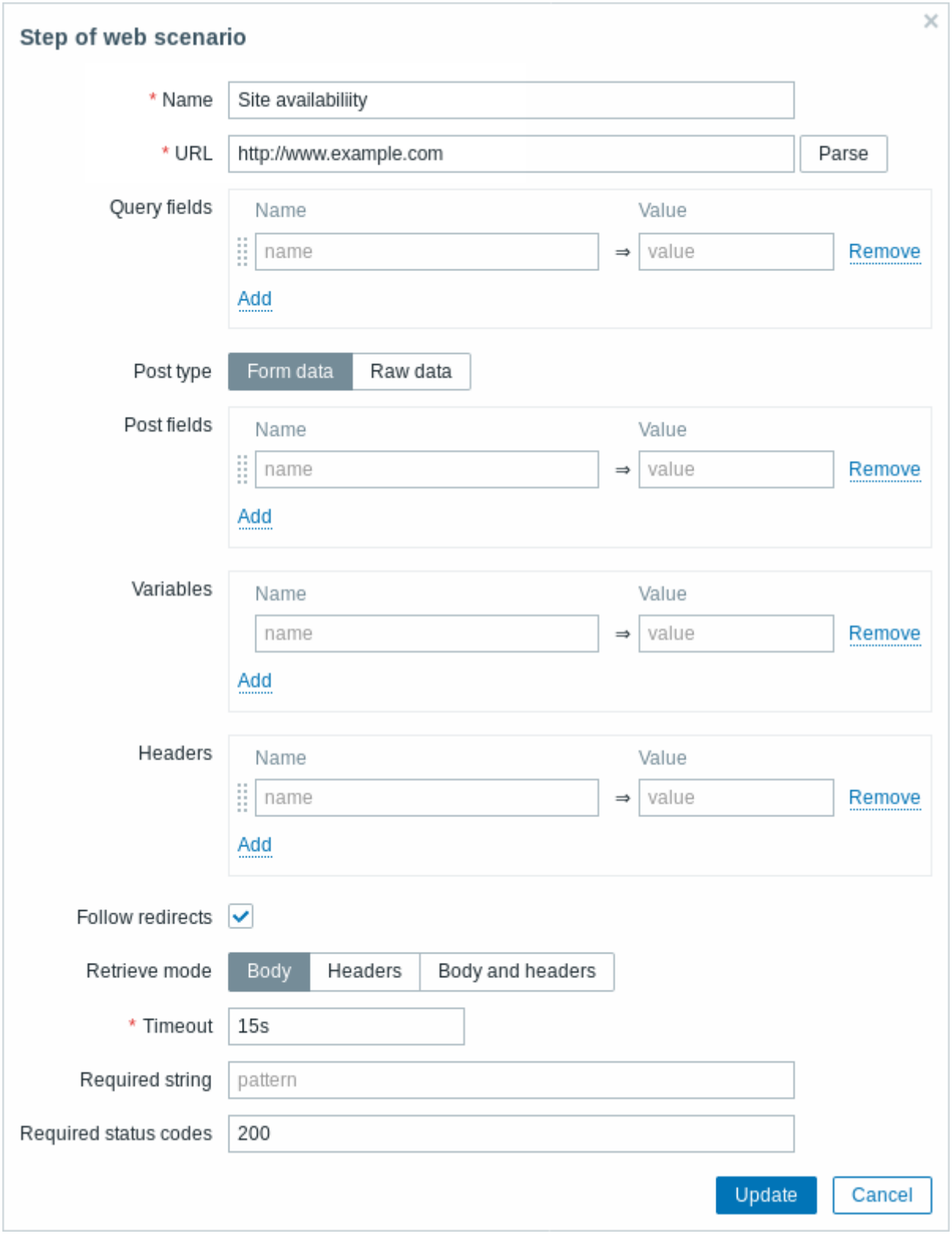
Clone	Создать другой сценарий на основе свойств существующего.
Clear history and trends	Удалить данные истории и трендов для сценария. После удаления данных сервер немедленно выполнит сценарий.

Note:
Если поле *HTTP proxy* оставлено пустым, другим способом использования HTTP-прокси является настройка связанных с прокси переменных окружения.
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`. Более подробная информация доступна при выполнении команды 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, так как они будут разрешены в "******".



Настройка шагов

Параметры шага:

Параметр	Описание
Имя	Уникальное имя шага. Поддерживаются пользовательские макросы. *Обратите внимание*, что если используются пользовательские макросы, эти макросы останутся неразрешёнными в именах элементов данных веб-мониторинга.

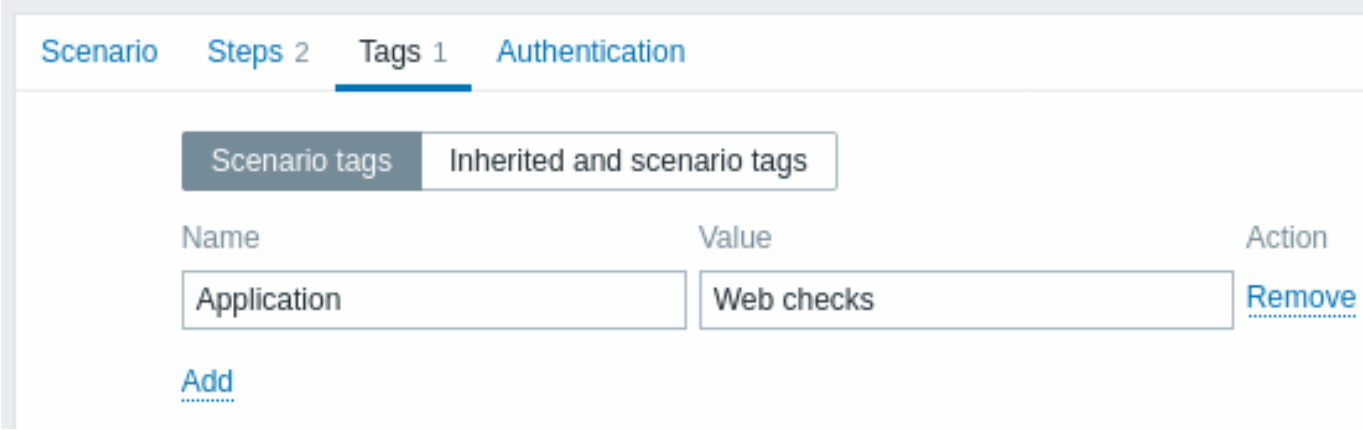
Параметр	Описание
URL	URL для подключения и получения данных. Например: https://www.example.com http://www.example.com/download Доменные имена могут быть указаны символами Unicode. При выполнении шага веб-сценария они автоматически преобразуются в punycode и затем в ASCII. Кнопка *Parse* может использоваться для отделения необязательных полей запроса (например, ?name=Admin&password=mypassword) от URL, перемещая атрибуты и значения в *Поля запроса* для автоматического URL-кодирования. В URL можно использовать переменные, применяя синтаксис {macro}. Переменные можно URL-кодировать вручную, используя синтаксис {{macro}.urlencode()}. Поддерживаются пользовательские макросы и макросы {HOST.*}. Ограничено 2048 символами.
Поля запроса	Переменные HTTP GET для URL. Указываются как пары атрибут-значение. Значения автоматически URL-кодируются. Значения из переменных сценария, пользовательских макросов или макросов {HOST.*} сначала разрешаются, а затем автоматически URL-кодируются. Использование синтаксиса {{macro}.urlencode()} приведёт к двойному URL-кодированию. Поддерживаются пользовательские макросы и макросы {HOST.*}.
Post	Переменные HTTP POST. В режиме **Form data** указываются как пары атрибут-значение. Значения автоматически URL-кодируются. Значения из переменных сценария, пользовательских макросов или макросов {HOST.*} сначала разрешаются, а затем автоматически URL-кодируются. В режиме **Raw data** атрибуты/значения отображаются в одной строке и объединяются символом **&**. Необработанные значения можно URL-кодировать/декодировать вручную, используя синтаксис {{macro}.urlencode()} или {{macro}.urldecode()}. Например: id=2345&userid={user} Если {user} определён как переменная веб-сценария, при выполнении шага он будет заменён своим значением. Если вы хотите URL-кодировать переменную, замените {user} на {{user}.urlencode()}. Поддерживаются пользовательские макросы и макросы {HOST.*}.
Переменные	Переменные уровня шага, которые могут использоваться для функций GET и POST. Указываются как пары атрибут-значение. Переменные уровня шага переопределяют переменные уровня сценария или переменные из предыдущего шага. Однако значение переменной уровня шага влияет только на следующий шаг (а не на текущий шаг). Они имеют следующий формат: **{macro}**=value **{macro}**=regex:<regular expression> Дополнительную информацию смотрите в описании переменных на уровне сценария. Переменные автоматически URL-кодируются при использовании в полях запроса или данных формы для post-переменных, но должны URL-кодироваться вручную при использовании в raw post или непосредственно в URL.
Заголовки	Пользовательские HTTP-заголовки, которые будут отправлены при выполнении запроса. Указываются как пары атрибут-значение. Заголовок, определённый на уровне шага, будет использоваться для этого конкретного шага. **Следует отметить, что определение заголовка на уровне шага автоматически отбрасывает все ранее определённые заголовки, кроме заголовка по умолчанию, который назначается выбором 'User-Agent' из выпадающего списка на уровне сценария.** Однако даже заголовок по умолчанию 'User-Agent' может быть переопределён, если указать его на уровне шага. Например, если задать имя заголовка, но не указывать значение, это снимет заголовок по умолчанию, заданный на уровне сценария. Поддерживаются пользовательские макросы и макросы {HOST.*}. Это задаёт параметр cURL CURLOPT_HTTPHEADER.
Следовать перенаправлениям	Установите флажок, чтобы следовать HTTP-перенаправлениям. Это задаёт параметр cURL CURLOPT_FOLLOWLOCATION.

Параметр	Описание
Режим получения	Выберите режим получения: **Body** — получать только тело из HTTP-ответа **Headers** — получать только заголовки из HTTP-ответа **Body and headers** — получать тело и заголовки из HTTP-ответа
Тайм-аут	Zabbix не будет тратить на обработку URL больше заданного времени (от одной секунды до максимум 1 часа). Фактически этот параметр определяет максимальное время для установления соединения с URL и максимальное время для выполнения HTTP-запроса. Следовательно, Zabbix не будет тратить на шаг более **2 x Timeout** секунд. Поддерживаются суффиксы времени, например 30s, 1m, 1h. Поддерживаются пользовательские макросы.
Обязательная строка	Обязательный шаблон регулярного выражения. Если полученное содержимое (HTML) не соответствует обязательному шаблону, шаг завершится ошибкой. Если поле пустое, проверка обязательной строки не выполняется. Например: Homepage of Zabbix Welcome.*admin *Примечание*: Ссылки на регулярные выражения, созданные в веб-интерфейсе Zabbix, в этом поле не поддерживаются. Поддерживаются пользовательские макросы и макросы {HOST.*}.
Обязательные коды состояния	Список ожидаемых HTTP-кодов состояния. Если Zabbix получит код, которого нет 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.

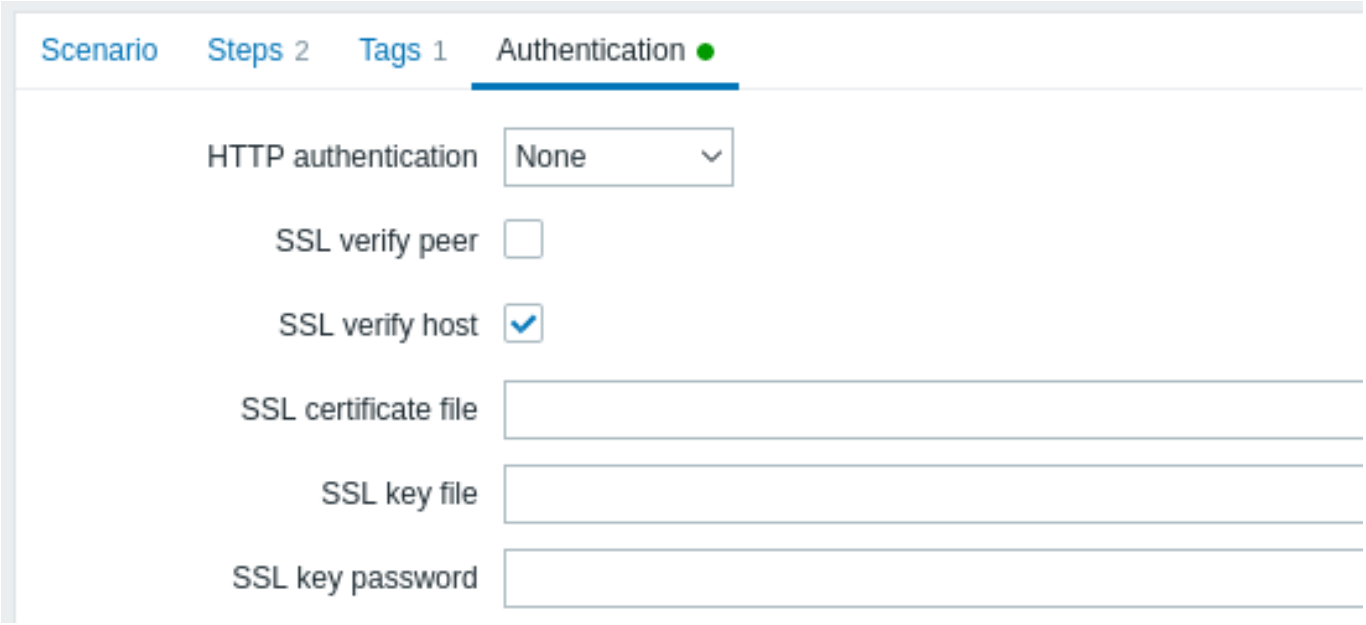
См. также практический пример того, как можно настроить шаги веб-мониторинга.

Настройка тегов Вкладка **Теги** позволяет задать теги уровня 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 на то, что включен какой-либо тип HTTP-аутентификации.



Параметры аутентификации:

Параметр	Описание
HTTP authentication	Выберите вариант аутентификации: **None** — аутентификация не используется; **Basic** — используется базовая аутентификация; **NTLM** — используется аутентификация NTLM (Windows NT LAN Manager); **Kerberos** — используется аутентификация Kerberos (см. также: Настройка Kerberos в Zabbix); **Digest** — используется аутентификация Digest.
User	Введите имя пользователя (до 255 символов). Это поле доступно, если для *HTTP authentication* выбрано Basic, NTLM, Kerberos или Digest. Поддерживаются пользовательские макросы.
Password	Введите пароль пользователя (до 255 символов). Это поле доступно, если для *HTTP authentication* выбрано Basic, NTLM, Kerberos или Digest. Поддерживаются пользовательские макросы.
SSL verify peer	Установите флажок, чтобы проверять SSL-сертификат веб-сервера. Сертификат сервера будет автоматически взят из общесистемного расположения центра сертификации (CA). Вы можете переопределить расположение файлов CA с помощью параметра конфигурации сервера или прокси Zabbix SSLCALocation. Устанавливает параметр cURL CURLOPT_SSL_VERIFYPEER.
SSL verify host	Установите флажок, чтобы проверять, что поле *Common Name* или поле *Subject Alternate Name* сертификата веб-сервера совпадает. Устанавливает параметр cURL CURLOPT_SSL_VERIFYHOST.
SSL certificate file	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 *SSL key file* пустым. Если ключ зашифрован, укажите пароль в поле *SSL key password*. 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.
SSL key file	Имя файла закрытого ключа SSL, 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.
SSL key password	Пароль файла закрытого ключа SSL. В этом поле можно использовать пользовательские макросы. Устанавливает параметр cURL CURLOPT_KEYPASSWD.

Attention:
[1] Zabbix поддерживает только файлы сертификатов и закрытых ключей в формате PEM. Если ваши данные сертификата и закрытого ключа находятся в файле формата PKCS #12 (обычно с расширением *.p12 или *.pfx), вы можете создать из него PEM-файл 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 в сертификатах без перезапуска.

Note:
Если сертификат клиента и закрытый ключ находятся в одном файле, просто укажите его в поле "SSL certificate file", а поле "SSL key file" оставьте пустым. Сертификат и ключ по-прежнему должны быть в формате PEM. Объединить сертификат и ключ легко:

```
cat client.crt client.key > client.pem
```

Отображение Чтобы просмотреть веб-сценарии, настроенные для узла сети, перейдите в *Мониторинг → Узлы сети*, найдите узел сети в списке и нажмите на гиперссылку *Веб* в последнем столбце. Нажмите на имя сценария, чтобы получить подробную информацию.



Обзор веб-сценариев также можно отобразить на *Панелях* с помощью виджета веб-мониторинга.

Последние результаты выполнения веб-сценария доступны в разделе *Мониторинг → Последние данные*.

Расширенный мониторинг Иногда необходимо записывать в журнал содержимое полученной HTML-страницы. Это особенно полезно, если какой-либо шаг веб-сценария завершается ошибкой. Для этой цели используется 5-й уровень отладки (trace). Этот уровень можно задать в конфигурационных файлах сервера и прокси или с помощью параметра run-time control (`-R log_level_increase="http poller,N"`, где N — номер процесса). Следующие примеры показывают, как можно запустить расширенный мониторинг, если уже установлен 4-й уровень отладки:

```
#                          http poller:
zabbix_server -R log_level_increase="http poller"

#                           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`.

1 Элементы данных веб-мониторинга

Обзор

Некоторые элементы данных добавляются автоматически для выполнения мониторинга при создании веб-сценариев.

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
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)*
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)*
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:
Элементы данных веб-мониторинга добавляются с 30 дневным периодом хранения истории и с 90 дневным периодом хранения динамики изменений.

Note:
Если имя сценария начинается с двойной кавычки, содержит запятую или квадратные скобки, оно будет корректно заключено в кавычки в ключах элементов данных. В других случаях дополнительные кавычки не добавляются.

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(/host/web.test.fail[Scenario])<>0
```

Не забудьте заменить ’Сценарий’ на реальное название вашего сценария.

Пример 2

Для создания триггера ”Проверка веб сценария завершилась ошибкой”, вы можете указать триггер с именем:

```
"Scenario"          : {ITEM.VALUE}
```

и выражением триггера:

```
length(last(/host/web.test.error[Scenario]))>0 and last(/host/web.test.fail[Scenario])>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(/host/web.test.in[Scenario,,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
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)*
Время ответа для шага <Шаг> сценария <Сценарий>	Этот элемент данных собирает информацию о времени ответа в секундах для указанного шага. Временем 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)*
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, содержит запятые или 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 настройке условий оповещения. Например для создания триггера ”Вход в Zabbix веб-интерфейс слишком медленный”, вы можете указать следующее выражение триггера:

```
last(/zabbix/web.test.time[ZABBIX GUI,Login,resp])>3
```

2 Сценарий из реальной жизни

Обзор

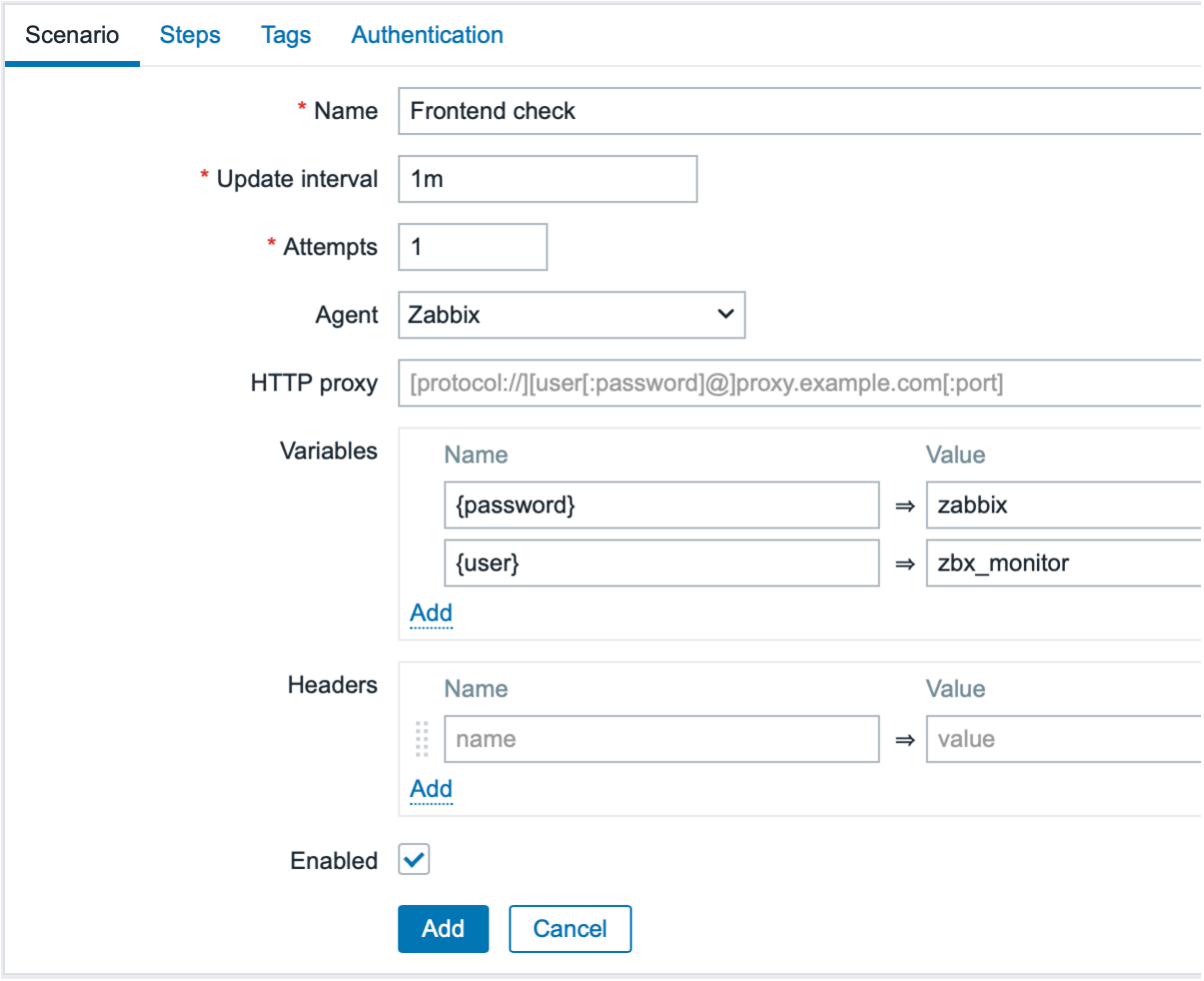
В этом разделе представлен пошаговый пример из реальной жизни, показывающий, как можно использовать веб-мониторинг.

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 он работает. Для этого требуется выполнить несколько шагов, включая проверку доступности первой страницы, вход с именем пользователя и паролем, проверку успешности входа, выход из системы и подтверждение выхода.

Сценарий

Добавление нового веб-сценария

Перейдите в *Сбор данных* → *Узлы сети*, выберите узел сети и нажмите *Веб* в строке этого узла сети.
Затем нажмите *Создать веб-сценарий*.



В форме нового сценария заполните следующие поля:

- **Имя** - Проверка веб-интерфейса
- **Интервал обновления** - 1m
- **Попытки** - 1
- **Агент** - Zabbix

В разделе *Переменные* добавьте две переменные: *{password}* и *{user}*.
Введите в качестве значений существующие учетные данные пользователя Zabbix.

В целях безопасности рекомендуется создать отдельного пользователя с минимальными правами доступа для использования в целях мониторинга.

При необходимости перейдите на вкладку *Теги* и добавьте теги веб-сценария.

Note:
После полной настройки этот веб-сценарий автоматически добавит на узел сети элемент данных Zabbix trapper.
Вы можете использовать теги веб-сценария, чтобы быстро находить связанные элементы данных и триггеры или выполнять поиск 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. Нажмите кнопку *Add*, чтобы добавить отдельный шаг.

Общие поля

Для каждого шага, описанного ниже, заполните следующие поля в дополнение к полям, специфичным для шага:

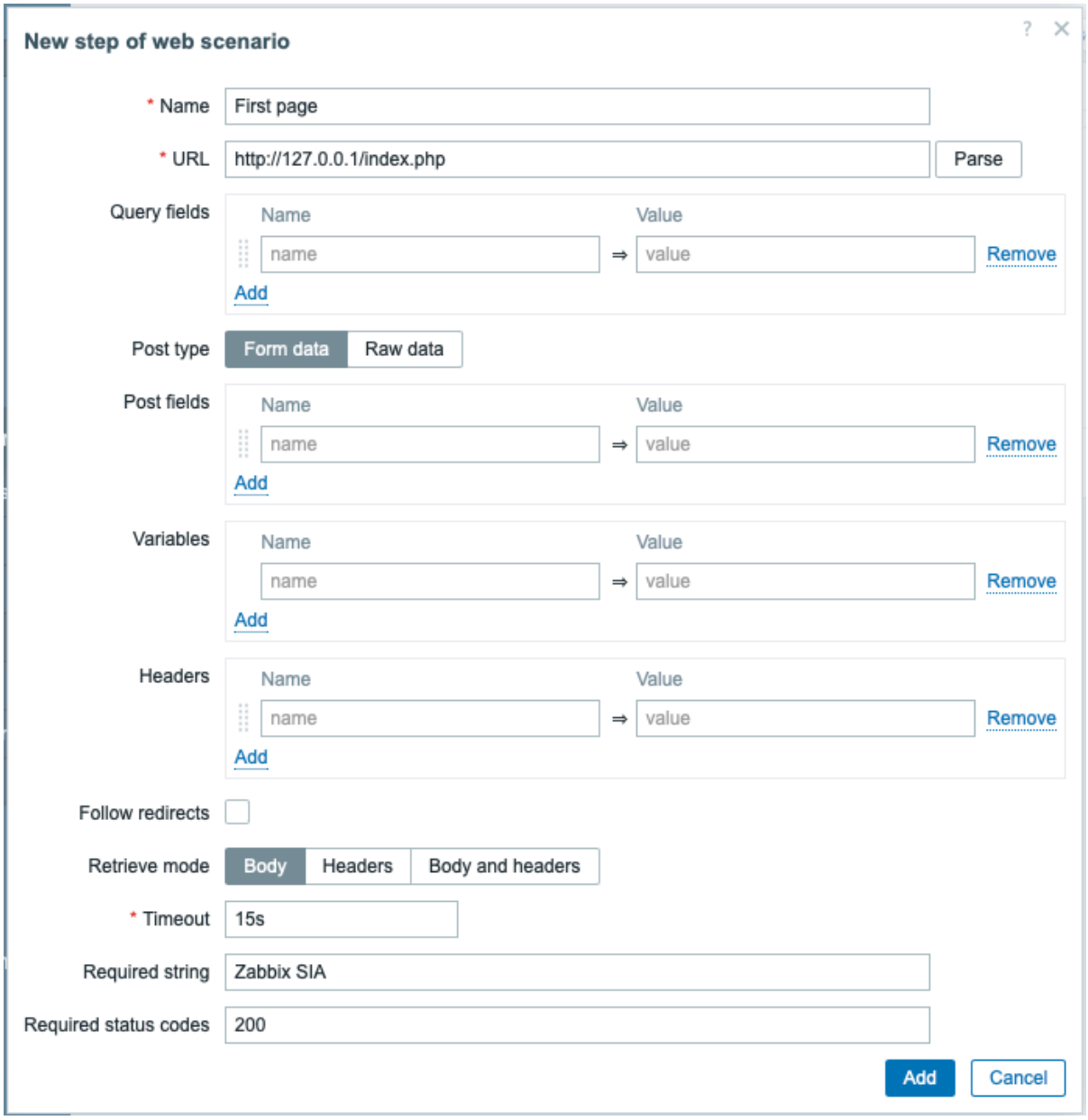
- **URL** — URL веб-интерфейса Zabbix
- **Timeout** — 15s
- **Required status codes** — 200

Шаг 1 веб-сценария

Проверьте, что первая страница отвечает корректно, возвращает HTTP-код ответа 200 и содержит текст "Zabbix SIA".

- В поле **Name** введите *First page*.
- В поле **Required string** введите *Zabbix SIA*.
- Заполните общие поля.

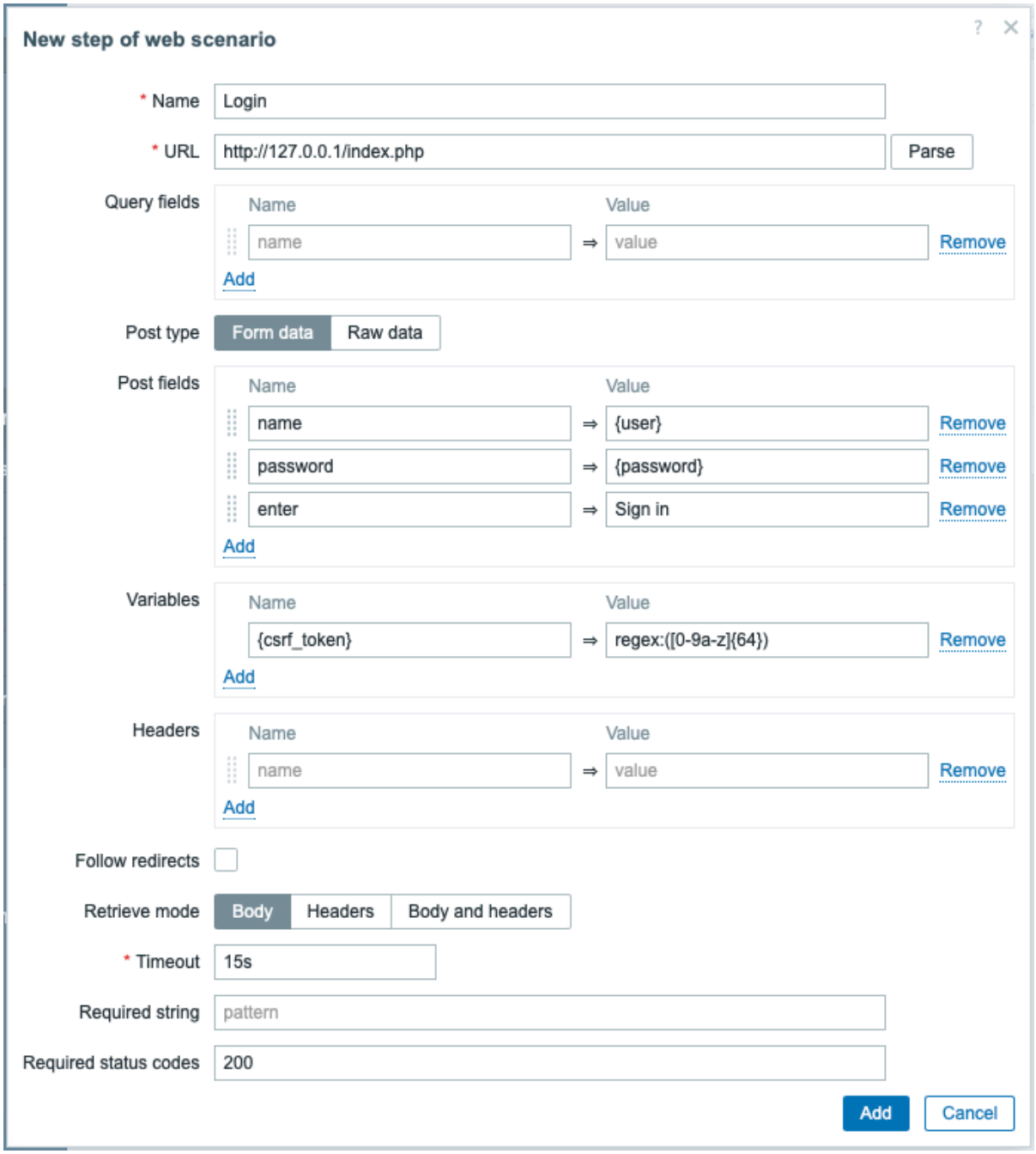
Когда настройка шага будет завершена, нажмите кнопку *Add*.



Шаг 2 веб-сценария

Войдите в веб-интерфейс Zabbix, используя макросы (переменные), определённые на уровне сценария — *{user}* и *{password}*.

- В поле **Имя** введите *Login*.
- В разделе **Поля POST** добавьте три POST-поля:
  - *name* со значением *{user}*
  - *password* со значением *{password}*
  - *enter* со значением *Sign in*
- В разделе **Переменные** добавьте новую переменную *{csrf_token}* со значением *regex:([0-9a-z]{64})*. Эта переменная перехватит значение назначенного CSRF-токена для повторного использования в шаге 4.
- Заполните общие поля.

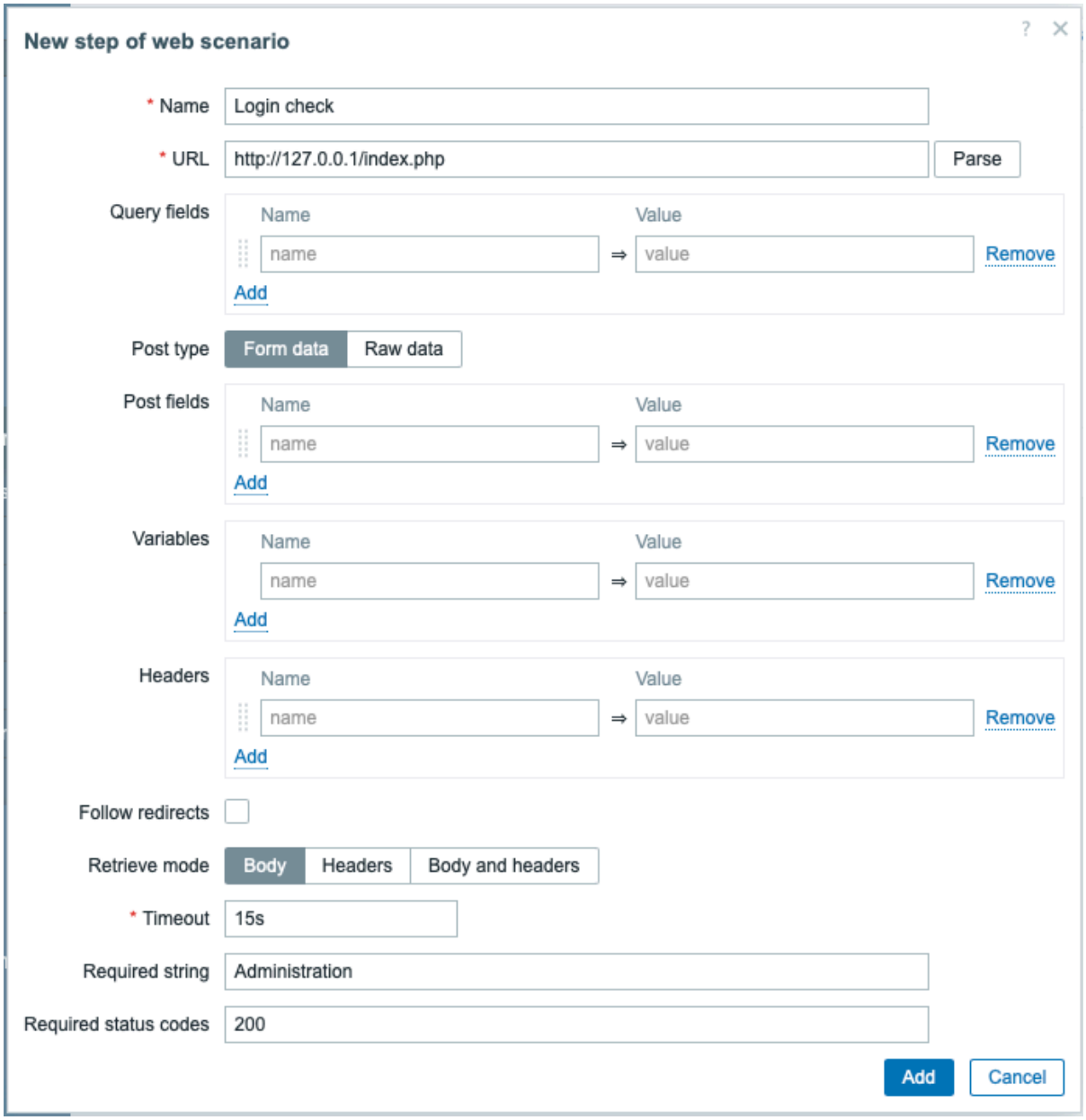


Attention:
Обратите внимание, что веб-интерфейс Zabbix при входе в систему использует перенаправление JavaScript, поэтому вход должен выполняться первым, а функции, доступные после входа, можно проверять только на следующих шагах. Кроме того, на шаге входа необходимо использовать полный URL к файлу **index.php**.

Шаг 3 веб-сценария

После входа в систему проверьте успешность входа, проверив наличие строки, видимой только после входа в систему, например *Administration*.

- В поле **Name** введите *Login check*.
- В поле **Required string** введите *Administration*.
- Заполните общие поля.

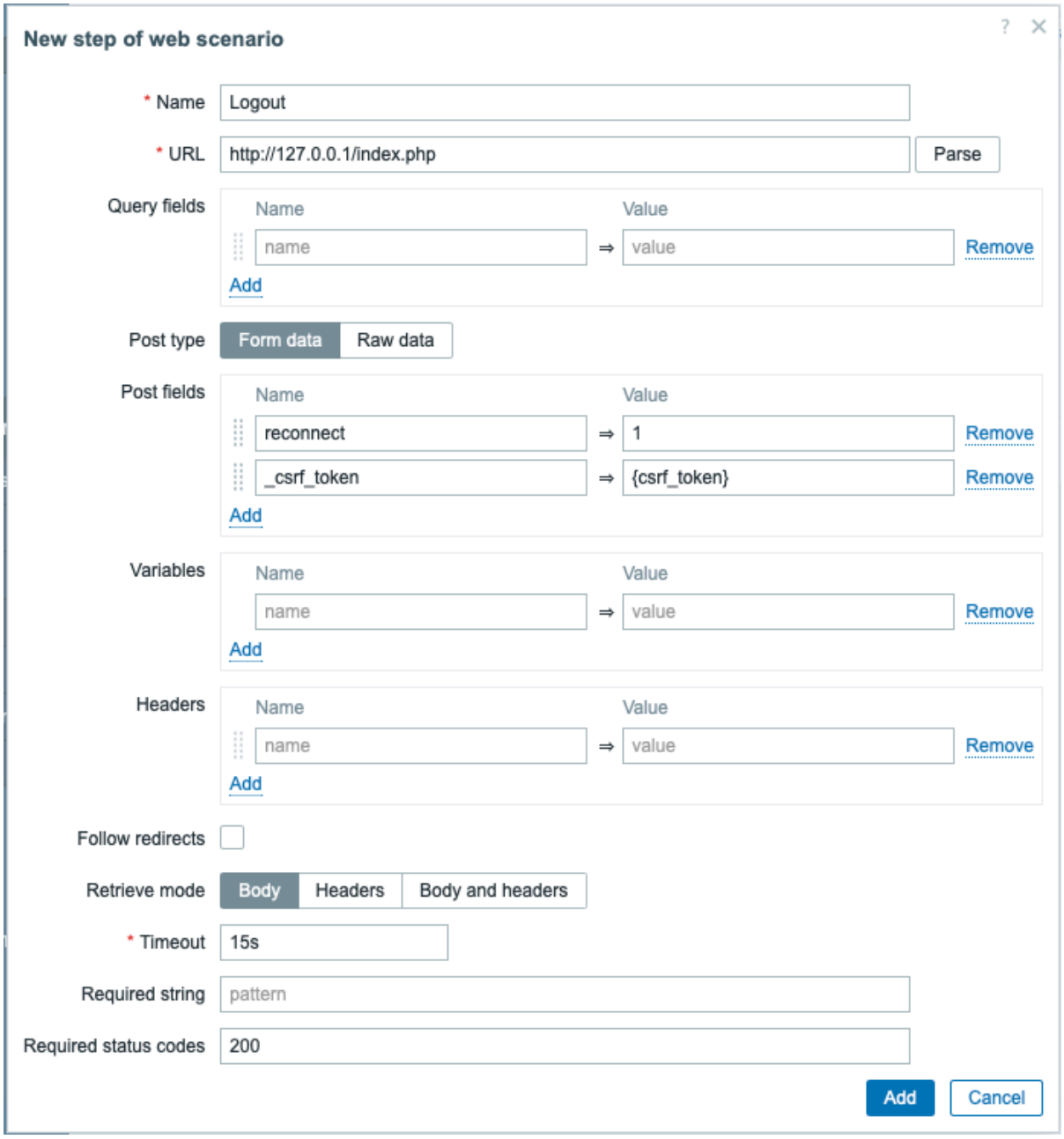


Шаг 4 веб-сценария

После того как доступность веб-интерфейса и вход в систему проверены, добавьте шаг выхода из системы — в противном случае база данных Zabbix будет засоряться множеством записей об открытых сессиях.

- В поле **Name** введите *Logout*.
- В разделе **Post fields** добавьте два post-поля:
  - *reconnect* со значением *1*
  - *_csrf_token* со значением *{csrf_token}*.
- Заполните общие поля.

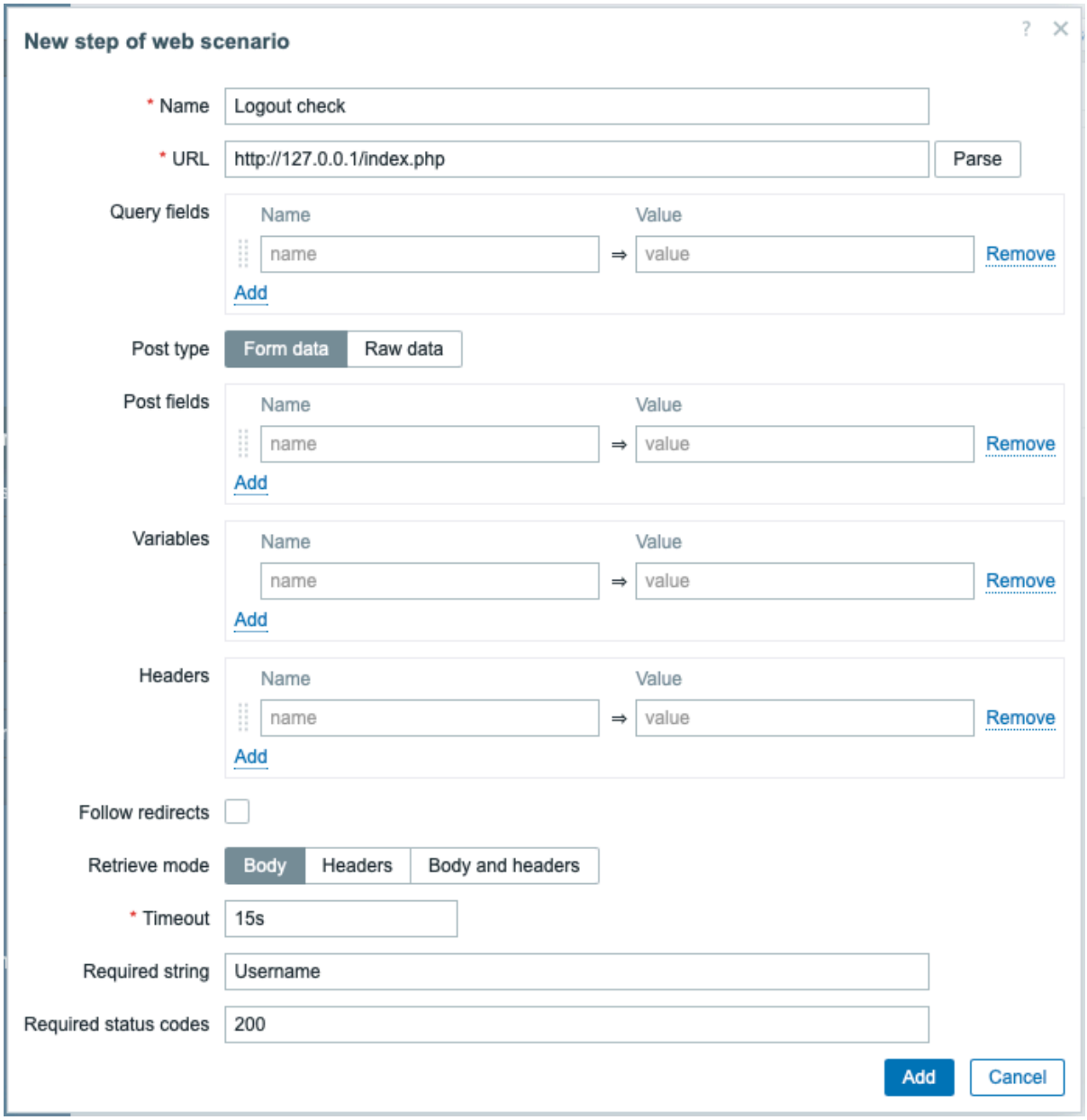
На этом шаге используется переменная {csrf_token}, полученная на шаге 2



Шаг 5 веб-сценария

Чтобы подтвердить выход из системы, проверьте наличие строки **Username**.

- В поле **Name** введите *Logout check*.
- В поле **Required string** введите *Username*.
- Заполните общие поля.



Полная настройка шагов

Полная настройка шагов веб-сценария должна выглядеть следующим образом:

	Name	Timeout	URL	Required	Status codes	Action
1:	First page	15s	http://127.0.0.1/index.php	Zabbix SIA	200	Remove
2:	Log in	15s	http://127.0.0.1/index.php		200	Remove
3:	Log in Check	15s	http://127.0.0.1/index.php	Administrati…	200	Remove
4:	Log out	15s	http://127.0.0.1/index.php		200	Remove
5:	Log out check	15s	http://127.0.0.1/index.php	Username	200	Remove

Add

Проверьте результаты

Сохраните готовый сценарий веб-мониторинга.

Сценарий будет добавлен к узлу сети. Чтобы просмотреть информацию о веб-сценарии, перейдите в *Мониторинг → Узлы сети*, найдите узел сети в списке и нажмите на гиперссылку Web в последнем столбце.



Нажмите на имя сценария, чтобы увидеть более подробную статистику:



8 Мониторинг виртуальных машин

Обзор Zabbix может использовать правила обнаружения низкого уровня для автоматического обнаружения гипервизоров VMware и виртуальных машин и создания узлов сети для их мониторинга на основе заранее определённых прототипов узлов сети.

Zabbix также включает готовые к использованию шаблоны для мониторинга VMware vCenter или гипервизоров ESXi.

Минимально требуемая версия VMware vCenter или vSphere — 5.1.

Сбор данных Мониторинг виртуальных машин состоит из двух этапов:

1. Процессы Zabbix *vmware collector* собирают данные виртуальных машин — эти процессы получают необходимую информацию из веб-сервисов VMware по протоколу SOAP, предварительно обрабатывают её и сохраняют в разделяемой памяти сервера Zabbix.
2. Процессы Zabbix *poller* получают данные с помощью простой проверки Zabbix ключи элементов данных мониторинга VMware.

Zabbix разделяет собираемые данные на данные конфигурации VMware и данные счётчиков производительности VMware. Оба типа данных собираются независимо процессами *vmware collector*.

На основе информации счётчиков производительности VMware доступна следующая статистика:

- Хранилище данных
- Дисковое устройство
- CPU
- Питание
- Сетевой интерфейс
- Пользовательские элементы данных счётчиков производительности

Полный список элементов данных, получающих данные из счётчиков производительности VMware, см. в разделе *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 Zabbix сервер скомпилирован из исходных кодов, он должен быть скомпилирован с параметрами конфигурации `--with-libcurl --with-libxml2`, чтобы включить мониторинг виртуальных машин. Пакеты Zabbix уже скомпилированы с включёнными этими параметрами.

Для мониторинга виртуальных машин можно изменить следующие параметры файла конфигурации Zabbix сервера:

- `StartVMwareCollectors`

Note:
Рекомендуется включить больше сборщиков, чем количество отслеживаемых сервисов VMware; в противном случае получение статистики счётчиков 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 сервисов VMware: количество сервисов < `StartVMwareCollectors` < (количество сервисов * 2). Например, при мониторинге одного сервиса VMware установите `StartVMwareCollectors` в 2; при мониторинге трёх сервисов установите `StartVMwareCollectors` в 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:
Для поддержки метрик ёмкости хранилища данных убедитесь, что значение ключа VMware `vpxd.stats.maxQueryMetrics` установлено как минимум в 64. Дополнительную информацию см. в статье *VMware Knowledge Base*.

Обнаружение

Zabbix может использовать правила низкоуровневого обнаружения (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 объектов. Для получения дополнительной информации см. *Прототипы узлов сети*.

Configuration examples

For a basic example of how to set up Zabbix for monitoring VMware using the *VMware FQDN* template, see *Monitor VMware with Zabbix*.

For a more detailed example of how to create a host, a low-level discovery rule, and a host prototype for monitoring VMware, see *Setup example*.

Расширенное журналирование Данные, собираемые процессами *vmware collector*, могут записываться в журнал для детальной отладки с использованием 5-го уровня отладки. 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 collector больше не требуется, рекомендуется снизить уровень отладки до значения по умолчанию (3), выполнив команду `-R log_level_decrease`.

Устранение неполадок

- Если метрики недоступны, убедитесь, что они не были сделаны недоступными или отключены по умолчанию в последних версиях VMware vSphere, а также что для запросов к базе данных метрик производительности не установлены какие-либо ограничения. Дополнительную информацию смотрите в ZBX-12094.
- Если возникает ошибка `config.vpxd.stats.maxQueryMetrics is invalid or exceeds the maximum number of characters permitted`, добавьте параметр `config.vpxd.stats.maxQueryMetrics` в настройки vCenter Server. Значение этого параметра должно совпадать со значением `maxQuerysize` в файле *web.xml* VMware. Дополнительную информацию смотрите в статье *VMware Knowledge Base*.
- Если вы подозреваете, что ваша установка Zabbix использует слишком много памяти, смотрите Профилирование чрезмерного использования памяти с tcmalloc.

1 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 по ключу элемента данных, чтобы увидеть полные сведения.

Item key	Description	Item group
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.	Кластер
vmware.cluster.alarms.get	Данные предупреждений 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 предупреждений 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, затрачиваемое на операцию чтения из хранилища данных.	
vmware.datastore.size	Объем пространства хранилища данных VMware в байтах или в процентах от общего объема.	
vmware.datastore.tags.get	Массив тегов хранилища данных VMware.	
vmware.datastore.write	Время, затрачиваемое на операцию записи в хранилище данных.	

Item key	Description	Item group
vmware.dc.alarms.get	Данные предупреждений 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.	Распределенный коммутатор vSphere
vmware.dvswitch.fetchports.get	Данные портов распределенного коммутатора VMware vSphere.	
vmware.hv.alarms.get	Данные предупреждений 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 (Гц).	
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, затрачиваемое на операцию чтения из хранилища данных.	
vmware.hv.datastore.size	Объем пространства хранилища данных VMware в байтах или в процентах от общего объема.	
vmware.hv.datastore.write	Среднее время, затрачиваемое на операцию 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 информацию 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.	
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 поставщика гипервизора VMware.	
vmware.hv.maintenance	Состояние обслуживания гипервизора VMware.	
vmware.hv.memory.size.ballooned	Размер ballooned-памяти гипервизора VMware.	
vmware.hv.memory.used	Объем используемой 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	Энергопотребление гипервизора VMware.	
vmware.hv.sensor.health.state	Сводный датчик состояния работоспособности гипервизора VMware.	
vmware.hv.sensors.get	Датчики состояния поставщика оборудования 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 пуле ресурсов VMware.	Пул ресурсов
vmware.rp.memory	Метрики памяти пула ресурсов VMware.	
vmware.alarms.get	Данные предупреждений виртуального центра VMware.	Виртуальный центр
vmware.vm.alarms.get	Данные предупреждений виртуальной машины VMware.	Виртуальная машина

Item key	Description	Item group
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 из-за конкуренции за доступ к физическому CPU(ам).	
vmware.vm.cpu.num	Количество процессоров виртуальной машины VMware.	
vmware.vm.cpu.readiness	Процент времени, в течение которого виртуальная машина была готова к выполнению, но не могла быть запланирована на выполнение на физическом CPU.	
vmware.vm.cpu.ready	Время, в течение которого виртуальная машина была готова к выполнению, но не могла быть запланирована на выполнение на физическом CPU.	
vmware.vm.cpu.swapwait	Процент процессорного времени, затраченного на ожидание подкачки из swap.	
vmware.vm.cpu.usage	Использование процессора виртуальной машины VMware (Гц).	
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 системы, 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	Размер ballooned-памяти виртуальной машины VMware.	
vmware.vm.memory.size.compressed	Размер сжатой памяти виртуальной машины VMware.	
vmware.vm.memory.size.consumed	Объем физической памяти узла сети, используемой для хранения страниц физической памяти гостевой системы.	
vmware.vm.memory.size.private	Размер частной памяти виртуальной машины VMware.	
vmware.vm.memory.size.shared	Размер общей памяти виртуальной машины VMware.	
vmware.vm.memory.size.swapped	Размер памяти виртуальной машины VMware, выгруженной в swap.	
vmware.vm.memory.size.usage.guest	Использование памяти гостевой системы виртуальной машины VMware.	
vmware.vm.memory.size.usage.host	Использование памяти узла сети виртуальной машины VMware.	
vmware.vm.memory.usage	Процент физической памяти узла сети, который был использован.	
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	Выделенное пространство хранения виртуальной машины VMware.	
vmware.vm.storage.readoio	Среднее количество незавершенных запросов чтения к виртуальному диску за интервал сбора.	
vmware.vm.storage.totalreadlatency	Среднее время, которое занимает чтение с виртуального диска.	
vmware.vm.storage.totalwritelatency	Среднее время, которое занимает запись на виртуальный диск.	
vmware.vm.storage.uncommitted	Невыделенное пространство хранения виртуальной машины VMware.	
vmware.vm.storage.unshared	Необщее пространство хранения виртуальной машины VMware.	
vmware.vm.storage.writeoio	Среднее количество незавершенных запросов записи к виртуальному диску за интервал сбора.	
vmware.vm.tags.get	Массив тегов виртуальной машины VMware.	
vmware.vm.tools	Состояние или версия гостевых инструментов 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.	

Item key	Description	Item group
vmware.vm.vfs.fs.discovery	Обнаружение файловых систем виртуальной машины VMware.	
vmware.vm.vfs.fs.size	Статистика файловой системы виртуальной машины VMware.	

Подробности ключа элемента данных Параметры без угловых скобок являются обязательными. Параметры, отмеченные угловыми скобками **< >**, являются необязательными.

vmware.eventlog[url,<mode>,<severity>]

 Журнал событий VMware.
 Возвращаемое значение: *Log*.

Параметры:

- **url** — URL сервиса VMware;
- **mode** — *all* (по умолчанию) или *skip* — пропустить обработку старых данных;
- **severity** — фильтр по уровню важности: *error*, *warning*, *info* или *user*. Этот параметр должен быть заключён в кавычки, если указано более одного уровня важности в списке, разделённом запятыми (например, `"error,warning,info,use` По умолчанию отключён.

Комментарии:

- Для каждого URL должен быть только один ключ элемента данных `vmware.eventlog`;
- См. также пример фильтрации записей журнала событий VMware.

vmware.fullnameurl

 Полное имя сервиса VMware.
 Возвращаемое значение: *String*.

Параметры:

- **url** — URL сервиса VMware.

vmware.versionurl

 Версия сервиса VMware.
 Возвращаемое значение: *String*.

Параметры:

- **url** — URL сервиса VMware.

vmware.cl.perfcounter[url,id,path,<instance>]

 Метрики счётчика производительности кластера VMware.
 Возвращаемое значение: *Integer*.

Параметры:

- **url** — URL сервиса VMware;
- **id** — ID кластера VMware. `id` можно получить из `vmware.cluster.discovery[]` как {#CLUSTER.ID}.
- **path** — путь счётчика производительности[1];
- **instance** — экземпляр счётчика производительности.

vmware.cluster.alarms.get[url,id]

 Данные о сигналах тревоги кластера VMware.
 Возвращаемое значение: объект JSON.

Параметры:

- **url** — URL сервиса VMware;
- **id** — ID кластера VMware.

vmware.cluster.discoveryurl

 Обнаружение кластеров VMware.
 Возвращаемое значение: объект JSON.

Параметры:

- **url** — URL сервиса VMware.

vmware.cluster.property[url,id,prop]

 Свойство кластера VMware.
 Возвращаемое значение: *String*.

Параметры:

- **url** — URL сервиса VMware;
- **id** — ID кластера VMware;

- **prop** — путь к свойству, то есть имя 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,name]

 Статус кластера VMware.
 Возвращаемое значение: *0* — серый; *1* — зеленый; *2* — желтый; *3* — красный.

Параметры:

- **url** — URL сервиса VMware;
- **name** — имя кластера VMware.

vmware.cluster.tags.get[url,id]

 Массив тегов кластера VMware.
 Возвращаемое значение: объект JSON.

Параметры:

- **url** — URL сервиса VMware;
- **id** — ID кластера VMware.

Этот элемент данных работает с vSphere 6.5 и новее.

vmware.datastore.alarms.get[url,uuid]

 Данные о предупреждениях хранилища данных VMware.
 Возвращаемое значение: объект JSON.

Параметры:

- **url** — URL сервиса 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 сервиса VMware;
- **filter_uuid** — один уникальный идентификатор HV или VM (по умолчанию: пусто). Если указан, будут обнаружены только связанные хранилища данных.

vmware.datastore.hv.list[url,datastore]

 Список гипервизоров хранилища данных.
 Возвращаемое значение: *String*.

Параметры:

- **url** — URL сервиса VMware;
- **datastore** — UUID или имя хранилища данных.

Пример вывода:

```
esx7-01-host.zabbix.sandbox
esx7-02-host.zabbix.sandbox
```

vmware.datastore.perfcounter[url,uuid,path,<instance>]

 Значение счётчика производительности хранилища данных VMware.
 Возвращаемое значение: *Целое число* [2].

Параметры:

- **url** — URL сервиса VMware;
- **uuid** — глобальный уникальный идентификатор хранилища данных VMware;
- **path** — путь счётчика производительности[1];
- **instance** — экземпляр счётчика производительности. Используйте пустой экземпляр для агрегированных значений (по умолчанию). `instance` можно получить из `vmware.datastore.discovery[]` как часть массива {#DATASTORE.EXTENT}.

vmware.datastore.property[url,uuid,prop]

 Свойство хранилища данных VMware.
 Возвращаемое значение: *String*.

Параметры:

- **url** — URL сервиса VMware;
- **uuid** — глобальный уникальный идентификатор хранилища данных VMware;
- **prop** — путь к свойству.

vmware.datastore.read[url,datastore,<mode>]

 Время, затрачиваемое на операцию чтения из хранилища данных (в миллисекундах).
 Возвращаемое значение: *Integer* [2].

Параметры:

- **url** — URL сервиса VMware;
- **datastore** — UUID или имя хранилища данных;
- **mode** — *latency* (среднее значение, по умолчанию) или *maxlatency* (максимальное значение).

vmware.datastore.size[url,datastore,<mode>]

 Пространство VMware datastore в байтах или в процентах от общего объема.
 Возвращаемое значение: *Integer* — для байтов; *Float* — для процентов.

Параметры:

- **url** — URL сервиса VMware;
- **datastore** — UUID или имя datastore;
- **mode** — возможные значения: *total* (по умолчанию), *free*, *pfree* (процент свободного пространства), *uncommitted*.

vmware.datastore.tags.get[url,uuid]

 Массив тегов хранилища данных VMware.
 Возвращаемое значение: объект JSON.

Параметры:

- **url** — URL сервиса VMware;
- **uuid** — глобальный уникальный идентификатор хранилища данных VMware.

Этот элемент данных работает с vSphere 6.5 и новее.

vmware.datastore.write[url,datastore,<mode>]

 Время, затрачиваемое на операцию записи в хранилище данных (в миллисекундах).
 Возвращаемое значение: *Integer* [2].

Параметры:

- **url** — URL сервиса VMware;
- **datastore** — UUID или имя хранилища данных;
- **mode** — *latency* (среднее значение, по умолчанию) или *maxlatency* (максимальное значение).

vmware.dc.alarms.get[url,id]

 Данные о предупреждениях центра обработки данных VMware.
 Возвращаемое значение: объект JSON.

Параметры:

- **url** — URL сервиса VMware;
- **id** — ID центра обработки данных VMware.

vmware.dc.discoveryurl

 Обнаружение центров обработки данных VMware.
 Возвращаемое значение: объект JSON.

Параметры:

- **url** — URL сервиса VMware.

vmware.dc.tags.get[url,id]

 Массив тегов центра обработки данных VMware.
 Возвращаемое значение: объект JSON.

Параметры:

- **url** — URL сервиса VMware;
- **id** — идентификатор центра обработки данных VMware.

Этот элемент данных работает с vSphere 6.5 и новее.

vmware.dvswitch.discoveryurl

 Обнаружение распределённых коммутаторов VMware vSphere.
 Возвращаемое значение: объект JSON.

Параметры:

- **url** — URL сервиса VMware.

vmware.dvswitch.fetchports.get[url,uuid,<filter>,<mode>]

 Данные о портах VMware vSphere Distributed Switch.
 Возвращаемое значение: объект JSON.

Параметры:

- **url** — URL сервиса VMware;
- **uuid** — глобальный уникальный идентификатор VMware DVSwitch.
- **filter** — одна строка с критериями выбора портов, разделёнными запятыми;
- **mode** — *state* (весь XML без XML-узлов "config", по умолчанию) или *full*.

Параметр **filter** поддерживает критерии, 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 о предупреждениях гипервизора VMware.
 Возвращаемое значение: объект JSON.

Параметры:

- **url** — URL сервиса VMware;
- **uuid** — глобальный уникальный идентификатор гипервизора VMware.

vmware.hv.cluster.name[url,uuid]

 Имя кластера гипервизора VMware.
 Возвращаемое значение: *String*.

Параметры:

- **url** — URL сервиса VMware;
- **uuid** — глобальный уникальный идентификатор гипервизора VMware.

vmware.hv.connectionstate[url,uuid]

 Состояние подключения гипервизора VMware.
 Возвращаемое значение: *String*: *connected*, *disconnected* или *notResponding*.

Параметры:

- **url** — URL сервиса VMware;
- **uuid** — глобальный уникальный идентификатор гипервизора VMware.

vmware.hv.cpu.usage[url,uuid]

 Использование процессора гипервизора VMware (Гц).
 Возвращаемое значение: *Целое число*.

Параметры:

- **url** — URL сервиса VMware;
- **uuid** — глобальный уникальный идентификатор гипервизора VMware.

vmware.hv.cpu.usage.perf[url,uuid]

 Использование процессора гипервизора VMware в процентах за интервал.
 Возвращаемое значение: *Float*.

Параметры:

- **url** — URL сервиса VMware;
- **uuid** — глобальный уникальный идентификатор гипервизора VMware.

vmware.hv.cpu.utilization[url,uuid]

 Использование процессора гипервизора VMware в процентах за интервал; зависит от управления питанием или HT.
 Возвращаемое значение: *Float*.

Параметры:

- **url** — URL сервиса VMware;
- **uuid** — глобальный уникальный идентификатор гипервизора VMware.

vmware.hv.datacenter.name[url,uuid]

 Имя центра обработки данных гипервизора VMware.
 Возвращаемое значение: *String*.

Параметры:

- **url** — URL сервиса VMware;
- **uuid** — глобальный уникальный идентификатор гипервизора VMware.

vmware.hv.datastore.discovery[url,uuid]

 Обнаружение хранилищ данных гипервизора VMware.
 Возвращаемое значение: объект JSON.

Параметры:

- **url** — URL сервиса VMware;
- **uuid** — глобальный уникальный идентификатор гипервизора VMware.

vmware.hv.datastore.list[url,uuid]

 Список хранилищ данных гипервизора VMware.
 Возвращаемое значение: *String*.

Параметры:

- **url** — URL сервиса VMware;
- **uuid** — глобальный уникальный идентификатор гипервизора VMware.

Пример вывода:

```
SSD-RAID1-VAULT1
SSD-RAID1-VAULT2
SSD-RAID10
```

vmware.hv.datastore.multipath[url,uuid,<datastore>,<partitionid>]

 Количество доступных путей к datastore.
 Возвращаемое значение: *Целое число*.

Параметры:

- **url** — URL сервиса VMware;
- **uuid** — глобальный уникальный идентификатор гипервизора VMware;
- **datastore** — UUID или имя datastore;
- **partitionid** — внутренний ID физического устройства из `vmware.hv.datastore.discovery`.

vmware.hv.datastore.read[url,uuid,datastore,<mode>]

 Среднее время выполнения операции чтения из хранилища данных (в миллисекундах).
 Возвращаемое значение: *Integer* [2].

Параметры:

- **url** — URL сервиса VMware;
- **uuid** — глобальный уникальный идентификатор гипервизора VMware;
- **datastore** — UUID или имя хранилища данных;
- **mode** — *latency* (по умолчанию).

vmware.hv.datastore.size[url,uuid,datastore,<mode>]

 Пространство хранилища данных VMware в байтах или в процентах от общего объема.
 Возвращаемое значение: *Integer* — для байтов; *Float* — для процентов.

Параметры:

- **url** — URL сервиса VMware;
- **uuid** — глобальный уникальный идентификатор гипервизора VMware;
- **datastore** — UUID или имя хранилища данных;
- **mode** — возможные значения: *total* (по умолчанию), *free*, *pfree* (процент свободного пространства), *uncommitted*.

vmware.hv.datastore.write[url,uuid,datastore,<mode>]

 Среднее время выполнения операции записи в хранилище данных (в миллисекундах).
 Возвращаемое значение: *Целое число* [2].

Параметры:

- **url** — URL сервиса VMware;
- **uuid** — глобальный уникальный идентификатор гипервизора VMware;

- **datastore** — UUID или имя хранилища данных;
- **mode** — *latency* (по умолчанию).

vmware.hv.discoveryurl

 Обнаружение гипервизоров VMware.
 Возвращаемое значение: объект JSON.

Параметры:

- **url** — URL сервиса VMware.

vmware.hv.diskinfo.get[url,uuid]

 Данные о дисках гипервизора VMware.
 Возвращаемое значение: объект JSON.

Параметры:

- **url** — URL сервиса VMware;
- **uuid** — глобальный уникальный идентификатор гипервизора VMware.

vmware.hv.fullname[url,uuid]

 Полное название продукта, включая информацию о версии.
 Возвращаемое значение: *String*.

Параметры:

- **url** — URL сервиса 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 сервиса VMware;
- **uuid** — глобальный уникальный идентификатор гипервизора VMware.

vmware.hv.hw.cpu.model[url,uuid]

 Модель процессора гипервизора VMware.
 Возвращаемое значение: *String*.

Параметры:

- **url** — URL сервиса VMware;
- **uuid** — глобальный уникальный идентификатор гипервизора VMware.

vmware.hv.hw.cpu.num[url,uuid]

 Количество ядер процессора на гипервизоре VMware.
 Возвращаемое значение: *Целое число*.

Параметры:

- **url** — URL сервиса VMware;
- **uuid** — глобальный уникальный идентификатор гипервизора VMware.

vmware.hv.hw.cpu.threads[url,uuid]

 Количество потоков процессора на гипервизоре VMware.
 Возвращаемое значение: *Целое число*.

Параметры:

- **url** — URL сервиса VMware;
- **uuid** — глобальный уникальный идентификатор гипервизора VMware.

vmware.hv.hw.memory[url,uuid]

 Общий размер памяти гипервизора VMware (в байтах).
 Возвращаемое значение: *Целое число*.

Параметры:

- **url** — URL сервиса VMware;
- **uuid** — глобальный уникальный идентификатор гипервизора VMware.

vmware.hv.hw.model[url,uuid]

 Модель гипервизора VMware.
 Возвращаемое значение: *String*.

Параметры:

- **url** — URL сервиса VMware;

- **uuid** — глобальный уникальный идентификатор гипервизора VMware.

vmware.hv.hw.sensors.get[url,uuid]

 Значение аппаратных датчиков гипервизора VMware.
 Возвращаемое значение: объект JSON.

Параметры:

- **url** — URL сервиса VMware;
- **uuid** — глобальный уникальный идентификатор гипервизора VMware.

vmware.hv.hw.serialnumber[url,uuid]

 Серийный номер гипервизора VMware.
 Возвращаемое значение: *String*.

Параметры:

- **url** — URL сервиса 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: *String*.

Параметры:

- **url** — URL сервиса 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: *String*.

Параметры:

- **url** — URL сервиса VMware;
- **uuid** — глобальный уникальный идентификатор гипервизора VMware.

Этот элемент данных работает с vSphere API 6.7 и новее.

vmware.hv.maintenance[url,uuid]

 Статус обслуживания гипервизора VMware.
 Возвращаемое значение: *0* — не в режиме обслуживания; *1* — в режиме обслуживания.

Параметры:

- **url** — URL сервиса VMware;
- **uuid** — глобальный уникальный идентификатор гипервизора VMware.

vmware.hv.memory.size.ballooned[url,uuid]

 Размер памяти VMware hypervisor, выделенной с помощью ballooning (в байтах).
 Возвращаемое значение: *Integer*.

Параметры:

- **url** — URL сервиса VMware;
- **uuid** — глобальный уникальный идентификатор VMware hypervisor.

vmware.hv.memory.used[url,uuid]

 Объём используемой памяти гипервизора VMware (в байтах).
 Возвращаемое значение: *Целое число*.

Параметры:

- **url** — URL сервиса 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 сервиса VMware;

- **uuid** — глобальный уникальный идентификатор гипервизора VMware.

vmware.hv.network.in[url,uuid,<mode>]

 Статистика входящего сетевого трафика гипервизора VMware (байт в секунду).
 Возвращаемое значение: *Integer* [2].

Параметры:

- **url** — URL сервиса VMware;
- **uuid** — глобальный уникальный идентификатор гипервизора VMware;
- **mode** — *bps* (по умолчанию), *packets*, *dropped*, *errors*, *broadcast*.

vmware.hv.network.linkspeed[url,uuid,ifname]

 Скорость сетевого интерфейса гипервизора VMware.
 Возвращаемое значение: *Целое число*. Возвращает *0*, если сетевой интерфейс не работает; в противном случае — значение скорости интерфейса.

Параметры:

- **url** — URL сервиса VMware;
- **uuid** — глобальный уникальный идентификатор гипервизора VMware;
- **ifname** — имя интерфейса.

vmware.hv.network.out[url,uuid,<mode>]

 Статистика исходящего сетевого трафика гипервизора VMware (байт в секунду).
 Возвращаемое значение: *Integer* [2].

Параметры:

- **url** — URL сервиса VMware;
- **uuid** — глобальный уникальный идентификатор гипервизора VMware;
- **mode** — *bps* (по умолчанию), *packets*, *dropped*, *errors*, *broadcast*.

vmware.hv.perfcounter[url,uuid,path,<instance>]

 Значение счётчика производительности гипервизора VMware.
 Возвращаемое значение: *Целое число* [2].

Параметры:

- **url** — URL сервиса VMware;
- **uuid** — глобальный уникальный идентификатор гипервизора VMware;
- **path** — путь счётчика производительности [1];
- **instance** — экземпляр счётчика производительности. Используйте пустой экземпляр для агрегированных значений (по умолчанию).

vmware.hv.property[url,uuid,prop]

 Свойство гипервизора VMware.
 Возвращаемое значение: *String*.

Параметры:

- **url** — URL сервиса VMware;
- **uuid** — глобальный уникальный идентификатор гипервизора VMware;
- **prop** — путь к свойству.

vmware.hv.power[url,uuid,<max>]

 Потребляемая мощность гипервизора VMware (Вт).
 Возвращаемое значение: *Целое число*.

Параметры:

- **url** — URL сервиса VMware;
- **uuid** — глобальный уникальный идентификатор гипервизора VMware;
- **max** — максимально допустимая потребляемая мощность.

vmware.hv.sensor.health.state[url,uuid]

 Сенсор сводного состояния работоспособности гипервизора VMware.
 Возвращаемое значение: *Integer*: *0* — серый; *1* — зелёный; *2* — жёлтый; *3* — красный.

Параметры:

- **url** — URL сервиса VMware;
- **uuid** — глобальный уникальный идентификатор гипервизора VMware.

Обратите внимание, что элемент данных может не работать в VMware vSphere 6.5 и более новых версиях, поскольку VMware объявила сенсор *VMware Rollup Health State* устаревшим.

vmware.hv.sensors.get[url,uuid]

 Датчики состояния производителя оборудования гипервизора VMware.
 Возвращаемое значение: объект JSON.

Параметры:

- **url** — URL сервиса VMware;
- **uuid** — глобальный уникальный идентификатор гипервизора VMware.

vmware.hv.status[url,uuid]

 Статус гипервизора VMware.
 Возвращаемое значение: *Integer*: *0* — серый; *1* — зеленый; *2* — желтый; *3* — красный.

Параметры:

- **url** — URL сервиса VMware;
- **uuid** — глобальный уникальный идентификатор гипервизора VMware.

Этот элемент данных использует свойство общего состояния хост-системы.

vmware.hv.tags.get[url,uuid]

 Массив тегов гипервизора VMware.
 Возвращаемое значение: объект JSON.

Параметры:

- **url** — URL сервиса VMware;
- **uuid** — глобальный уникальный идентификатор гипервизора VMware.

Этот элемент данных работает с vSphere 6.5 и новее.

vmware.hv.uptime[url,uuid]

 Время непрерывной работы гипервизора VMware (в секундах).
 Возвращаемое значение: *Целое число*.

Параметры:

- **url** — URL сервиса VMware;
- **uuid** — глобальный уникальный идентификатор гипервизора VMware.

Этот элемент данных использует свойство общего состояния хостовой системы.

vmware.hv.version[url,uuid]

 Версия гипервизора VMware.
 Возвращаемое значение: *String*.

Параметры:

- **url** — URL сервиса 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 сервиса 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: *Integer*.

Параметры:

- **url** — URL сервиса VMware;
- **rpid** — ID пула ресурсов VMware.

vmware.rp.memory[url,rpid,<mode>]

 Метрики памяти пула ресурсов VMware.
 Возвращаемое значение: *Integer*.

Параметры:

- **url** — URL сервиса VMware;

- **rpid** — ID пула ресурсов VMware;
- **mode** — возможные значения:
consumed (по умолчанию) — объём физической памяти узла сети, используемой для резервирования страниц физической памяти гостевой системы
 ballooned — объём физической памяти гостевой системы, возвращённой из виртуальной машины драйвером balloon в гостевой системе
overhead — объём физической памяти узла сети, потребляемой структурами данных ESXi для работы виртуальных машин

vmware.alarms.geturl

 Данные о сигналах тревоги VMware VirtualCenter.
 Возвращаемое значение: объект JSON.

Параметры:

- **url** — URL сервиса VMware.

vmware.vm.alarms.get[url,uuid]

 Данные о предупреждениях виртуальной машины VMware.
 Возвращаемое значение: объект JSON.

Параметры:

- **url** — URL сервиса VMware;
- **uuid** — глобальный уникальный идентификатор виртуальной машины VMware.

vmware.vm.attribute[url,uuid,name]

 Значение пользовательского атрибута виртуальной машины VMware.
 Возвращаемое значение: *String*.

Параметры:

- **url** — URL сервиса VMware;
- **uuid** — глобальный уникальный идентификатор виртуальной машины VMware;
- **name** — имя пользовательского атрибута.

vmware.vm.cluster.name[url,uuid]

 Имя виртуальной машины VMware.
 Возвращаемое значение: *String*.

Параметры:

- **url** — URL сервиса VMware;
- **uuid** — глобальный уникальный идентификатор виртуальной машины VMware.

vmware.vm.consolidationneeded[url,uuid]

 Диску виртуальной машины VMware требуется консолидация.
 Возвращаемое значение: *String*: *true* — консолидация требуется; *false* — консолидация не требуется.

Параметры:

- **url** — URL сервиса VMware;
- **uuid** — глобальный уникальный идентификатор виртуальной машины VMware.

vmware.vm.cpu.latency[url,uuid]

 Процент времени, в течение которого виртуальная машина не может выполняться из-за конкуренции за доступ к физическим CPU.
 Возвращаемое значение: *Float*.

Параметры:

- **url** — URL сервиса 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 сервиса VMware;
- **uuid** — глобальный уникальный идентификатор виртуальной машины VMware.

vmware.vm.cpu.readiness[url,uuid,<instance>]

 Процент времени, в течение которого виртуальная машина была готова, но не могла быть запланирована для выполнения на физическом CPU.
 Возвращаемое значение: *Float*.

Параметры:

- **url** — URL сервиса VMware;

- **uuid** — глобальный уникальный идентификатор виртуальной машины VMware;
- **instance** — экземпляр CPU.

vmware.vm.cpu.ready[url,uuid]

 Время (в миллисекундах), в течение которого виртуальная машина была готова, но не могла быть запланирована для выполнения на физическом CPU. Время готовности CPU зависит от количества виртуальных машин на узле сети и их загрузки CPU (%).
 Возвращаемое значение: *Целое число* [2].

Параметры:

- **url** — URL сервиса VMware;
- **uuid** — глобальный уникальный идентификатор виртуальной машины VMware.

vmware.vm.cpu.swapwait[url,uuid,<instance>]

 Процент времени CPU, затраченного на ожидание подкачки в память.
 Возвращаемое значение: *Float*.

Параметры:

- **url** — URL сервиса VMware;
- **uuid** — глобальный уникальный идентификатор виртуальной машины VMware;
- **instance** — экземпляр CPU.

vmware.vm.cpu.usage[url,uuid]

 Использование процессора виртуальной машины VMware (Гц).
 Возвращаемое значение: *Целое число*.

Параметры:

- **url** — URL сервиса VMware;
- **uuid** — глобальный уникальный идентификатор виртуальной машины VMware.

vmware.vm.cpu.usage.perf[url,uuid]

 Использование процессора виртуальной машины VMware в процентах за интервал.
 Возвращаемое значение: *Float*.

Параметры:

- **url** — URL сервиса VMware;
- **uuid** — глобальный уникальный идентификатор виртуальной машины VMware.

vmware.vm.datacenter.name[url,uuid]

 Имя центра обработки данных виртуальной машины VMware.
 Возвращаемое значение: *String*.

Параметры:

- **url** — URL сервиса VMware;
- **uuid** — глобальный уникальный идентификатор виртуальной машины VMware.

vmware.vm.discoveryurl

 Обнаружение виртуальных машин VMware.
 Возвращаемое значение: объект JSON.

Параметры:

- **url** — URL сервиса VMware.

vmware.vm.guest.memory.size.swapped[url,uuid]

 Объём физической памяти гостевой системы, выгруженной в пространство подкачки (КБ).
 Возвращаемое значение: *Целое число*.

Параметры:

- **url** — URL сервиса VMware;
- **uuid** — глобальный уникальный идентификатор виртуальной машины VMware.

vmware.vm.guest.osuptime[url,uuid]

 Общее время, прошедшее с момента последней загрузки операционной системы (в секундах).
 Возвращаемое значение: *Целое число*.

Параметры:

- **url** — URL сервиса 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: *String*.

Параметры:

- **url** — URL сервиса 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 сервиса VMware;
- **uuid** — глобальный уникальный идентификатор виртуальной машины VMware.

vmware.vm.memory.size.ballooned[url,uuid]

 Размер памяти (в байтах), выделенной с помощью ballooning, виртуальной машины VMware.
 Возвращаемое значение: *Целое число*.

Параметры:

- **url** — URL сервиса 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 сервиса VMware;
- **uuid** — глобальный уникальный идентификатор виртуальной машины VMware.

vmware.vm.memory.size.consumed[url,uuid]

 Объём физической памяти узла сети, используемой для резервирования страниц физической памяти гостевой системы (КБ).
 Возвращаемое значение: *Целое число*.

Параметры:

- **url** — URL сервиса VMware;
- **uuid** — глобальный уникальный идентификатор виртуальной машины VMware.

vmware.vm.memory.size.private[url,uuid]

 Размер частной памяти виртуальной машины VMware (в байтах).
 Возвращаемое значение: *Целое число*.

Параметры:

- **url** — URL сервиса VMware;
- **uuid** — глобальный уникальный идентификатор виртуальной машины VMware.

vmware.vm.memory.size.shared[url,uuid]

 Размер общей памяти виртуальной машины VMware (в байтах).
 Возвращаемое значение: *Целое число*.

Параметры:

- **url** — URL сервиса VMware;
- **uuid** — глобальный уникальный идентификатор виртуальной машины VMware.

vmware.vm.memory.size.swapped[url,uuid]

 Размер памяти подкачки виртуальной машины VMware (в байтах).
 Возвращаемое значение: *Целое число*.

Параметры:

- **url** — URL сервиса VMware;
- **uuid** — глобальный уникальный идентификатор виртуальной машины VMware.

vmware.vm.memory.size.usage.guest[url,uuid]

 Использование памяти гостевой системы виртуальной машины VMware (в байтах).
 Возвращаемое значение: *Целое число*.

Параметры:

- **url** — URL сервиса VMware;
- **uuid** — глобальный уникальный идентификатор виртуальной машины VMware.

vmware.vm.memory.size.usage.host[url,uuid]

 Использование памяти узла сети виртуальной машины VMware (в байтах).
 Возвращаемое значение: *Целое число*.

Параметры:

- **url** — URL сервиса VMware;
- **uuid** — глобальный уникальный идентификатор виртуальной машины VMware.

vmware.vm.memory.usage[url,uuid]

 Процент физической памяти узла сети, который был использован.
 Возвращаемое значение: *Float*.

Параметры:

- **url** — URL сервиса 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 сервиса VMware;
- **uuid** — глобальный уникальный идентификатор виртуальной машины VMware.

vmware.vm.net.if.in[url,uuid,instance,<mode>]

 Статистика входящего трафика сетевого интерфейса виртуальной машины VMware (байт/пакетов в секунду).
 Возвращаемое значение: *Целое число* [2].

Параметры:

- **url** — URL сервиса VMware;
- **uuid** — глобальный уникальный идентификатор виртуальной машины VMware;
- **instance** — экземпляр сетевого интерфейса;
- **mode** — *bps* (по умолчанию) или *pps* — байты или пакеты в секунду.

vmware.vm.net.if.out[url,uuid,instance,<mode>]

 Статистика исходящего трафика сетевого интерфейса виртуальной машины VMware (байт/пакетов в секунду).
 Возвращаемое значение: *Целое число* [2].

Параметры:

- **url** — URL сервиса VMware;
- **uuid** — глобальный уникальный идентификатор виртуальной машины VMware;
- **instance** — экземпляр сетевого интерфейса;
- **mode** — *bps* (по умолчанию) или *pps* — байты или пакеты в секунду.

vmware.vm.net.if.usage[url,uuid,<instance>]

 Использование сети виртуальной машиной VMware (суммарная скорость передачи и приема) за интервал времени (КБ/с).
 Возвращаемое значение: *Integer*.

Параметры:

- **url** — URL сервиса VMware;
- **uuid** — глобальный уникальный идентификатор виртуальной машины VMware;
- **instance** — экземпляр сетевого интерфейса.

vmware.vm.perfcounter[url,uuid,path,<instance>]

 Значение счётчика производительности виртуальной машины VMware.
 Возвращаемое значение: *Целое число* [2].

Параметры:

- **url** — URL сервиса VMware;
- **uuid** — глобальный уникальный идентификатор виртуальной машины VMware;
- **path** — путь счётчика производительности [1];
- **instance** — экземпляр счётчика производительности. Используйте пустой экземпляр для агрегированных значений (по умолчанию).

vmware.vm.powerstate[url,uuid]

 Состояние питания виртуальной машины VMware.
 Возвращаемое значение: *0* — poweredOff; *1* — poweredOn; *2* — suspended.

Параметры:

- **url** — URL сервиса VMware;
- **uuid** — глобальный уникальный идентификатор виртуальной машины VMware.

vmware.vm.property[url,uuid,prop]

 Свойство виртуальной машины VMware.
 Возвращаемое значение: *String*.

Параметры:

- **url** — URL сервиса VMware;
- **uuid** — глобальный уникальный идентификатор виртуальной машины VMware;
- **prop** — путь к свойству.

vmware.vm.snapshot.get[url,uuid]

 Состояние снимка виртуальной машины VMware.
 Возвращаемое значение: объект JSON.

Параметры:

- **url** — URL сервиса VMware;
- **uuid** — глобальный уникальный идентификатор виртуальной машины VMware.

vmware.vm.state[url,uuid]

 Состояние виртуальной машины VMware.
 Возвращаемое значение: *String*: *notRunning*, *resetting*, *running*, *shuttingDown*, *standby* или *unknown*.

Параметры:

- **url** — URL сервиса VMware;
- **uuid** — глобальный уникальный идентификатор виртуальной машины VMware.

vmware.vm.storage.committed[url,uuid]

 Выделенное пространство хранилища виртуальной машины VMware (в байтах).
 Возвращаемое значение: *Целое число*.

Параметры:

- **url** — URL сервиса VMware;
- **uuid** — глобальный уникальный идентификатор виртуальной машины VMware.

vmware.vm.storage.readoio[url,uuid,instance]

 Среднее количество незавершённых запросов на чтение виртуального диска в течение интервала сбора данных.
 Возвращаемое значение: *Целое число*.

Параметры:

- **url** — URL сервиса VMware;
- **uuid** — глобальный уникальный идентификатор виртуальной машины VMware;
- **instance** — экземпляр дискового устройства.

vmware.vm.storage.totalreadlatency[url,uuid,instance]

 Среднее время, которое занимает чтение с виртуального диска (в миллисекундах).
 Возвращаемое значение: *Integer*.

Параметры:

- **url** — URL сервиса VMware;
- **uuid** — глобальный уникальный идентификатор виртуальной машины VMware;
- **instance** — экземпляр устройства диска.

vmware.vm.storage.totalwritelatency[url,uuid,instance]

 Среднее время, которое занимает запись на виртуальный диск (в миллисекундах).
 Возвращаемое значение: *Integer*.

Параметры:

- **url** — URL сервиса VMware;
- **uuid** — глобальный уникальный идентификатор виртуальной машины VMware;
- **instance** — экземпляр устройства диска.

vmware.vm.storage.uncommitted[url,uuid]

 Незафиксированное пространство хранилища виртуальной машины VMware (в байтах).
 Возвращаемое значение: *Integer*.

Параметры:

- **url** — URL сервиса VMware;
- **uuid** — глобальный уникальный идентификатор виртуальной машины VMware.

vmware.vm.storage.unshared[url,uuid]

 Неразделяемое пространство хранилища виртуальной машины VMware (в байтах).
 Возвращаемое значение: *Integer*.

Параметры:

- **url** — URL сервиса VMware;
- **uuid** — глобальный уникальный идентификатор виртуальной машины VMware.

vmware.vm.storage.writeoio[url,uuid,instance]

 Среднее количество незавершённых запросов записи на виртуальный диск в течение интервала сбора данных.
 Возвращаемое значение: *Integer*.

Параметры:

- **url** — URL сервиса VMware;
- **uuid** — глобальный уникальный идентификатор виртуальной машины VMware;
- **instance** — экземпляр устройства диска.

vmware.vm.tags.get[url,uuid]

 Массив тегов виртуальной машины VMware.
 Возвращаемое значение: объект JSON.

Параметры:

- **url** — URL сервиса VMware;
- **uuid** — глобальный уникальный идентификатор виртуальной машины VMware.

Этот элемент данных работает с vSphere 6.5 и новее.

vmware.vm.tools[url,uuid,<mode>]

 Состояние гостевых инструментов виртуальной машины VMware.
 Возвращаемое значение: *String*.
 В режиме *status*: *guestToolsExecutingScripts* — VMware Tools запускается; *guestToolsNotRunning* — VMware Tools не запущен; *guestToolsRunning* — VMware Tools запущен.
 В режиме *version*: версия.

Параметры:

- **url** — URL сервиса VMware;
- **uuid** — глобальный уникальный идентификатор виртуальной машины VMware;
- **mode** — *version*, *status* (по умолчанию).

vmware.vm.uptime[url,uuid]

 Время непрерывной работы виртуальной машины VMware (в секундах).
 Возвращаемое значение: *Integer*.

Параметры:

- **url** — URL сервиса 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 сервиса VMware;
- **uuid** — глобальный уникальный идентификатор виртуальной машины VMware.

vmware.vm.vfs.dev.read[url,uuid,instance,<mode>]

 Статистика чтения дискового устройства виртуальной машины VMware (байт/операций в секунду).
 Возвращаемое значение: *Integer* [2].

Параметры:

- **url** — URL сервиса VMware;
- **uuid** — глобальный уникальный идентификатор виртуальной машины VMware;
- **instance** — экземпляр дискового устройства;
- **mode** — *bps* (по умолчанию) или *ops* — байты или операции в секунду.

vmware.vm.vfs.dev.write[url,uuid,instance,<mode>]

 Статистика записи дискового устройства виртуальной машины VMware (байты/операции в секунду).
 Возвращаемое значение: *Integer* [2].

Параметры:

- **url** — URL сервиса VMware;
- **uuid** — глобальный уникальный идентификатор виртуальной машины VMware;
- **instance** — экземпляр дискового устройства;
- **mode** — *bps* (по умолчанию) или *ops* — байты или операции в секунду.

vmware.vm.vfs.fs.discovery[url,uuid]

 Обнаружение файловых систем виртуальной машины VMware.
 Возвращаемое значение: объект JSON.

Параметры:

- **url** — URL сервиса VMware;
- **uuid** — глобальный уникальный идентификатор виртуальной машины VMware.

Для работы этого элемента данных на гостевой виртуальной машине должны быть установлены VMware Tools.

vmware.vm.vfs.fs.size[url,uuid,fsname,<mode>]

 Статистика файловой системы виртуальной машины VMware (байты/проценты).
 Возвращаемое значение: *Integer* — для байтов; *Float* — для процентов.

Параметры:

- **url** — URL сервиса VMware;
- **uuid** — глобальный уникальный идентификатор виртуальной машины VMware;
- **fsname** — имя файловой системы;
- **mode** — *total*, *free*, *used*, *pfree* или *pused*.

Для работы этого элемента данных на гостевой виртуальной машине должны быть установлены VMware Tools.

Сноски

[1] См. Создание пользовательских имен счетчиков производительности для VMware.

[2] Значение этих элементов данных получается из счетчиков производительности VMware, а для обновления их данных в кэше VMware Zabbix используется параметр VMwarePerfFrequency:

- 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.

2 Ключевые поля 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 кластеров.	{#CLUSTER.ID}	Идентификатор кластера.
	{#CLUSTER.NAME}	Имя кластера.

	”resource_pool”	Массив, содержащий данные пула ресурсов, включая ID группы ресурсов, массив тегов, путь пула ресурсов, количество виртуальных машин.
		
		Структура массива:
		`[{`
		`"rpid":"id ",`
		`"tags":[{}],`
		`"rpath":" ",`
		`"vm_count":0`
		`}]`

	”tags”	Структуру массива `"tags"` см. в поле ”tags”.
		Массив, содержащий теги с именем тега, описанием и категорией.
		
		Структура массива:
		`[{`
		`"tag":" ",`
		`"tag_description":" ",`
		`"category":" "`
		`}]`

vmware.datastore.discovery		
Выполняет обнаружение хранилищ данных.	{#DATASTORE}	Имя хранилища данных.

	{#DATASTORE.EXTENT}	Массив, содержащий ID раздела extent хранилища данных и имя экземпляра.
		
		Структура массива:
		`[{`
		`"partitionid":1,`
		`"instance":" "`
		`}]`

	{#DATASTORE.TYPE}	Тип хранилища данных. Примеры значений: `VMFS`, `NFS`, `vsan` и т. д.
	{#DATASTORE.UUID}	Идентификатор хранилища данных.

Ключ элемента данных		
	”tags”	Массив, содержащий теги с именем тега, описанием и категорией. Структура массива: [{ "tag":" ", "tag_description":" ", "category":" " }]
vmware.dc.discovery		
Выполняет обнаружение дата-центров.	{#DATACENTER}	Имя дата-центра.
	{#DATACENTERID}	Идентификатор дата-центра.
	”tags”	Массив, содержащий теги с именем тега, описанием и категорией. Структура массива: [{ "tag":" ", "tag_description":" ", "category":" " }]
vmware.dvswitch.discovery		
Выполняет обнаружение распределённых коммутаторов vSphere.	{#DVS.NAME}	Имя коммутатора.
	{#DVS.UUID}	Идентификатор коммутатора.
vmware.hv.discovery		
Выполняет обнаружение гипервизоров.	{#HV.UUID}	Уникальный идентификатор гипервизора.
	{#HV.ID}	Идентификатор гипервизора (имя управляемого объекта HostSystem).
	{#HV.NAME}	Имя гипервизора.
	{#HV.NETNAME}	Сетевое имя узла гипервизора.
	{#HV.IP}	IP-адрес гипервизора, может быть пустым. В случае HA-конфигурации с несколькими сетевыми интерфейсами соблюдается следующий приоритет выбора интерфейса: - предпочтение отдаётся IP-адресу, который находится в той же IP-подсети, что и IP-адрес vCenter; - затем предпочтение отдаётся IP-адресу из IP-подсети со шлюзом по умолчанию; - затем предпочтение отдаётся IP-адресу интерфейса с наименьшим ID.
	{#CLUSTER.NAME}	Имя кластера, может быть пустым.
	{#DATACENTER.NAME}	Имя дата-центра.
	{#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.

Ключ элемента данных		
	”resource_pool”	Массив, содержащий данные пула ресурсов, включая ID группы ресурсов, массив тегов, путь пула ресурсов, количество виртуальных машин. Структура массива: [{ "rpid":"id ", "tags":[{}], "rpath":" ", "vm_count":0 }]
	”tags”	Структуру массива "tags" см. в поле ”tags”. Массив, содержащий теги с именем тега, описанием и категорией. Структура массива: [{ "tag":" ", "tag_description":" ", "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 разделов диска.
	”datastore_extent”	Массив, содержащий имя экземпляра extent хранилища данных и ID раздела. Структура массива: [{ "partitionid":1, "instance":" " }]
	”tags”	Массив, содержащий теги с именем тега, описанием и категорией. Структура массива: [{ "tag":" ", "tag_description":" ", "category":" " }]
vmware.hv.net.if.discovery		
Выполняет обнаружение сетевых интерфейсов гипервизора.	{#IFNAME}	Имя интерфейса.
	{#IFDRIVER}	Драйвер интерфейса.
	{#IFDUPLEX}	Настройки дуплекса интерфейса.

Ключ элемента данных		
	{#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 управляемого объекта VirtualMachine).
	{#VM.NAME}	Имя виртуальной машины.
	{#HV.NAME}	Имя гипервизора.
	{#HV.UUID}	Уникальный идентификатор гипервизора.
	{#HV.ID}	Идентификатор гипервизора (имя управляемого объекта HostSystem).
	{#CLUSTER.NAME}	Имя кластера, может быть пустым.
	{#DATACENTER.NAME}	Имя дата-центра.
	{#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 ОС гостевой виртуальной машины, может быть пустым.
	{#VM.GUESTFULLNAME}	Полное имя ОС гостевой 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 пула ресурсов.
	{#VM.RPOOL.PATH}	Полный путь пула ресурсов без корневого имени ”Resources”. Имена папок объединяются с помощью ”/”.
	{#VM.SNAPSHOT.COUNT}	Количество снимков ВМ.

	”tags”	Массив, содержащий теги с именем тега, описанием и категорией.
		
		Структура массива:
		`[{`
		`"tag":" ",`
		`"tag_description":" ",`
		`"category":" "`
		`}]`

	”vm_customattribute”	Массив пользовательских атрибутов виртуальной машины (если определены).
		
		Структура массива:
		`[{`
		`"name":" ",`
		`"value":" "`
		`}]`

Ключ элемента данных		
	”net_if”	Массив сетевых интерфейсов виртуальной машины. Структура массива: [{ "ifname": " ", "ifdesc": " ", "ifmac": "00:00:00:00:00:00", "ifconnected": true, "iftype": " ", "ifbackingdevice": " ", "ifdvswitch_uuid": "uuid ", "ifdvswitch_portgroup": " ", "ifdvswitch_port": " ", "ifip": ["ip- "] }] 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` — отключён; `true` — подключён).
	{#IFTYPE}	Тип интерфейса.
	{#IFBACKINGDEVICE}	Имя базового устройства.
	{#IFDVSWITCH.UUID}	Уникальный идентификатор vSphere Distributed Switch.
	{#IFDVSWITCH.PORTGROUP}	Группа распределённых 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.

3 Примеры JSON для элементов данных VMware

Обзор В этом разделе представлена дополнительная информация об объектах JSON, 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
        },
        {
            "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
        },
        {
            //
        }
    ]
}
```

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 со следующей структурой (значения приведены в качестве примера):

```
{
    "tags": [
        {
            "name": "Windows",
        "description": "               ",
        "category": "    "
    },
    {
            "name": "SQL Server",
        "description": "                    ",
        "category": "        "
        },
    {
            //
        }
    ]
}
```

vmware.hv.diskinfo.get Элемент данных **vmware.hv.diskinfo.get[]** возвращает объекты JSON со следующей структурой (значения приведены в качестве 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 со следующей структурой (значения приведены в качестве 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 JSON-объекты со следующей структурой (значения приведены в качестве 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"
                 }
            },
            {
                    // other hv hw sensors
                }
        ]
        }
}
```

vmware.hv.sensors.get Элемент данных **vmware.hv.sensors.get[]** возвращает JSON-объекты со следующей структурой (значения приведены в качестве 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"
        },
                {
                        //          hv
                }
        ]
        }
}
```

vmware.vm.snapshot.get Если существуют какие-либо снимки, элемент данных **vmware.snapshot.get[]** возвращает объект JSON со следующей структурой (значения приведены в качестве 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
}
```

Если снимков не существует, 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
}
```

4 Пример настройки мониторинга VMware

Обзор

Следующий пример описывает, как настроить Zabbix для мониторинга виртуальных машин VMware. Это включает:

- создание узла сети, представляющего вашу среду VMware;
- создание правила низкоуровневого обнаружения, которое обнаруживает виртуальные машины в вашей среде VMware;
- создание прототипа узла сети, на основе которого Zabbix будет создавать реальные узлы сети для виртуальных машин, обнаруженных правилом низкоуровневого обнаружения.

Предварительные требования

> **Note:**
> Этот пример не охватывает настройку VMware. Предполагается, что VMware уже настроен.

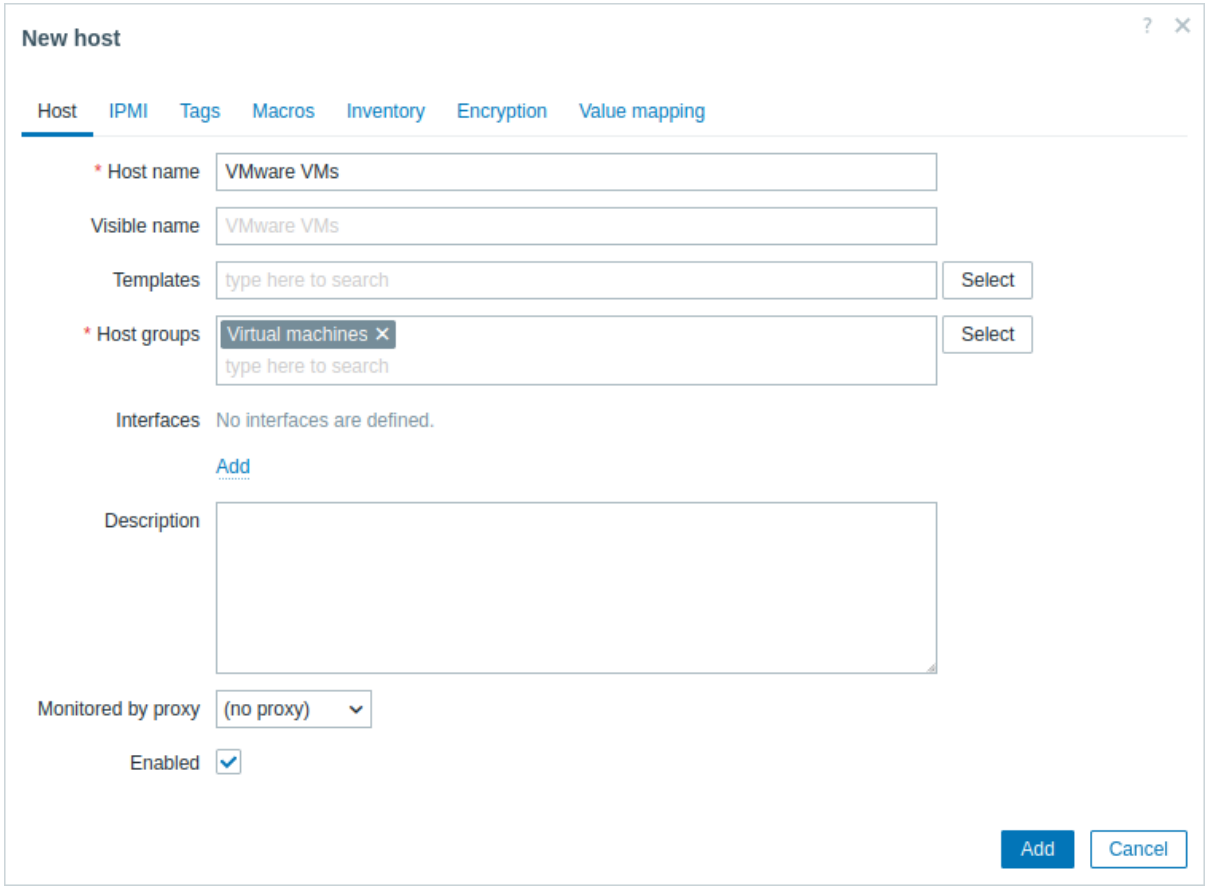
Прежде чем продолжить, установите параметр `StartVMwareCollectors` в конфигурационном файле сервера Zabbix в значение 2 или больше (по умолчанию используется значение 0).

Создание узла сети

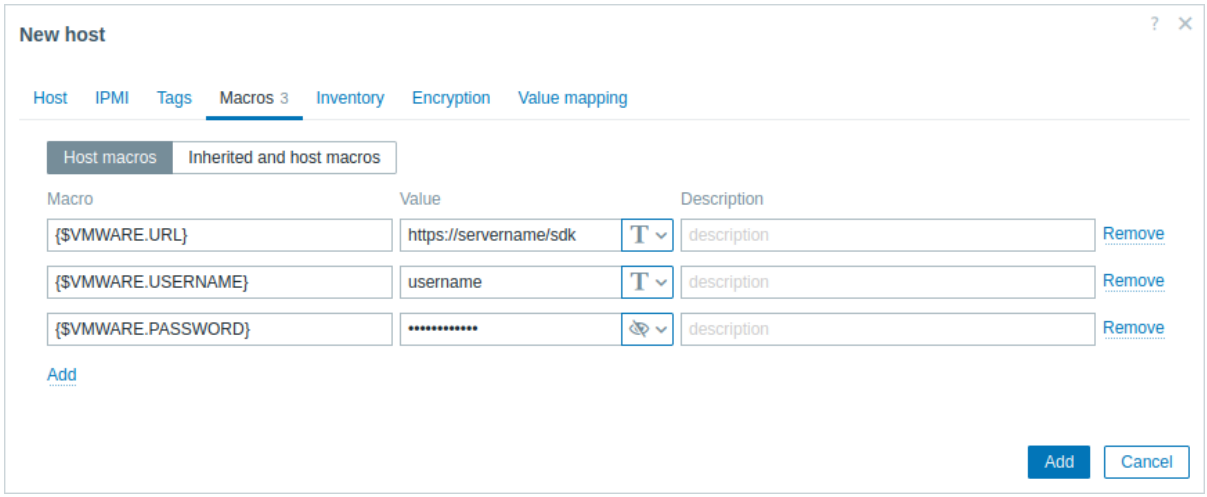
1. Перейдите в *Сбор данных* → *Узлы сети*.

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:
  - {$VMWARE.URL} - URL SDK сервиса VMware (гипервизора ESXi) (https://servername/sdk)
  - {$VMWARE.USERNAME} - имя пользователя сервиса VMware
  - {$VMWARE.PASSWORD} - пароль пользователя {$VMWARE.USERNAME} сервиса VMware



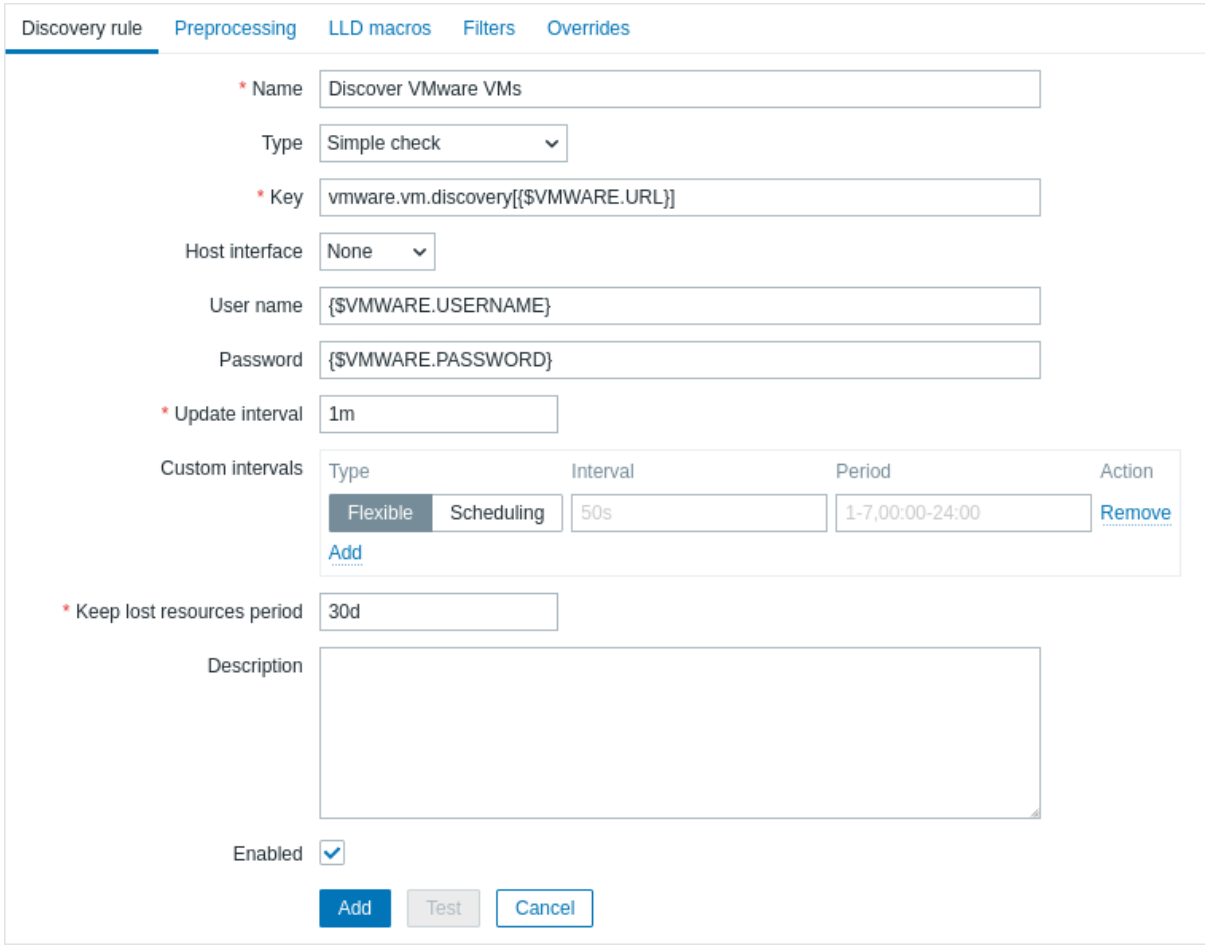
3. Нажмите кнопку *Add*, чтобы создать узел сети. Этот узел сети будет представлять вашу среду VMware.

Создание правила низкоуровневого обнаружения

1. Нажмите *Discovery* для созданного узла сети, чтобы перейти к списку правил низкоуровневого обнаружения для этого узла сети.

2. Создайте правило низкоуровневого обнаружения:

- В поле *Name* введите имя правила низкоуровневого обнаружения (например, «Обнаружение виртуальных машин VMware»).
- В поле *Type* выберите "Simple check".
- В поле *Key* введите встроенный ключ элемента данных для обнаружения виртуальных машин VMware: `vmware.vm.discovery`
- В полях *User name* и *Password* введите соответствующие макросы, ранее настроенные на узле сети.



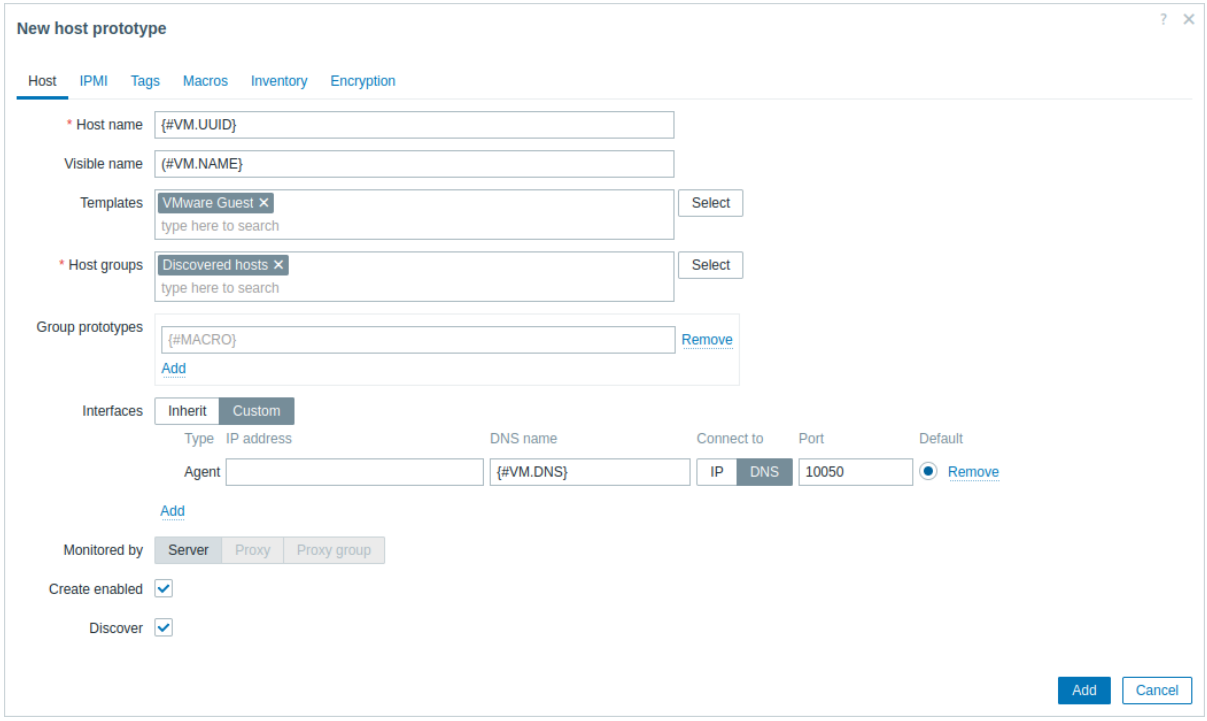
3. Нажмите кнопку *Add*, чтобы создать правило низкоуровневого обнаружения. Это правило обнаружения найдет виртуальные машины в вашей среде VMware.

Создание прототипа узла сети

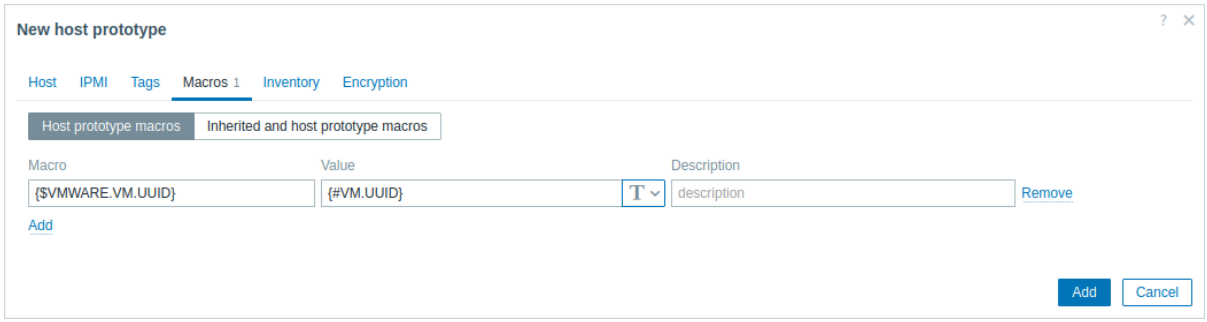
1. В списке правил низкоуровневого обнаружения нажмите *Host prototypes* для ранее созданного правила низкоуровневого обнаружения.

2. Создайте прототип узла сети. Поскольку прототипы узлов сети являются шаблонами для создания узлов сети с помощью 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 *Host name* введите макрос {#VM.UUID}.
- В поле *Visible name* введите макрос {#VM.NAME}.
- В поле *Templates* введите или выберите шаблон ”VMware Guest”. Этот шаблон содержит элементы данных VMware и правила обнаружения для мониторинга состояния питания виртуальной машины, использования CPU, использования памяти, сетевых устройств и т. д.
- В поле *Host groups* введите или выберите группу узлов сети (например, ”Discovered hosts”).
- В поле *Interfaces* добавьте пользовательский интерфейс узла сети. Затем введите макрос {#VM.DNS} в поле *DNS name* или макрос {#VM.IP} в поле *IP address*. Либо, если виртуальные машины в вашей среде VMware имеют несколько интерфейсов, перейдите к разделу *Расширенная настройка интерфейса узла сети*. Настройка пользовательского интерфейса узла сети необходима для корректной работы шаблона *VMware Guest*.



- На вкладке *Macros* задайте макрос {$VMWARE.VM.UUID} со значением {#VM.UUID}. Это необходимо для корректной работы шаблона *VMware Guest*, который использует этот макрос как пользовательский макрос уровня узла сети в параметрах элементов данных (например, *vmware.vm.net.if.discovery[{$VMWARE.URL}, {$VMWARE.VM.UUID}]*).

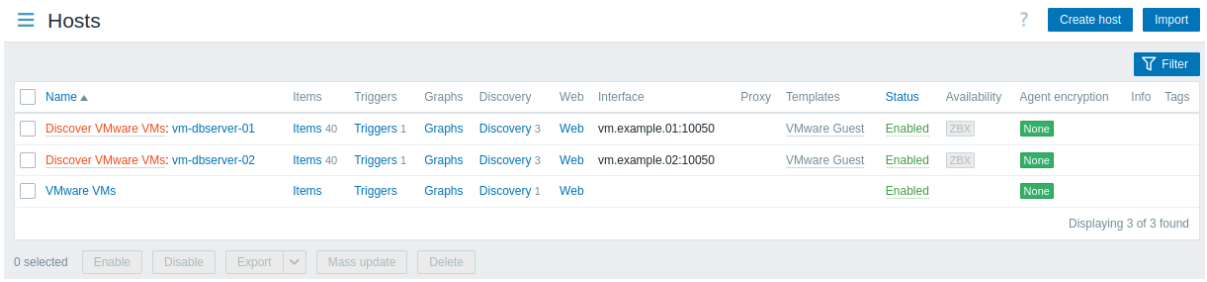


3. Нажмите кнопку *Add*, чтобы создать прототип узла сети. Этот прототип узла сети будет использоваться для создания узлов сети для виртуальных машин, обнаруженных ранее созданным правилом низкоуровневого обнаружения.

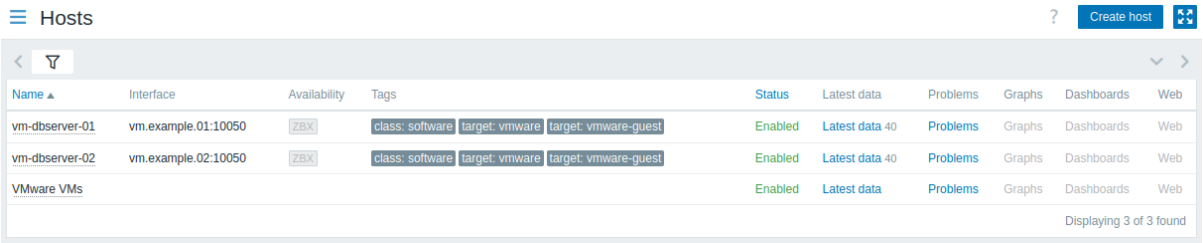
Просмотр узлов сети и метрик

После создания прототипа узла сети правило низкоуровневого обнаружения создаст узлы сети для обнаруженных виртуальных машин VMware, и Zabbix начнет их мониторить. Обратите внимание, что обнаружение и создание узлов сети при необходимости также можно выполнить вручную.

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

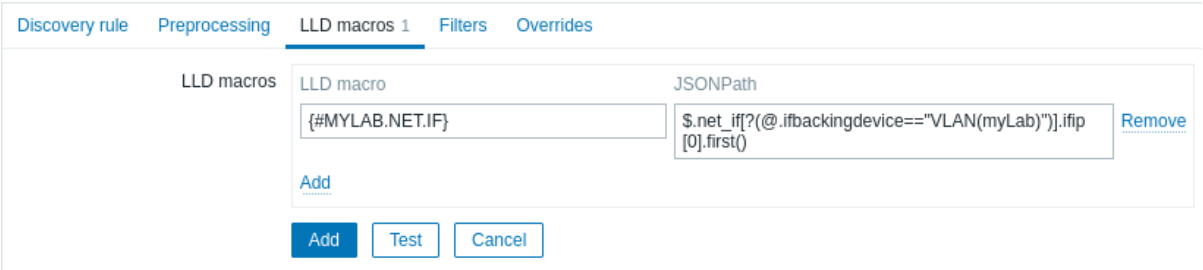
Ключ элемента данных `vmware.vm.discovery[{$VMWARE.URL}]`, настроенный в разделе *Создание правила обнаружения низкого уровня*, возвращает данные о сетевых интерфейсах в поле "net_if":


```
"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": []
      }
    ]
```

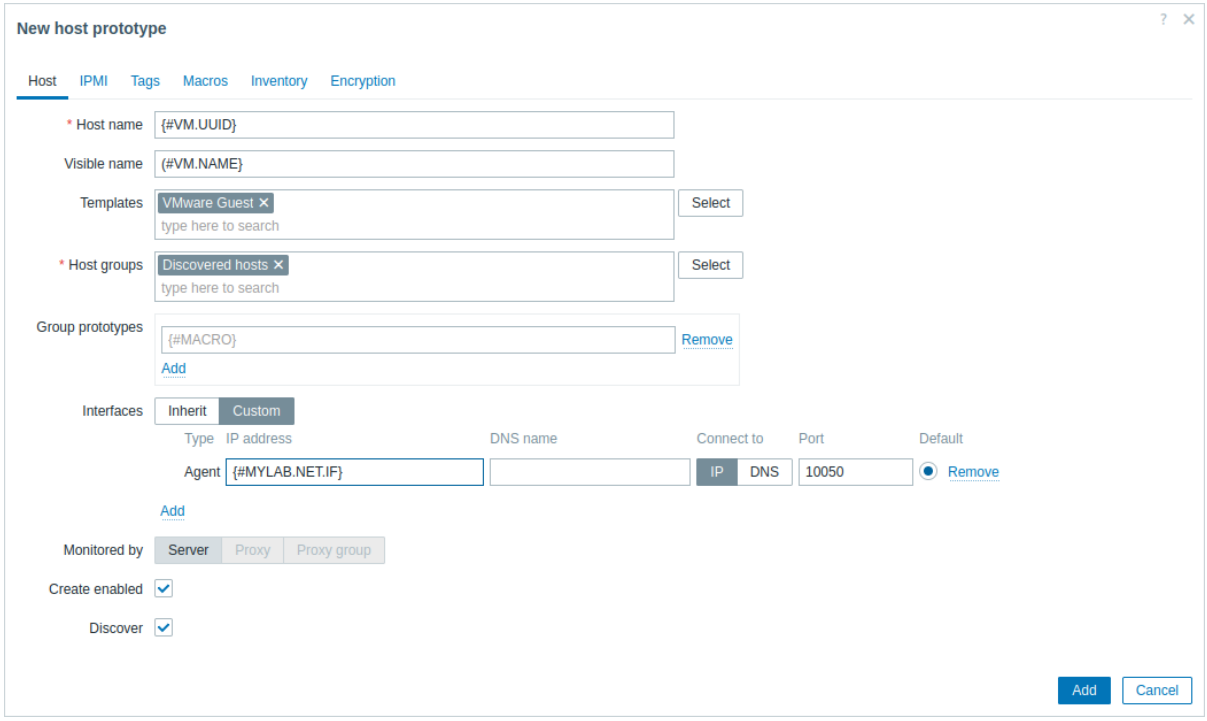

Эти данные можно использовать для настройки пользовательского интерфейса узла сети.

1. При создании правила обнаружения низкого уровня дополнительно настройте макрос обнаружения низкого уровня. На вкладке *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 name* или *IP address*.



9 Обслуживание

Обзор В Zabbix можно определять периоды обслуживания для узлов сети и групп узлов сети.

Кроме того, можно определить обслуживание только для одного триггера (или подмножества триггеров), указав теги триггера. В этом случае обслуживание будет активировано только для этих триггеров; все остальные триггеры узла сети или группы узлов сети не будут находиться в обслуживании.

Существует два типа обслуживания — со сбором данных и без сбора данных.

Во время обслуживания «со сбором данных» триггеры обрабатываются как обычно, и при необходимости создаются события. Однако эскалации проблем для узлов сети/триггеров в обслуживании приостанавливаются, если в настройках действия установлен флажок *Pause operations for suppressed problems*. В этом случае шаги эскалации, которые могут включать отправку уведомлений или удалённые команды, будут игнорироваться на протяжении всего периода обслуживания. Обратите внимание, что операции восстановления и обновления проблем во время обслуживания не подавляются, подавляются только эскалации.

Например, если шаги эскалации запланированы на 0, 30 и 60 минут после начала проблемы, и имеется получасовое обслуживание, продолжающееся с 10-й по 40-ю минуту после возникновения реальной проблемы, то второй и третий шаги будут выполнены на полчаса позже, то есть на 60-й и 90-й минутах (если проблема всё ещё существует). Аналогично, если проблема возникает во время обслуживания, эскалация начнётся после окончания обслуживания.

Чтобы получать уведомления о проблемах во время обслуживания в обычном режиме (без задержки), необходимо снять флажок *Pause operations for suppressed problems* в настройках действия.

Note:
Если хотя бы один узел сети (используемый в выражении триггера) не находится в режиме обслуживания, Zabbix отправит уведомление о проблеме.

Сервер Zabbix должен быть запущен во время обслуживания. Периоды обслуживания пересчитываются каждую минуту или сразу после перезагрузки кэша конфигурации, если в период обслуживания были внесены изменения.

Процессы timer проверяют, нужно ли изменить статус узла сети на/с обслуживания, на 0-й секунде каждой минуты. Кроме того, каждую секунду процесс timer проверяет, нужно ли запускать/останавливать какие-либо обслуживания на основании изменений в [периодах обслуживания] после обновления конфигурации. Таким образом, скорость запуска/остановки периодов обслуживания зависит от интервала обновления конфигурации (по умолчанию 10 секунд). Обратите внимание, что изменения периода обслуживания не включают настройки *Active since/Active till*. Также, если узел сети/группа узлов сети добавляется в уже активный период обслуживания, изменения будут применены процессом timer только в начале следующей минуты.

Обратите внимание, что когда узел сети входит в обслуживание, процессы timer сервера Zabbix считывают все открытые проблемы, чтобы проверить, требуется ли их подавить. Это может повлиять на производительность, если имеется много открытых проблем. Сервер Zabbix также считывает все открытые проблемы при запуске, даже если на тот момент обслуживание не настроено.

Обратите внимание, что сервер Zabbix (или прокси) всегда собирает данные независимо от типа обслуживания (включая обслуживание «без данных»). Позже эти данные игнорируются сервером, если установлен режим «без сбора данных».

Когда обслуживание «без данных» заканчивается, триггеры, использующие функцию nodata(), не сработают до следующей проверки в пределах периода, который они проверяют.

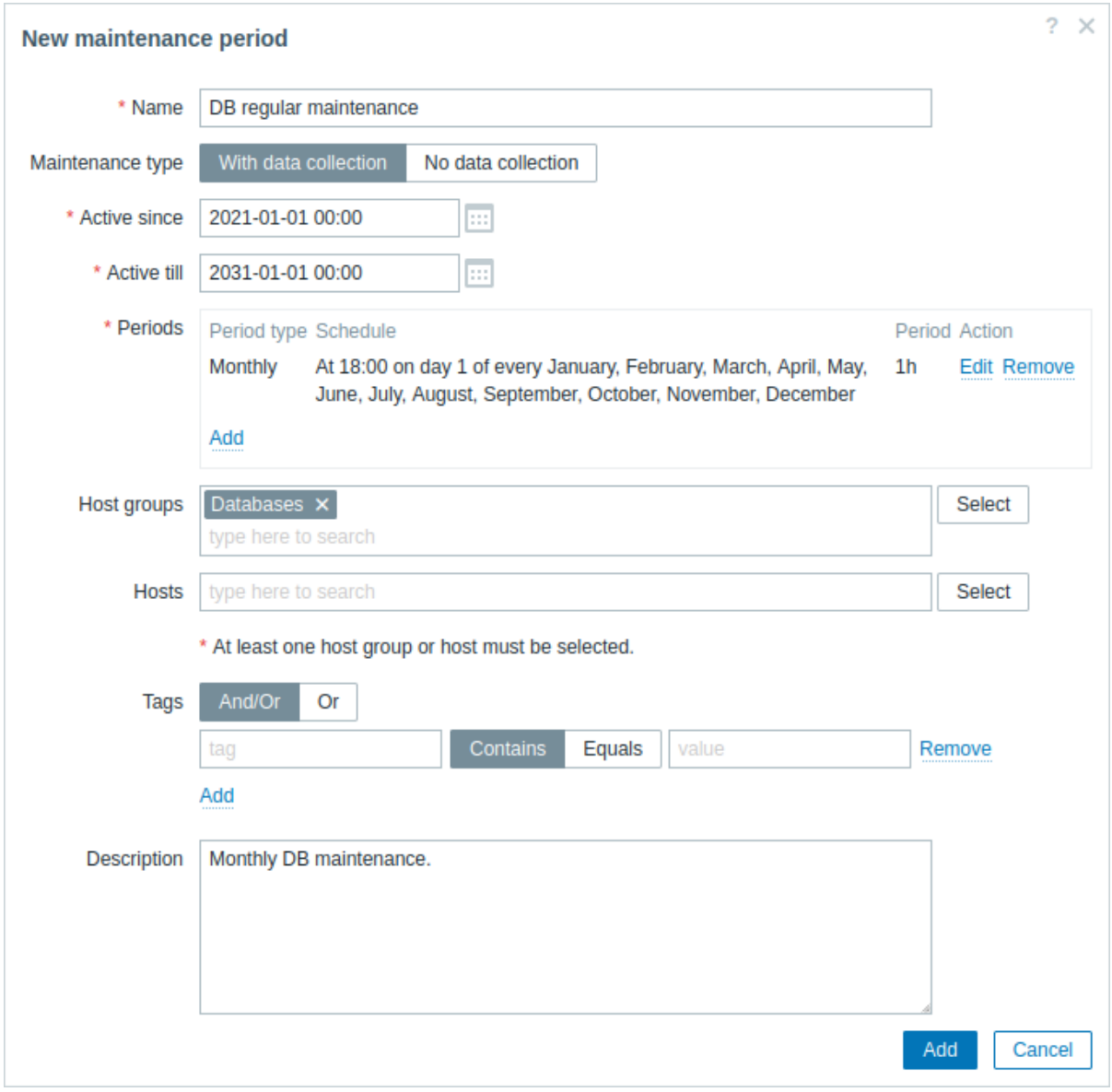
Если элемент данных журнала добавлен, пока узел сети находится в обслуживании, и обслуживание заканчивается, будут собраны только новые записи журнала с момента окончания обслуживания.

Если для узла сети, находящегося в обслуживании типа «без данных», отправляется значение с временной меткой (например, с использованием Zabbix sender), это значение будет отброшено, однако можно отправить значение с временной меткой для уже истёкшего периода обслуживания, и оно будет принято.

Если пользователь изменяет период обслуживания, узлы сети, группы, или теги, изменения вступят в силу только после синхронизации кэша конфигурации.

Настройка Чтобы настроить период обслуживания:

1. Перейдите в: *Сбор данных* > *Обслуживание*.
2. Нажмите *Создать период обслуживания* (или на имя существующего периода обслуживания).
3. Введите параметры обслуживания в форме.

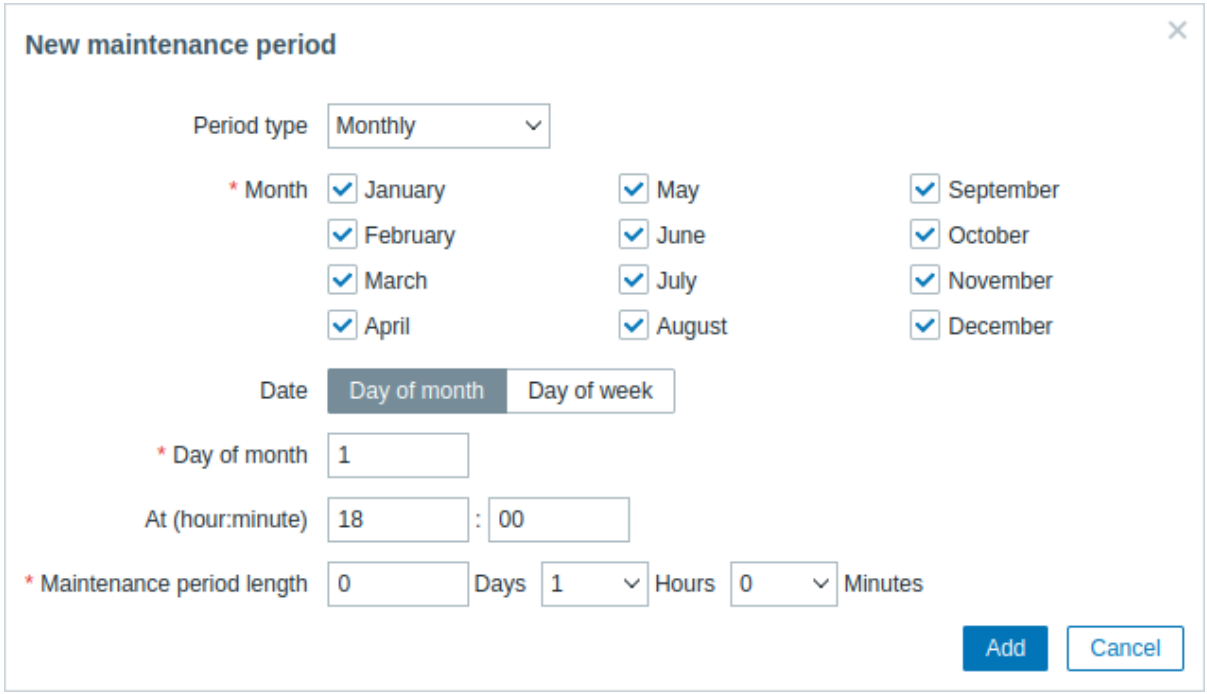


Все обязательные поля ввода отмечены красной звёздочкой.

Параметр	Описание
Имя	Имя периода обслуживания.
Тип обслуживания	Можно задать два типа обслуживания: **Со сбором данных** — данные будут собираться сервером во время обслуживания, триггеры будут обрабатываться; **Без сбора данных** — данные не будут собираться сервером во время обслуживания. См. Влияние периодов обслуживания, чтобы узнать, как каждый тип влияет на отчёт о доступности.
Активно с	Дата и время, с которых выполнение периодов обслуживания становится активным. *Примечание:* установка только этого времени не активирует период обслуживания; периоды обслуживания должны быть настроены в *Периодах* (см. ниже).
Активно до	Дата и время, после которых выполнение периодов обслуживания перестаёт быть активным.
Периоды	Этот блок позволяет определить точные дни и часы, когда выполняется обслуживание. Нажатие на Add открывает всплывающее окно с гибкой формой *Период обслуживания*, где можно определить 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, поэтому при вводе в нём будет отображаться выпадающий список всех доступных групп узлов сети. Указание родительской группы узлов сети неявно выбирает все вложенные группы узлов сети. Таким образом, обслуживание также будет активировано на узлах сети из вложенных групп.
Узлы сети	Выберите узлы сети, для которых будет активировано обслуживание. Это поле поддерживает автодополнение, поэтому при вводе в нём будет отображаться выпадающий список всех доступных узлов сети.
Теги	Укажите теги для подавления проблем с совпадающими тегами на узлах сети, находящихся на обслуживании. Можно задать несколько условий. Сопоставление имён тегов всегда чувствительно к регистру. Для каждого условия доступны два оператора: **Содержит** — включить указанные имена тегов, где значения тегов содержат введённую строку (поиск подстроки, с учётом регистра); **Равно** — включить указанные имена и значения тегов (с учётом регистра). Для условий доступны два типа вычисления: **И/Или** — должны быть выполнены все условия, условия с одинаковым именем тега будут сгруппированы по условию Или; **Или** — достаточно выполнения одного условия. Теги можно указывать только если выбран режим *Со сбором данных*.
Описание	Описание периода обслуживания.

Периоды обслуживания

Окно периода обслуживания предназначено для планирования времени для повторяющегося или разового обслуживания. Форма является динамической, и доступные поля изменяются в зависимости от выбранного *Типа периода*.



Тип периода	Описание
Только один раз	Настройте разовый период обслуживания: *Дата* — дата и время периода обслуживания; *Длительность периода обслуживания* — как долго обслуживание будет активно.
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. Если 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)* установлено в "23:00", тогда первый период обслуживания начнется 1 января в 23:00, а второй период обслуживания начнется 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)* установлено в "01:00", тогда первый период обслуживания начнется 3 января в 01:00, а второй период обслуживания начнется 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 выполняться обслуживание; *В (час:минута)* — время суток, когда начинается обслуживание; *Длительность периода обслуживания* — как долго обслуживание будет активно. Если 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 выполняться в одну и ту же дату каждого месяца (например, в 1-й день каждого месяца), затем выберите нужный день в появившемся поле *День месяца*; *Дата:* ***День недели*** — выберите этот вариант, если обслуживание должно выполняться только в определенные дни (например, каждый первый понедельник месяца), затем выберите (в выпадающем списке) нужную неделю месяца (первую, вторую, третью, четвертую или последнюю), а затем отметьте флажки для дня(дней) 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 (*Ежедневно*, *Еженедельно*, *Ежемесячно*) используется часовой пояс сервера Zabbix. Чтобы обеспечить предсказуемое поведение повторяющихся периодов обслуживания, необходимо использовать общий часовой пояс для всех компонентов Zabbix.

Когда закончите, нажмите *Добавить*, чтобы добавить период обслуживания в блок *Периоды*.

Обратите внимание, что переходы 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 изменяется с 02:00 на 03:00, обслуживание продолжится ещё один час (до 04:00);
- если после двух часов обслуживания (в 03:00) происходит переход на летнее/зимнее время и текущее время изменяется с 03:00 на 02:00, обслуживание остановится, поскольку прошло два часа;
- если период обслуживания начинается в течение часа, который пропускается при переходе на летнее/зимнее время, то обслуживание не начнётся.

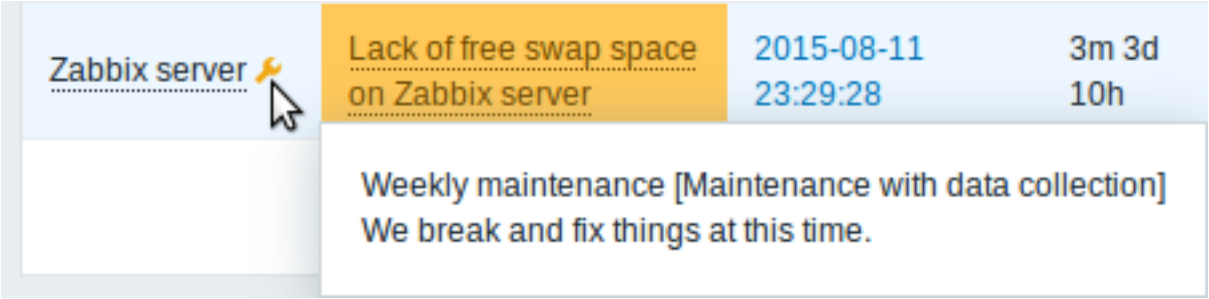
Если период обслуживания задан как "1 день" (фактическая длительность обслуживания составляет 24 часа, поскольку Zabbix рассчитывает дни в часах), начинается в 00:00 и заканчивается в 00:00 следующего дня:

- обслуживание остановится в 01:00 следующего дня, если текущее время переводится на один час вперёд;
- обслуживание остановится в 23:00 этого дня, если текущее время переводится на один час назад.

Просмотр Отображение узлов сети в обслуживании

Оранжевый значок гаечного ключа рядом с именем узла сети указывает, что этот узел сети находится в обслуживании в следующих разделах:

- *Панели*
- *Мониторинг* > *Проблемы*
- *Инвентаризация* > *Узлы сети* > *Сведения об инвентаризации узла сети*
- *Сбор данных* > *Узлы сети* (см. столбец «Состояние»)



Сведения об обслуживании отображаются при наведении указателя мыши на значок.

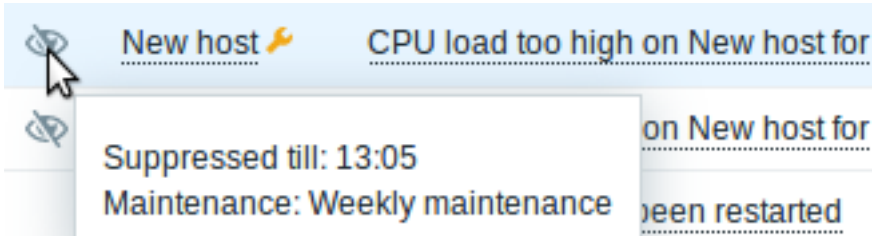
Кроме того, узлы сети, находящиеся в обслуживании, выделяются оранжевым фоном в разделе *Мониторинг* > *Карты*.

Отображение подавленных проблем

Обычно проблемы для узлов сети, находящихся на обслуживании, подавляются, то есть не отображаются в веб-интерфейсе. Однако также можно настроить отображение подавленных проблем, выбрав опцию *Показывать подавленные проблемы* в следующих местах:

- *Панели* (в настройках виджетов *Узлы сети с проблемами*, *Проблемы*, *Проблемы по важности*, *Обзор триггеров*)
- *Мониторинг* > *Проблемы* (в фильтре)
- *Мониторинг* > *Карты* (в настройках карты)
- Глобальные уведомления (в настройках профиля пользователя)

Когда отображаются подавленные проблемы, показывается следующий значок: . При наведении указателя мыши на значок отображаются дополнительные сведения.



Расчет очередей во время обслуживания Очереди, отображаемые в веб-интерфейсе Zabbix (Administration > Queue), рассчитываются сервером Zabbix. Они не включают элементы данных, находящиеся на обслуживании без сбора данных — длина очереди для таких элементов данных всегда равна нулю, даже если получение их значений задерживается. Задержанные элементы данных на обслуживании со сбором данных по-прежнему учитываются в очереди.

Zabbix прокси не знает о периодах обслуживания, поскольку между сервером Zabbix и прокси не выполняется синхронизация конфигурации обслуживания. Внутренние проверки, рассчитываемые на прокси Zabbix (например, `zabbix[queue,,]` и `zabbix[stats,,,queue,,]`), сообщают о задержанных элементах данных независимо от статуса обслуживания на сервере Zabbix.

В результате веб-интерфейс Zabbix и внутренние проверки на прокси Zabbix могут сообщать разную длину очереди для одних и тех же элементов данных, находящихся на обслуживании без сбора данных.

10 Регулярные выражения

Обзор Zabbix поддерживает Perl совместимые регулярные выражения (PCRE, PCRE2).

Имеется два варианта использования регулярных выражений в Zabbix:

- ввод регулярного выражения вручную
- использование глобальных регулярных выражений созданных в Zabbix

Регулярные выражения Вы можете вручную ввести регулярное выражение в поддерживаемых местах. Обратите внимание, что выражение не может начинаться с @, поскольку этот символ используется в Zabbix для ссылок на глобальные регулярные выражения.

> **Warning:**
> При использовании регулярных выражений возможно переполнение стека. Дополнительную информацию смотрите на странице руководства pcrestack.

Обратите внимание, что при многострочном сопоставлении якоря ^ и $ соответствуют началу/концу каждой строки соответственно, а не началу/концу всей строки.

Смотрите также примеры правильного экранирования в различных контекстах.

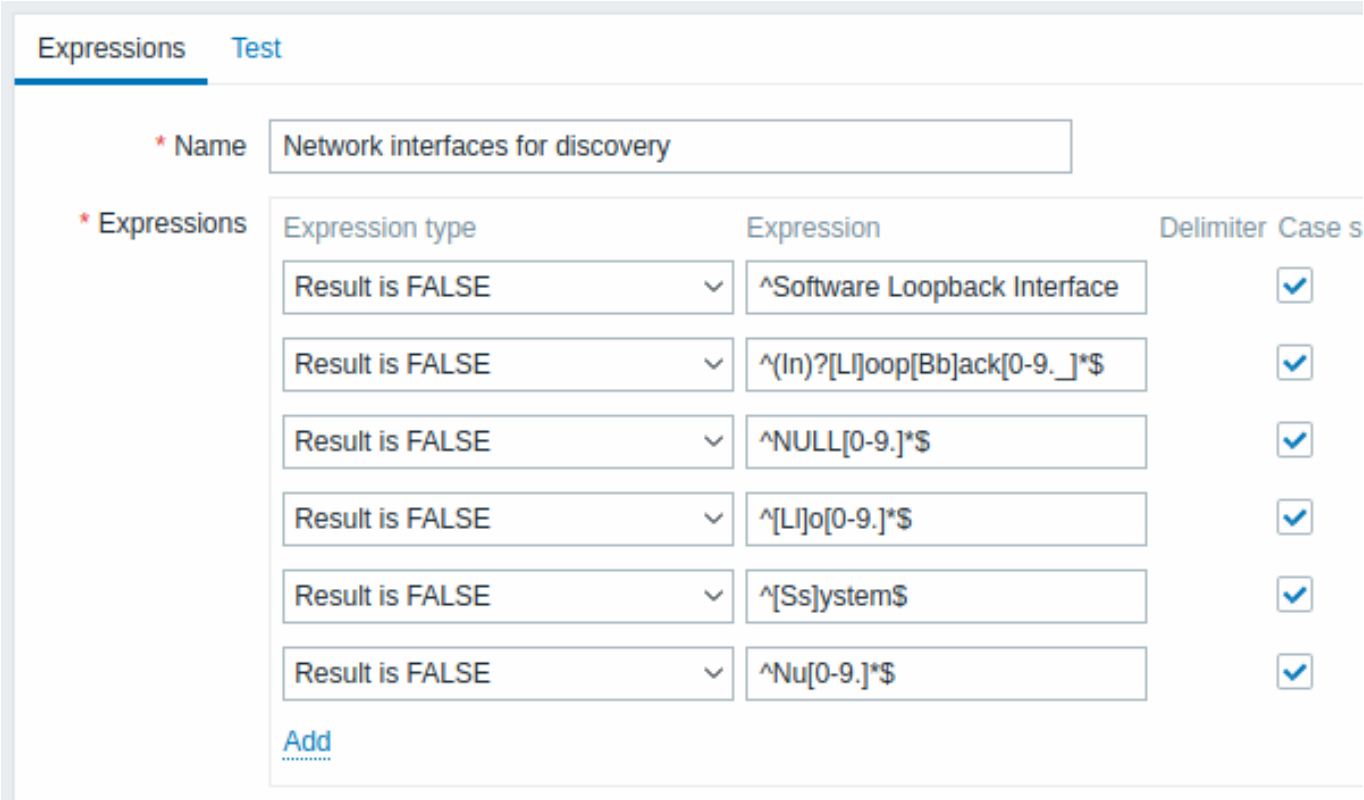
Глобальные регулярные выражения В Zabbix веб-интерфейсе имеется продвинутый редактор для создания и тестирования сложных регулярных выражений.

После того как регулярное выражение создано, его можно использовать в нескольких местах веб-интерфейса, используя ссылку на его имя с префиксом @, например, *@мое_пользовательское_выражение*.

Для создания глобального регулярного выражения:

- Перейдите в: *Администрирование* → *Общие*
- Выберите *Регулярные выражения* в выпадающем списке
- Нажмите на *Новое регулярное выражение*

Вкладка **Выражения** позволяет указать имя регулярного выражения и добавить подвыражения.



Все обязательные поля ввода отмечены красной звёздочкой.

Параметр	Описание
Имя	Укажите имя регулярного выражения. Разрешены любые Unicode символы.
Выражения	Нажмите на *Добавить* в Блоке выражений, чтобы добавить новое подвыражение.
Тип выражения	Выберите тип выражения: **Строка символов включена** - совпадение по подстроке **Любая строка символов включена** - совпадение по любой из подстрок из списка с разделителями. Разделители в списке включают в себя запятую (,), точку (.) или прямую косую черту (/). **Строка символов не включена** - совпадение по любой строке за исключением подстроки **Результат ПРАВДА** - совпадение с регулярным выражением **Результат ЛОЖЬ** - отсутствие совпадения с регулярным выражением
Выражение	Укажите подстроку/регулярное выражение.
Разделитель	Запятая (,), точка (.) или прямая косая черта (/) для разделения текстовых строк в регулярном выражении. Этот параметр доступен только когда выбран режим типа выражения *"Любоая строка символов включена"*.
Регистрозависимое	Отметьте эту опцию чтобы указать является ли регулярное выражение чувствительным к регистру букв.

Косая черта (/) в выражении воспринимается буквально, нежели как разделитель. Таким образом можно сохранять выражения содержащие косую черту, без ошибок.

Attention:
В Zabbix имя пользовательского регулярного выражения может содержать запятые, пробелы и т.п. В этих случаях, там где имеется вероятность неверной интерпретации при использовании ссылки (например, запятая в параметре ключа элемента данных) нужно всю ссылку заключить в кавычки, вот таким образом: *"@Мое выражение для цели, цели2"*.
В других местах (например, в свойствах LLD правил) имена регулярных выражение нельзя заключать в кавычки.

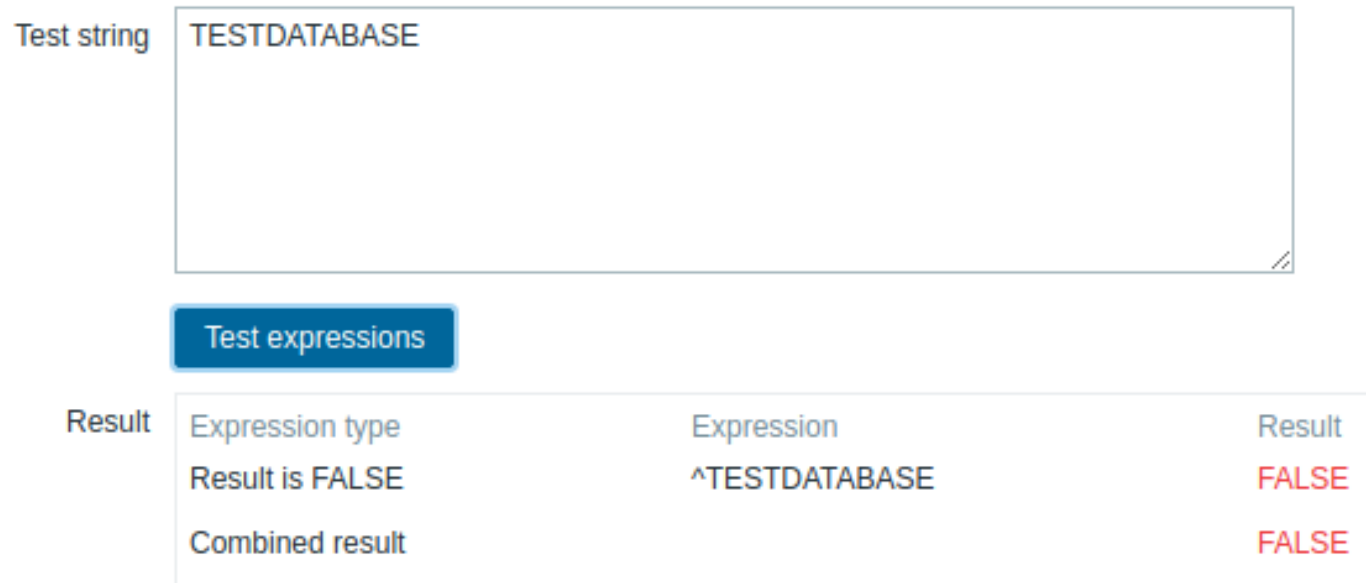
Глобальные регулярные выражения по умолчанию Zabbix поставляется с несколькими глобальными регулярными выражениями в своем наборе данных по умолчанию.

Name	Expression	Matches
Файловые системы для обнаружения	^(btrfs|ext2|ext3|ext4|jfs|reiser|xfs|ffs|ufs|jfs|jfs2|vxfs|hfs|refs|apfs|ntfs|fat32|zfs)	”btrfs” или ”ext2” или ”ext3” или ”ext4” или ”jfs” или ”reiser” или ”xfs” или ”ffs” или ”ufs” или ”jfs” или ”jfs2” или ”vxfs” или ”hfs” или ”refs” или ”apfs” или ”ntfs” или ”fat32” или ”zfs”
Сетевые интерфейсы для обнаружения	^Software Loopback Interface	Строки, начинающиеся с ”Software Loopback Interface”.
	^lo$	”lo”
	^(In)?[Ll]oop[Bb]ack[0-9._]*$	Строки, которые могут начинаться с ”In”, затем содержат ”L” или ”l”, затем ”oop”, затем ”B” или ”b”, затем ”ack”, после чего опционально может следовать любое количество цифр, точек или символов подчеркивания.
	^NULL[0-9.]*$	Строки, начинающиеся с ”NULL”, после которых опционально может следовать любое количество цифр или точек.
	^[Ll]o[0-9.]*$	Строки, начинающиеся с ”Lo” или ”lo”, после которых опционально может следовать любое количество цифр или точек.
	^[Ss]ystem$	”System” или ”system”
	^Nu[0-9.]*$	Строки, начинающиеся с ”Nu”, после которых опционально может следовать любое количество цифр или точек.
Устройства хранения для SNMP-обнаружения	^(Physical memory|Virtual memory|Memory buffers|Cached memory|Swap space)$	”Physical memory” или ”Virtual memory” или ”Memory buffers” или ”Cached memory” или ”Swap space”
Имена служб Windows для обнаружения	^(MMCSS|gupdate|SysmonLog|clr_optimization_v2.0.50727_32|clr_optimization_v4.0.30319_32)$	”MMCSS” или ”gupdate” или ”SysmonLog” или строки вида ”clr_optimization_v2.0.50727_32” и ”clr_optimization_v4.0.30319_32”, где вместо точек можно использовать любой символ, кроме символа новой строки.
Состояния запуска служб Windows для обнаружения	^(automatic|automatic delayed)$	”automatic” или ”automatic delayed”

Примеры

Пример Использование следующего регулярного выражения в LLD для обнаружения баз данных не будет принимать во внимание базу данных с заданным именем:

```
^TESTDATABASE$
```

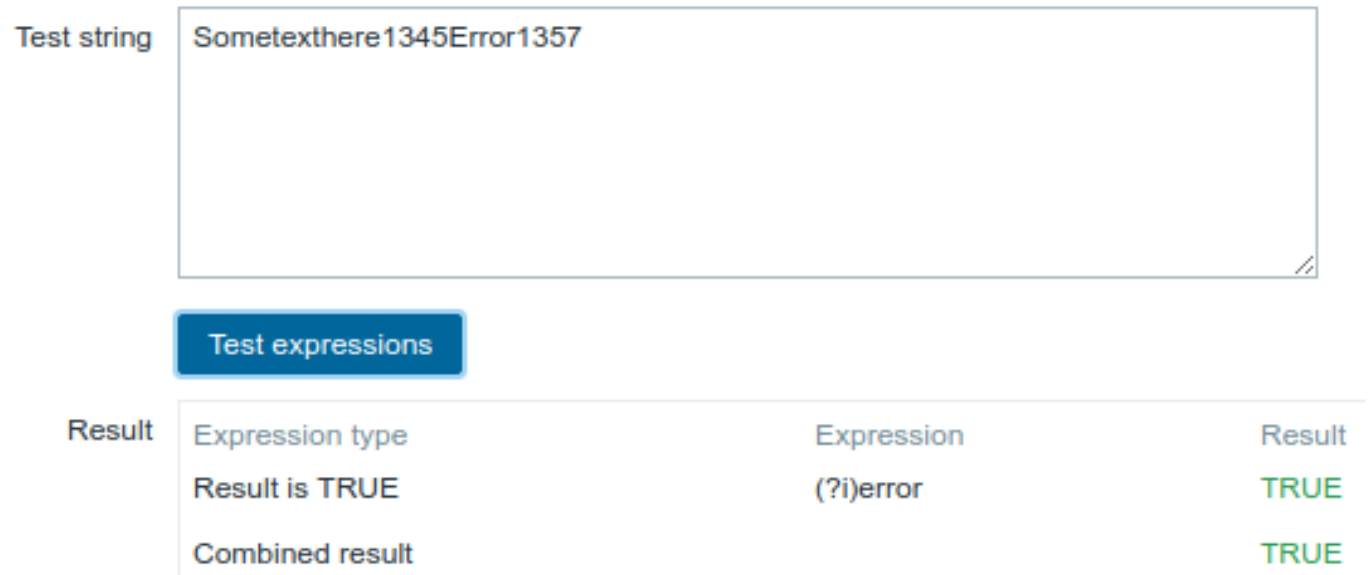


Выбран *Тип выражения*: ”Результат ЛОЖЬ”. Не соответствует имени, содержащему строку *”TESTDATABASE”*.

Пример с использованием inline модификатора регулярного выражения

Использование следующего регулярного выражения, которое включает inline модификатор (?i) для поиска соответствия символов ”error”:

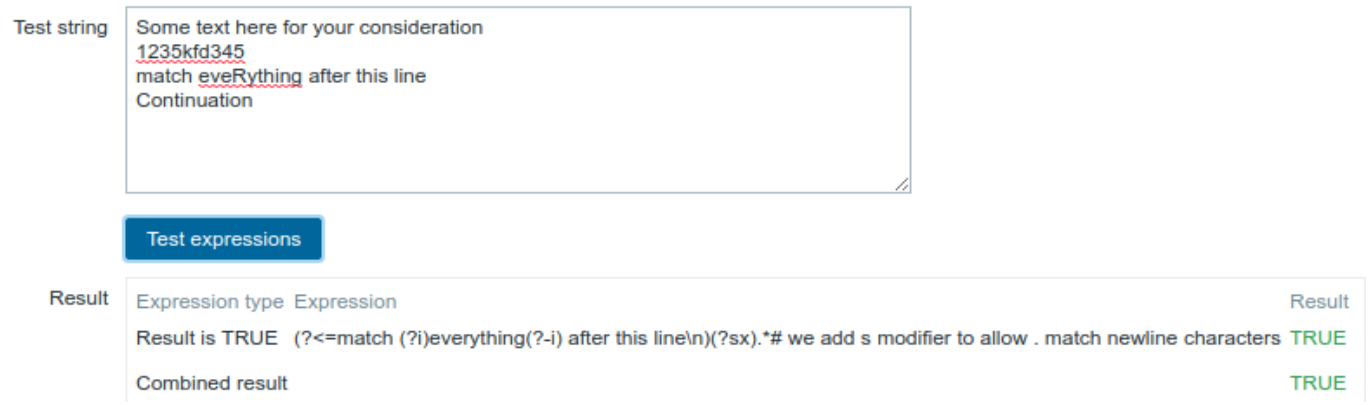
```
(?i)error
```



Выбран *Тип выражения*: ”Результат ПРАВДА”. Есть соответствие по символам ”error”.

Другой пример с использованием inline модификатора регулярного выражения Использование следующего регулярного выражения, которое включает несколько inline модификаторов для поиска соответствия символов после указанной строки:

```
(?<=match (?i)everything(?-i) after this line\n)(?sx).*#        s       ,           .
```



Выбран Тип выражения: ”Релультат ПРАВДА”. Символы после указанной строки совпадают.

Attention:
Модификатор **g** нельзя указывать в строке. Список доступных модификаторов можно найти на странице помощи по pcresyntax [en]. Для получения более подробной информации о синтаксисе PCRE, пожалуйста, обратитесь к PCRE HTML документации [en].

Поддержка регулярных выражений по расположению

Расположение	Регулярное выражение	Глобальное регулярное выражение	Многострочное сопоставление	Комментарии
Элементы данных агента				
eventlog[]	Yes	Yes	Yes	параметры `regexp`, `severity`, `source`, `eventid`
eventlog.count[]				параметры `regexp`, `severity`, `source`, `eventid`
log[]				параметр `regexp`
log.count[]				
logrt[]		Yes/No		параметр `regexp` поддерживает оба варианта, параметр `file_regexp` поддерживает только неглобальные выражения
logrt.count[]				
proc.cpu.util[]		No	No	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`
vfs.dir.size[]				параметры `regex_incl`, `regex_excl`, `regex_excl_dir`
vfs.file.regexp[]			Yes	параметр `regexp`
vfs.file.regmatch[]				
web.page.regexp[]				
SNMP traps				
snmptrap[]	Yes	Yes	No	параметр `regexp`

Расположение	Регулярное выражение	Глобальное регулярное выражение	Многострочное сопоставление	Комментарии
Предобработка значений элементов данных	Yes	No	No	параметр `pattern`
Функции для триггеров/вычисляемых элементов данных				
count()	Yes	Yes	Yes	параметр `pattern`, если параметр `operator` имеет значение *regexp* или *iregexp*
countunique()	Yes	Yes		
find()	Yes	Yes		
logeventid()	Yes	Yes	No	параметр `pattern`
logsource()				
Обнаружение низкого уровня				
Фильтры	Yes	Yes	No	поле *Regular expression*
Переопределения	Yes	No		В параметрах *matches*, *does not match* для условий *Operation*
Условия действий	Yes	No	No	В параметрах *matches*, *does not match* для условий авторегистрации *Host name* и *Host metadata*
Скрипты	Yes	Yes	No	поле *Input validation rule*
Веб-мониторинг	Yes	No	Yes	*Variables* с префиксом **regex:** поле *Required string*
Контекст пользовательского макроса	Yes	No	No	В контексте макроса с префиксом **regex:**
Функции макросов				
regsub()	Yes	No	No	параметр `pattern`
iregsub()				
Индикаторы связей на картах	Yes	No	No	поле *Pattern* (для текстовых элементов данных)
Сопоставление иконок	Yes	Yes	No	поле *Expression*
Сопоставление значений	Yes	No	No	поле *Value*, если тип сопоставления — `regexp`

11 Подтверждение проблемы

Обзор Проблемные события в Zabbix могут быть подтверждены пользователями.

Если пользователь получает уведомление о проблемном событии, он может перейти в интерфейс Zabbix, открыть всплывающее окно обновления проблемы этой проблемы одним из перечисленных ниже способов и подтвердить проблему. При подтверждении они могут ввести свой комментарий к ней, сказав, что они работают над ней или что-то еще, что они могут захотеть сказать об этом.

Таким образом, если другой пользователь системы обнаружит ту же проблему, он сразу увидит, была ли она подтверждена, и комментарии к ней на данный момент.

Таким образом, рабочий процесс решения проблем с более чем одним пользователем системы может осуществляться скоординированным образом.

Статус подтверждения также используется при определении действий операций. Вы можете определить, например, что уведомление отправляется менеджеру более высокого уровня, только если событие не подтверждено в течение некоторого времени.

Чтобы подтвердить события и прокомментировать их, пользователь должен иметь как минимум разрешения на чтение соответствующих триггеров. Чтобы изменить серьезность проблемы или закрыть проблему, пользователь должен иметь разрешения на чтение и запись соответствующих триггеров.

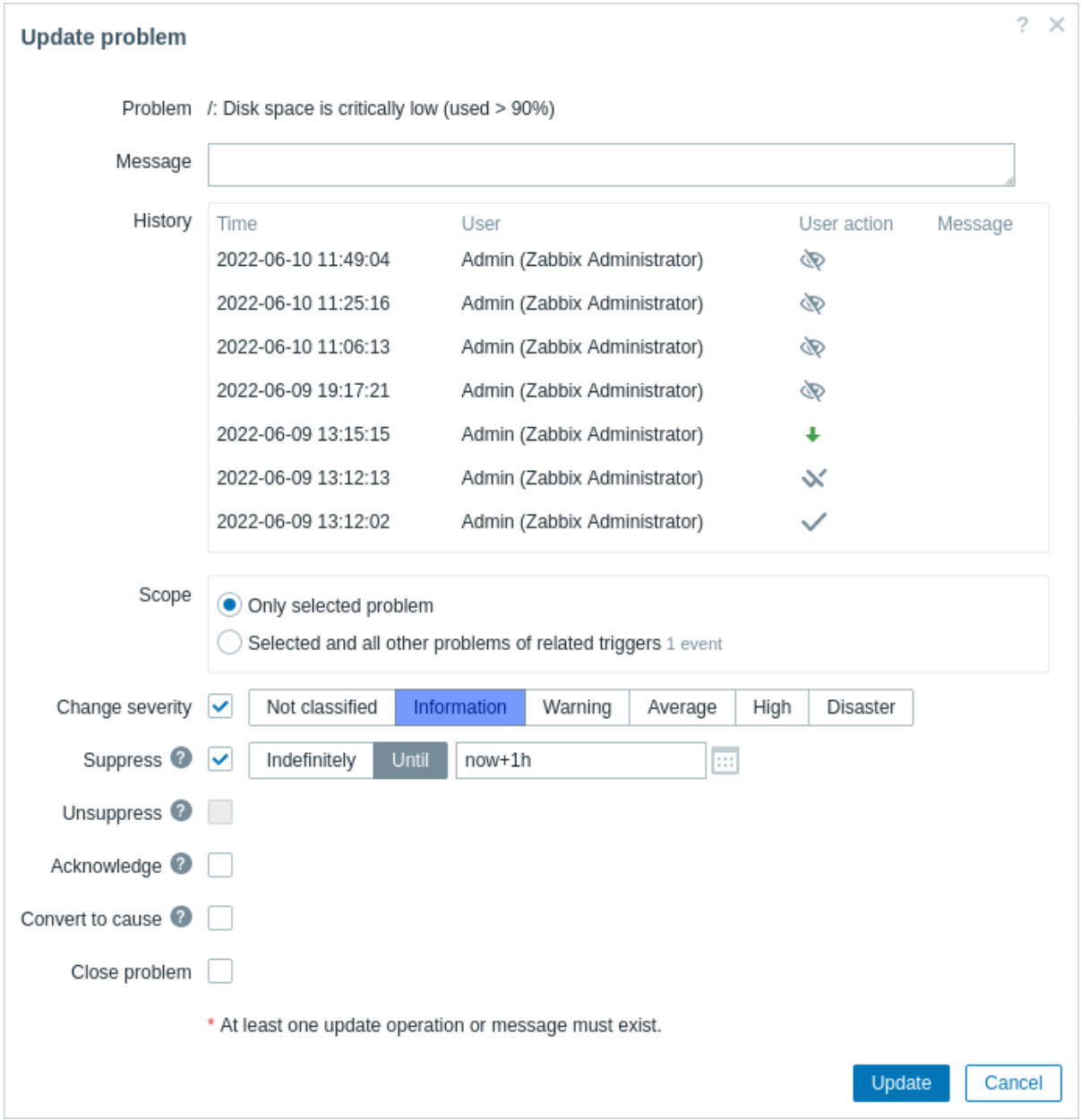
Существует **несколько** способов доступа к всплывающему окну обновления проблемы, что позволяет подтвердить проблему.

- Вы можете выбрать проблемы в *Мониторинг* → *Проблемы*, а затем нажать на *Массовое обновление* под списком
- Вы можете нажать на *Обновление* в столбце *Обновление* проблемы в:
- *Панели мониторинга* (виджеты *Проблемы* и *Проблемы по серьезности*)
- *Мониторинг* → *Проблемы*
- *Мониторинг* → *Проблемы* → *Подробности события*
- Вы можете нажать на ячейку нерешенной проблемы в:
- *Панели мониторинга* (виджет *Обзор триггеров*)

Всплывающее меню содержит опцию *Обновление*, которая перенесет вас в окно обновления проблемы.

Обновление проблем Всплывающее окно обновления проблемы позволяет:

- комментировать проблему
- просматривать комментарии и действия на данный момент
- изменять серьезность проблемы
- скрывать/отменять скрытие проблемы
- подтверждать/отменять подтверждение проблемы
- изменять симптом проблемы на причину проблемы
- вручную закрывать проблему



Все обязательные поля ввода отмечены красной звездочкой.

Параметр	Описание
Проблема	Если выбрана только одна проблема, отображается название проблемы. Если выбрано несколько проблем, отображается *N выбранных проблем*.
Сообщение	Введите текст для комментария к проблеме (максимум 2048 символов).
История	Предыдущие действия и комментарии по проблеме перечислены вместе со временем и данными пользователя. Значение значков, используемых для обозначения действий пользователя, см. на странице сведения о событии. Обратите внимание, что история отображается, если для обновления выбрана только одна проблема.
Область	Определите область таких действий, как изменение серьезности, подтверждение или ручное закрытие проблем: **Только выбранная проблема** — повлияет только на это событие **Выбранные и все другие проблемы связанных триггеров** — в случае подтверждения/закрытия проблемы повлияет на это событие и все другие проблемы, которые еще не подтверждены/закрыты. Если область содержит уже подтвержденные или закрытые проблемы, эти проблемы не будут подтверждаться/закрываться повторно. С другой стороны, количество операций изменения сообщений и серьезности не ограничено.

Параметр	Описание
Изменить серьезность	Отметьте флажок и нажмите кнопку серьезности, чтобы обновить серьезность проблемы. Флажок для изменения серьезности доступен, если разрешения на чтение и запись существуют хотя бы для одной из выбранных проблем. Только те проблемы, которые доступны для чтения и записи, будут обновлены при нажатии на *Обновить*. Если разрешения на чтение и запись отсутствуют ни для одного из выбранных триггеров, флажок будет отключен.
Подавить	Отметьте флажок, чтобы подавить проблему: **Бессрочно** - подавить бессрочно **До** - подавить до указанного времени. Поддерживаются как абсолютные, так и относительные 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 абсолютной даты/времени Обратите внимание, что простой период (например, `1d`, `1w`) не поддерживается. Доступность этой опции зависит от настроек роли пользователя «Подавлять проблемы». См. также: Подавление проблем
Отменить подавление	Отметьте флажок, чтобы отменить подавление проблемы. Этот флажок активен, только если подавлена хотя бы одна из выбранных проблем. Доступность этой опции зависит от настроек роли пользователя «Подавлять проблемы».
Подтвердить	Отметьте 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).
Отменить подтверждение	Отметьте флажок, чтобы отменить подтверждение проблемы. Этот флажок доступен, если среди выбранных есть хотя бы одна подтвержденная проблема.
Преобразовать в причину	Отметьте флажок, чтобы преобразовать симптом(ы) проблемы в причину(ы).
Закрыть проблему	Отметьте флажок, чтобы вручную закрыть выбранные проблемы. Флажок для закрытия проблемы доступен, если в конфигурации триггера отмечена опция *Разрешить ручное закрытие* хотя бы для одной из выбранных проблем. Будут закрыты только те проблемы, которые разрешено закрывать при нажатии *Обновить*. Если ни одна проблема не может быть закрыта вручную, флажок будет отключен. Уже закрытые проблемы не будут закрываться повторно.

Отображение На основе информации о подтверждении можно настроить отображение количества проблем на панели или картах. Для этого вам необходимо сделать выбор в опции *Отображение проблемы*, доступной как в настройках карты, так и в *Проблемы по степени важности* панель управления виджетами. Можно отображать все количество проблем, а также количество неподтвержденных проблем отдельно от общего количества или количества неподтвержденных проблем.

На основе информации об обновлении проблемы (подтверждение и т. д.) можно настроить операции обновления - отправить сообщение или выполнить удаленные команды.

1 Подавление проблем

Обзор

Подавление проблем предлагает способ временно скрыть проблему, с которой можно разобраться позже. Это полезно для очистки списка проблем, чтобы придать наивысший приоритет наиболее срочным проблемам. Например, иногда на выходных может возникнуть проблема, которая недостаточно срочная, чтобы ее решать незамедлительно, поэтому ее можно «отложить» до утра понедельника.

Подавление проблем позволяет скрыть *единственную* проблему, в отличие от подавления проблем посредством обслуживания узла сети, когда все проблемы обслуживаемого узла сети скрыты.

Операции по триггерным действиям будут приостановлены для подавленных проблем так же, как это делается с помощью обслуживания узла сети.

Настройка

Проблему можно подавить через окно **обновление проблемы**, где подавление является одним из вариантов обновления проблемы наряду с комментированием, изменением серьезности, подтверждением и т. д.

Проблема также может быть восстановлена через то же окно обновления проблемы.

Отображение

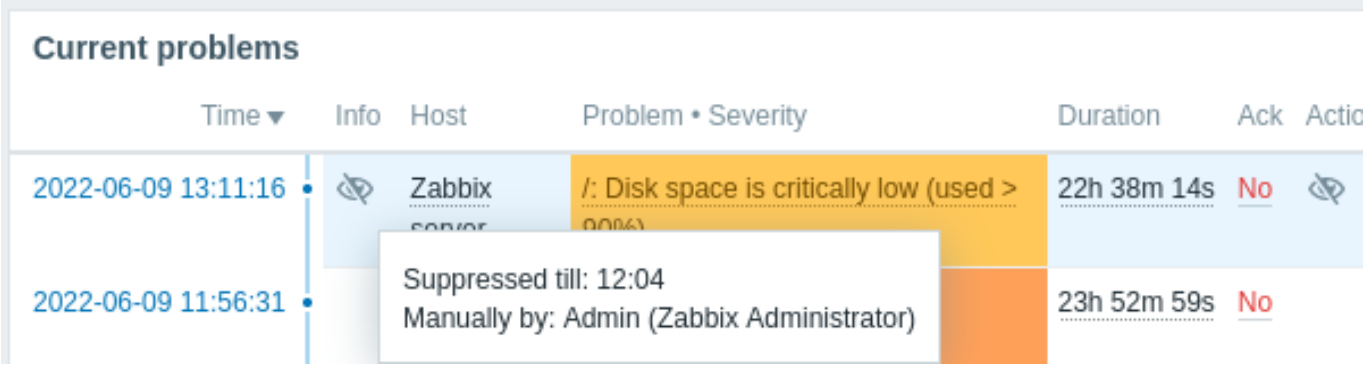
После подавления проблема помечается мигающим значком подавления в столбце *Информация*, прежде чем быть скрытой. .

Значок подавления мигает, пока задача подавления находится в списке ожидания. Как только 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 не работает и диспетчер задач не может выполнить задачу. Та же логика применима и к отказу от подавления. В течение короткого периода времени после отправки задачи и ее невыполнения сервером значок отмены подавления мигает.

Подавленная проблема может быть как скрытой, так и отображенной, в зависимости от настроек фильтра/виджета проблем.

Подавленная проблема может быть как скрытой, так и отображенной, в зависимости от настроек фильтра/виджета проблем.

При отображении в списке проблем подавленная проблема отмечается значком подавления, а подробности подавления отображаются при наведении указателя мыши:



Подробности о подавлении также отображаются во всплывающем окне при наведении указателя мыши на значок подавления в столбце *Действия*.

12 Экспорт/импорт конфигурации

Обзор Функциональность экспорта/импорта в Zabbix позволяет обмениваться различными сущностями конфигурации между одной системой Zabbix и другой.

Типичные варианты использования этой функциональности:

- совместное использование шаблонов или карт сети — пользователи Zabbix могут делиться своими параметрами конфигурации
- загрузка шаблона в Zabbix Community templates. Затем другие пользователи смогут скачать шаблон и импортировать файл в Zabbix.
- интеграция со сторонними инструментами — универсальные форматы YAML, XML и JSON позволяют выполнять интеграцию, а также импорт/экспорт данных с помощью сторонних инструментов и приложений

Что можно импортировать / экспортировать

Объекты, которые можно импортировать / экспортировать:

- группы хостов (*только через Zabbix API*)
- шаблоны
- хосты
- карты сети
- способы оповещения
- изображения

Формат экспорта

Данные можно экспортировать при помощи 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), которые наследуются от связанных шаблонов, не экспортируются. Любые изменения, внесённые в эти сущности на уровне узла сети (например, изменённый интервал элемента данных, изменённое регулярное выражение или добавленные прототипы в правило low-level discovery), будут потеряны при экспорте; при импорте все сущности из связанных шаблонов будут созданы заново в соответствии с исходным связанным шаблоном.
- Сущности, созданные low-level discovery, и любые сущности, зависящие от них, не экспортируются. Например, триггер, созданный для элемента данных, сгенерированного правилом LLD, не будет экспортирован.
- Если экспортируемый узел сети/шаблон содержит сущности, поддерживающие тайм-ауты, значения тайм-аутов будут экспортированы, если для этих сущностей настроены собственные тайм-ауты.

Дополнительная информация об импорте

- Импорт останавливается при первой ошибке.
- При обновлении существующих изображений при импорте изображений поле «тип изображения» игнорируется, т.е. изменить тип изображения посредством импорта невозможно.
- При импорте хостов/шаблонов с использованием опции «Удалить отсутствующие» макросы хоста/шаблона, отсутствующие в файле импорта, будут удалены из хоста/шаблона после импорта.
- Пустые теги для элементов, триггеров, графиков, discoveryRules, itemPrototypes, ignorePrototypes, graphPrototypes не имеют смысла, т.е. это то же самое, как если бы они отсутствовали.
- Если для объектов импортированного хоста/шаблона настроены собственные таймауты, они будут применены; в противном случае будут применены прокси/глобальные таймауты.
- Импорт поддерживает YAML, XML и JSON, файл импорта должен иметь правильное расширение файла: .yaml и .yml для YAML, .xml для XML и .json для JSON. См. информацию о совместимости о поддерживаемых версиях XML.
- Импорт поддерживает файлы конфигурации только в кодировке UTF-8 (с [BOM] или без него (https://en.wikipedia.org/wiki/Byte_or 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.

Формат XML Формат экспорта XML содержит следующие теги:

- Заголовок по умолчанию 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 в JSON
- Версия экспорта

```
{
    "zabbix_export": {
        "version": "8.0"
    }
}
```

Другие объекты зависят от экспортируемых объектов.

1 Группы шаблонов

Обзор

В веб-интерфейсе группы шаблонов можно экспортировать только вместе с экспортом шаблонов. 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.

2 Группы узлов сети

Обзор

В веб-интерфейсе группы узлов сети можно экспортировать только вместе с экспортом узлов 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'
```

Экспортируемые элементы

Элемент	Тип	Описание
uuid	string	Уникальный идентификатор этой группы узлов сети.
name	string	Имя группы.

3 Шаблоны

Обзор

Шаблоны экспортируются вместе со множеством связанных объектов и со связями с объектами.

Экспорт шаблона содержит:

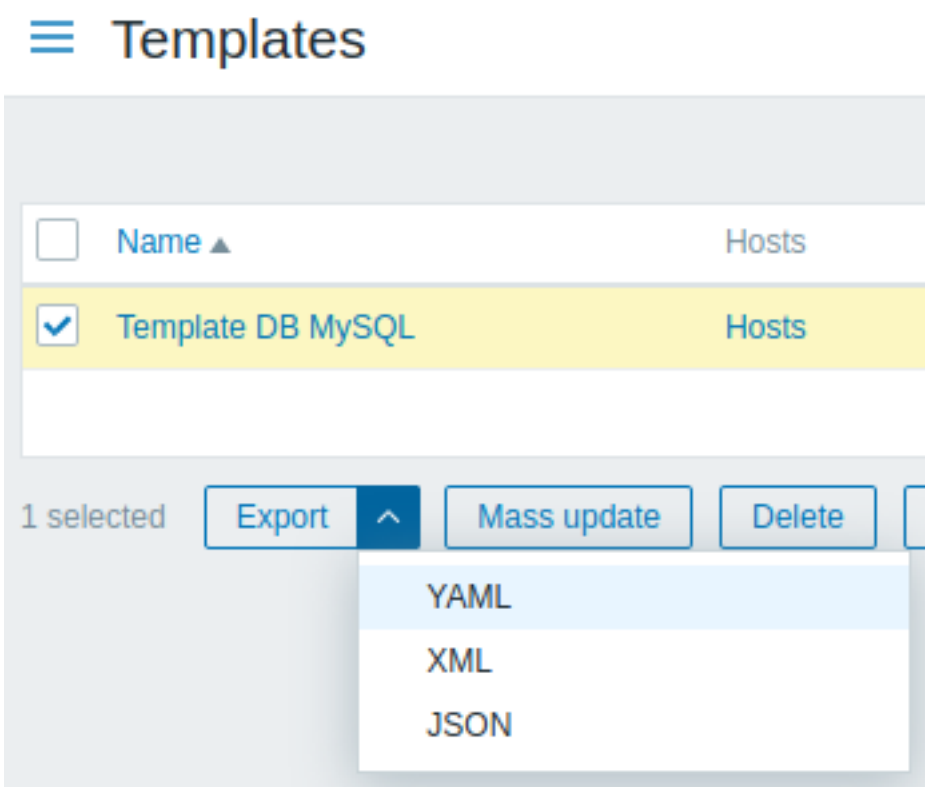
- присоединенные группы хостов
- данные шаблона
- связи с другими шаблонами
- связи с группами хостов
- напрямую присоединенные элементы данных
- напрямую присоединенные триггеры
- напрямую присоединенные графики
- напрямую присоединенные панели

- напрямую присоединенные правила обнаружения со всеми прототипами
- напрямую присоединенные веб-сценарии
- преобразования значений

Экспорт

Чтобы экспортировать шаблоны, выполните следующие действия:

1. Перейдите в *Сбор данных* → *Шаблоны*.
2. Установите флажки у шаблонов, которые нужно экспортировать.
3. Нажмите *Экспорт* под списком.



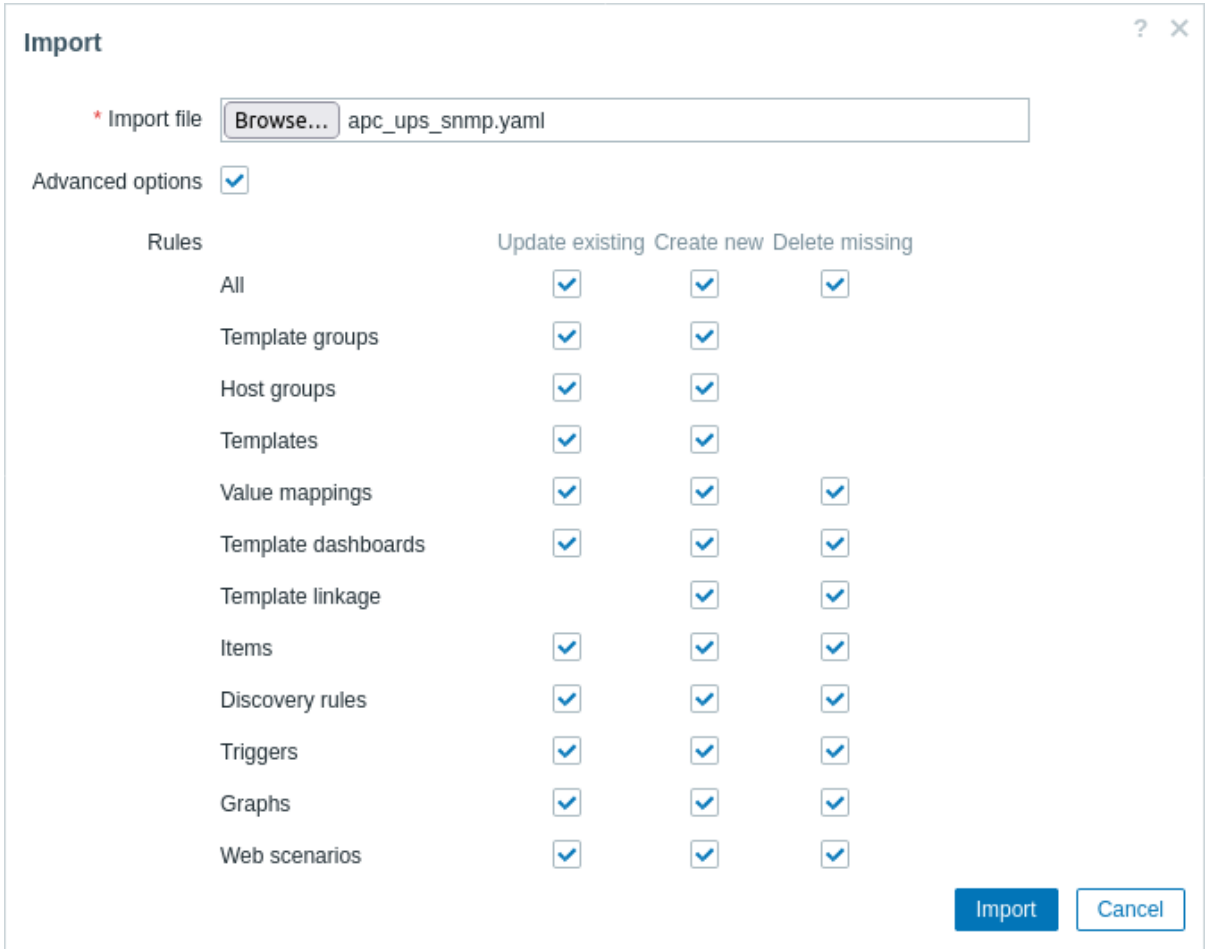
В зависимости от выбранного формата шаблоны экспортируются в локальный файл с именем по умолчанию:

- `zabbix_export_templates.yaml` — при экспорте в YAML (вариант по умолчанию для экспорта);
- `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 *Data collection* → *Templates*.
2. Нажмите *Import* в правом верхнем углу.
3. Выберите файл импорта.
4. Нажмите *Import* в правом нижнем углу формы настройки.



Если установить флажок *Advanced options*, будет отображен подробный список всех элементов, доступных для импорта, — отметьте или снимите отметку с каждого правила импорта по мере необходимости.

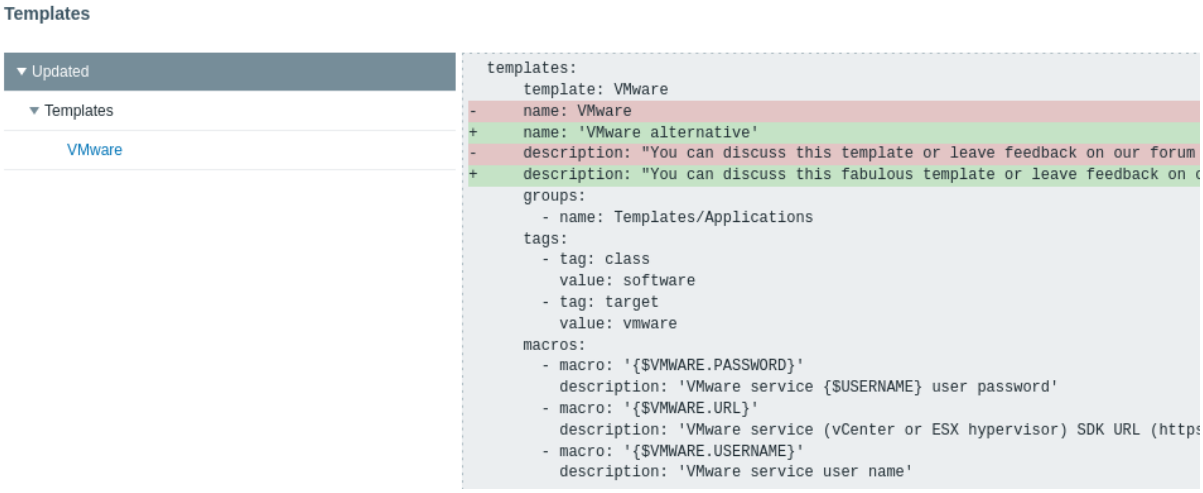
Если установить или снять флажок в строке *All*, будут отмечены/сняты все элементы ниже.

Правила импорта:

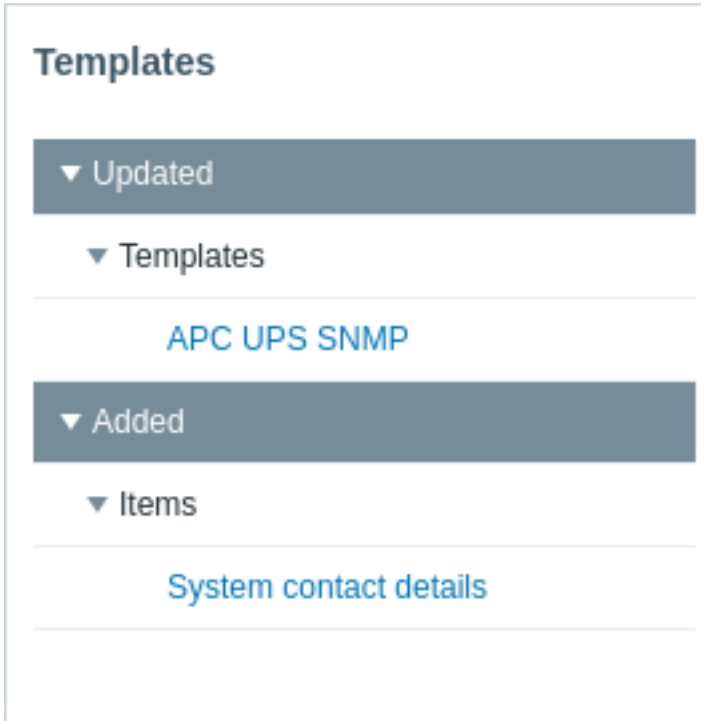
Rule	Description
Update existing	Существующие элементы будут обновлены с использованием данных из файла импорта. В противном случае они не будут обновлены.
Create new	Новые элементы будут созданы с использованием данных из файла импорта. В противном случае они не будут созданы.
Delete missing	Существующие элементы, отсутствующие в файле импорта, будут удалены. В противном случае они не будут удалены. Если для *Template linkage* отмечен параметр *Delete missing*, текущая привязка шаблонов, отсутствующая в файле импорта, будет отвязана. Сущности (элементы данных, триггеры, графики и т. д.), унаследованные от отвязанных шаблонов, не будут удалены (если только параметр *Delete missing* не выбран также для каждой сущности).

Attention:
Если шаблоны с такими же именами уже существуют, при импорте следует установить параметры *Delete missing*, 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 вы сможете просмотреть содержимое импортируемого шаблона. Если это новый шаблон, все элементы будут показаны зеленым цветом. При обновлении существующего шаблона новые элементы шаблона будут выделены зеленым цветом; удаленные элементы шаблона будут выделены красным цветом; элементы, которые не изменились, будут показаны на сером фоне.



Меню слева можно использовать для навигации по списку изменений. Раздел *Updated* выделяет все изменения, внесенные в существующие элементы шаблона. В разделе *Added* перечислены новые элементы шаблона. Элементы в каждом разделе сгруппированы по типу элемента; нажмите серую стрелку, чтобы развернуть или свернуть группу элементов.



Просмотрите изменения шаблона, затем нажмите *Import*, чтобы выполнить импорт шаблона. Сообщение об успешном или неуспешном импорте будет отображено в веб-интерфейсе.

Формат экспорта При экспорте шаблона Zabbix генерирует структурированный формат YAML, JSON или XML. Экспорт включает элементы шаблона, такие как метаданные шаблона, элементы данных, макросы, триггеры, панели и другое.

Каждый элемент служит определённой цели и может содержать вложенные элементы.

В следующих разделах описывается каждый элемент в формате экспорта. В примерах используется шаблон Linux by Zabbix agent.

Многоточие (`...`) указывает на элементы, опущенные для краткости. Примечание (`see table below`) используется, когда элемент более подробно объясняется в последующих разделах.

```
zabbix_export:
  version: '8.0'
  template_groups:
    - uuid: 846977d1dfed4968bc5f8bdb363285bc
      name: 'Templates/Operating systems'
  templates: ( .        )
```

```
  triggers: ( .        )
  graphs: ( .        )
```

Метаданные шаблона

Элемент	Тип	Описание
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, используемых 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.

```
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: ( .        )
      discovery_rules: ( .        )
      tags:
        - tag: class
          value: os
        - tag: target
          value: linux
      macros:
        - macro: '{$AGENT.TIMEOUT}'
          value: 3m
          description: '  -  ,                              .                  ,             /    Zabbix (
          config: ( .        )
        - macro: '{$CPU.UTIL.CRIT}'
          value: '90'
          description: '                 CPU,         %.'
          config: ( .        )
        (...)
      dashboards: ( .        )
      valuemaps: ( .        )
    (...)
```

Шаблоны

Элемент	Type	Описание
uuid	string	(обязательно) Уникальный идентификатор этого шаблона.
template	string	(обязательно) Уникальное имя шаблона.
name	string	Видимое имя шаблона.
description	text	Описание шаблона.
wizard_ready	text	Доступен ли шаблон для выбора в мастере узла сети. Возможные значения:[1] NO (0, по умолчанию), YES (1). См. также: объект шаблона (`wizard_ready`).
readme	text	Инструкции по настройке, специфичные для шаблона, которые отображаются в мастере узла сети. Поддерживается форматирование Markdown.
vendor		Корневой элемент для поставщика шаблона (присутствует, если экспортированный шаблон содержит данные о поставщике).
name	string	(обязательно) Имя поставщика шаблона.
version	string	(обязательно) Версия шаблона. Для готовых шаблонов версия отображается следующим образом: основная версия Zabbix, разделитель (”-”), номер редакции (увеличивается с каждой новой версией шаблона и сбрасывается с каждой новой основной версией Zabbix). Например, 7.0-0, 7.0-3, 8.0-0, 8.0-3.
templates		Корневой элемент для связанных шаблонов.
name	string	(обязательно) Имя шаблона.
groups		Корневой элемент для групп шаблонов.
name	string	(обязательно) 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	(обязательно) Имя тега.
value	string	Значение тега.
macros		Корневой элемент для пользовательских макросов шаблона.
macro	string	(обязательно) Имя пользовательского макроса.
type	string	Тип пользовательского макроса. Возможные значения:[1] TEXT (0, по умолчанию), SECRET_TEXT (1), VAULT (2). См. также: объект пользовательского макроса (`type`).
value	string	Значение пользовательского макроса.
description	string	Описание пользовательского макроса.
config		Корневой элемент для конфигурации макроса шаблона, отвечающей за то, как макрос отображается в мастере узла сети.
dashboards		Корневой элемент для панелей шаблона.
valuemaps		Корневой элемент для карт соответствия 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	(обязательный) Уникальный идентификатор этого элемента данных.
name	string	(обязательный) 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).
snmp_oid	string	(обязательный для элементов данных SNMP_AGENT) ID объекта SNMP.
key	string	(обязательный) Ключ элемента данных.
delay	string	Интервал обновления элемента данных. По умолчанию: 1m. Для элементов данных TRAP значение всегда будет 0.
history	string	Период времени (с использованием суффикса времени, пользовательского макроса или LLD-макроса), в течение которого должны храниться исторические данные. По умолчанию: 31d.
trends	string	Период времени (с использованием суффикса времени, пользовательского макроса или LLD-макроса), 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 получаемого 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, которым разрешено отправлять данные для элемента данных, разделённых запятыми. Поддерживается для элементов данных TRAP и HTTP_AGENT.
units	string	Единицы измерения получаемого значения (bps, B и т. д.).
params	text	Дополнительные параметры в зависимости от типа элемента данных (выполняемый 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	Сенсор 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).

Элемент	Тип	Описание
username	string	(обязательный для элементов данных SSH и TELNET) Имя пользователя для аутентификации. Поддерживается для SIMPLE, ODBC, JMX и HTTP_AGENT элементов данных. При использовании для элементов данных JMX также должен быть указан `password` (см. ниже), либо оба элемента должны быть оставлены пустыми.
password	string	(обязательный для элементов данных SSH и TELNET) Пароль для аутентификации. Поддерживается для SIMPLE, ODBC, JMX и HTTP_AGENT элементов данных. При использовании для элементов данных JMX также должен быть указан `username` (см. выше), либо оба элемента должны быть оставлены пустыми.
publickey	string	(обязательный для элементов данных SSH) Имя файла открытого ключа.
privatekey	string	(обязательный для элементов данных SSH) Имя файла закрытого ключа.
description	text	Описание элемента данных.
inventory_link	string	Поле инвентаризации узла сети, которое заполняется этим элементом данных. Возможные значения:[1] NONE (0), ALIAS (4) и т. д. (поддерживаемые поля см. в разделе Инвентаризация узла сети).
valuemap		Корневой элемент для карт соответствия значений элемента данных.
name	string	(обязательный) Имя карты соответствия значений, используемой для элемента данных.
logtimefmt	string	Формат времени в записях журнала. Поддерживается для элементов данных со значением типа LOG.
preprocessing		Корневой элемент для предобработки значений элемента данных.
step		Корневой элемент для шагов предобработки значений элемента данных шаблона.
jmx_endpoint	string	Конечная точка JMX. Поддерживается для элементов данных JMX.
master_item		(обязательный для элементов данных DEPENDENT) Корневой элемент для мастер-элемента данных зависимого элемента данных.
key	string	(обязательный) Ключ мастер-элемента данных зависимого элемента данных.
timeout	string	Тайм-аут запроса опроса данных элемента данных. Поддерживается для списка типов элементов данных Timeouts.
url	string	(обязательный для элементов данных HTTP_AGENT) Строка URL.
query_fields		Корневой элемент для параметров запроса. Поддерживается для элементов данных HTTP_AGENT.
name	string	(обязательный 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	(обязательный 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 HTTP(S)-запроса. 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 опросе данных. Поддерживается для элементов данных HTTP_AGENT. Возможные значения:[1] NO (0), YES (1, по умолчанию).
post_type	string	Тип тела отправляемых данных. Поддерживается для элементов данных HTTP_AGENT. Возможные значения:[1] RAW (0, по умолчанию), JSON (2), XML (3).
http_proxy	string	Строка подключения к HTTP(S) прокси. Поддерживается для элементов данных HTTP_AGENT.
headers		Корневой элемент для заголовков HTTP(S)-запроса. Поддерживается для элементов данных HTTP_AGENT.
name	string	(обязательный для элементов данных HTTP_AGENT) Имя заголовка.
value	string	(обязательный для элементов данных HTTP_AGENT) Значение заголовка.
retrieve_mode	string	Какая часть ответа должна быть сохранена. Поддерживается для элементов данных HTTP_AGENT. Возможные значения:[1] BODY (0, по умолчанию), HEADERS (1), BOTH (2).

Элемент	Тип	Описание
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	Как обрабатывать ответ. Поддерживается для элементов данных HTTP_AGENT. Возможные значения:[1] RAW (0, по умолчанию), JSON (1).
allow_traps	string	Разрешить заполнять значение 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	Следует ли проверять подлинность сертификата узла сети. Поддерживается для элементов данных HTTP_AGENT. Возможные значения:[1] NO (0, по умолчанию), YES (1).
verify_host	string	Следует ли проверять, что имя узла сети для соединения совпадает с именем в сертификате узла сети. Поддерживается для элементов данных HTTP_AGENT. Возможные значения:[1] NO (0, по умолчанию), YES (1).
tags		Корневой элемент для тегов элемента данных.
tag	string	(обязательный) Имя тега.
value	string	Значение тега.
triggers		Корневой элемент для триггеров элемента данных шаблона.

Note:
См. также: Объект элемента данных (обратитесь к соответствующему свойству с совпадающим именем).

Шаги предобработки значения элемента данных шаблона

```
zabbix_export:
  (...)
  templates:
    (...)
      items:
        (...)
        - uuid: 58818005e76d46dda14d6592f601ab00
          (...)
          preprocessing:
            - type: JSONPATH
              parameters:
                - $.length()
            - type: DISCARD_UNCHANGED_HEARTBEAT
              parameters:
                - 12h
          (...)
```

Элемент	Тип	Описание
type	string	(обязательно) Тип шага предобработки 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 как: полученное значение - предыдущее значение), CHANGE_PER_SECOND (10, вычисляется как: (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 предобработки значения элемента данных.
parameter	string	Отдельный параметр шага предобработки значения элемента данных.
error_handler	string	Тип действия, используемого в случае сбоя шага предобработки. 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 предобработки элемента данных (см. соответствующее свойство с совпадающим именем).

Триггеры элементов 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
        (...)
```

Элемент	Type	Описание
uuid	string	(обязательно) Уникальный идентификатор этого триггера.
expression	string	(обязательно) Выражение триггера.
recovery_mode	string	Основа для генерации OK-событий. Возможные значения:[1] EXPRESSION (0, по умолчанию), 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, по умолчанию), TAG_VALUE (1).
correlation_tag	string	Имя тега, которое будет использоваться для корреляции событий.
name	string	(обязательно) Имя триггера.
event_name	string	Имя события.
opdata	string	Операционные данные.
url_name	string	Метка для URL, связанного с триггером.
url	string	URL, связанный с триггером.
status	string	Состояние триггера. Возможные значения:[1] ENABLED (0, по умолчанию), DISABLED (1).
priority	string	Важность триггера. Возможные значения:[1] NOT_CLASSIFIED (0, по умолчанию), INFO (1), WARNING (2), AVERAGE (3), HIGH (4), DISASTER (5).
description	text	Описание триггера.
type	string	Тип генерации событий (одно событие о проблеме или несколько событий о проблеме). Возможные значения:[1] SINGLE (0, по умолчанию), MULTIPLE (1).
manual_close	string	Ручное закрытие событий о проблеме. Возможные значения:[1] NO (0, по умолчанию), YES (1).
dependencies		Корневой элемент для зависимостей.
name	string	(обязательно) Имя триггера зависимости.
expression	string	(обязательно) Выражение триггера зависимости.
recovery_expression	string	Выражение восстановления триггера зависимости.
tags		Корневой элемент для тегов триггера.
tag	string	(обязательно) Имя тега.
value	string	Значение тега.

Note:
См. также: Trigger object (см. соответствующее свойство с совпадающим именем).

Правила низкоуровневого обнаружения шаблона

```
zabbix_export:
  (...)
  templates:
    (...)
      discovery_rules:
        - uuid: acfdea9c46ef48c68e6636f43b8f96a2
          name: '                    '
          key: net.if.discovery
          delay: 1h
          filter: ( .         )
          description: '                    .'
          item_prototypes: ( .         )
          trigger_prototypes: ( .          )
          graph_prototypes: ( .         )
        (...)
```

Attention:
Большинство элементов правил низкоуровневого обнаружения шаблона совпадают с элементами элементов данных шаблона. В таблице ниже описаны те элементы, которые отличаются от элементов данных шаблона.

Element	Type	Description
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), JMX (16), DEPENDENT (18), HTTP_AGENT (19), SNMP_AGENT (20), ITEM_TYPE_SCRIPT (21), ITEM_TYPE_BROWSER (22).
key	string	(обязательно) Ключ правила низкоуровневого обнаружения.
filter		Корневой элемент для фильтров правил низкоуровневого обнаружения шаблона.
lifetime	string	Период времени (с использованием секунд, суффикса времени или пользовательского макроса), по истечении которого 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 (с использованием секунд, суффикса времени или пользовательского макроса), по истечении которого 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 элементов данных шаблона, которые совпадают с элементами данных шаблона.
trigger_prototypes		Корневой элемент для элементов прототипов триггеров шаблона, которые совпадают с триггерами элементов данных шаблона.
graph_prototypes		Корневой элемент для элементов прототипов графиков шаблона, которые совпадают с графиками узла сети.
host_prototypes		Корневой элемент для элементов 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, являющийся прямым дочерним элементом указанного правила/прототипа правила.
key	string	(обязательно) Ключ родительского правила низкоуровневого обнаружения (или прототипа правила).
master_item	string	(обязательно для правил DEPENDENT) Корневой элемент для master-элемента данных зависимого правила.
lld_macro_paths		Корневой элемент для путей макросов правила низкоуровневого обнаружения.
lld_macro	string	(обязательно) Имя макроса правила низкоуровневого обнаружения.
path	string	(обязательно) Селектор значения, которое будет присвоено соответствующему макросу.
preprocessing		Корневой элемент для предобработки значений правила низкоуровневого обнаружения.
step		Корневой элемент для элементов шагов предобработки значений правила низкоуровневого обнаружения, которые совпадают с шагами предобработки значений элемента данных шаблона, за исключением меньшего количества возможных значений. См. также: объект предобработки правила LLD.
overrides		Корневой элемент для правил переопределения правила низкоуровневого обнаружения.
name	string	(обязательно) Уникальное имя переопределения.
step	string	(обязательно) Уникальный порядковый номер переопределения.
stop	string	Остановить обработку следующих переопределений при совпадении.
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 (обратитесь к соответствующему свойству с совпадающим именем).

Фильтры правил 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
          (...)
```

Элемент	Type	Описание
evaltype	string	Переопределить метод вычисления 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	(обязательно) Имя макроса низкоуровневого обнаружения, для которого выполняется проверка.
value	string	Значение для сравнения.
operator	string	Оператор условия. Возможные значения:[1] MATCHES_REGEX (8, по умолчанию), NOT_MATCHES_REGEX (9).
formulaid	string	(обязательно) Произвольный уникальный ID, используемый для ссылки на условие из пользовательского выражения. Может содержать только заглавные буквы. ID должен быть определён пользователем при изменении условий фильтра, но будет сгенерирован заново при их последующем запросе.

Note:
См. также: объект фильтра правила LLD (обратитесь к соответствующему свойству с совпадающим именем).

Шаблон низкоуровневого правила обнаружения переопределяет операции

Элемент	Тип	Описание
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 при операция переопределения.

Элемент	Тип	Описание
discover	string	Добавлен ли объект в результате обнаружения.
delay	string	Интервал обновления, установленный для прототипа элемента при операции переопределения.
history	string	Период хранения истории, установленный для прототипа элемента при операции переопределения.
trends	string	Период хранения тренда, установленный для прототипа элемента при операции переопределения.
severity	string	Серьезность прототипа триггера, установленная при операции переопределения.
tags		Корневой элемент для тегов, установленных для объекта при операции переопределения.
tag	string	(обязательно) Имя тега.
value	string	Значение тега.
templates		Корневой элемент для шаблонов, связанных с прототипом хоста при операции переопределения.
name	string	(обязательно) Имя шаблона.
inventory_mode	string	Режим инвентаризации прототипа хоста, установленный при операции переопределения.

::: noteclassicСм. также: объект операции переопределения правила LLD (ссылка на соответствующее свойство с соответствующим именем).:::

Веб-сценарии шаблона

Элемент	Тип	Описание
uuid	string	(обязательно) Уникальный идентификатор этого веб-сценария.
name	string	(обязательно) Имя веб-сценария.
delay	string	Частота выполнения веб-сценария (в секундах, с использованием суффикса времени или пользовательского макроса). По умолчанию: 1m.
attempts	integer	Количество попыток выполнения шагов веб-сценария. Возможные значения:[1] 1-10 (по умолчанию: 1).
agent	string	Клиентский агент. Zabbix будет представляться выбранным браузером. Это полезно, если веб-сайт возвращает разное содержимое для разных браузеров. По умолчанию: Zabbix.
http_proxy	string	Прокси, который будет использоваться веб-сценарием, задаётся в виде: `http://[username[:password]@]proxy.example.com[:port]`
variables		Корневой элемент для переменных веб-сценария, которые могут использоваться в шагах сценария.
name	string	(обязательно) Имя переменной.
value	text	(обязательно) Значение переменной.
headers		Корневой элемент для HTTP-заголовков, отправляемых при выполнении запроса. Заголовки должны быть перечислены с использованием того же синтаксиса, в котором они отображаются в протоколе HTTP.
name	string	(обязательно) Имя заголовка.
value	text	(обязательно) Значение заголовка.
status	string	Состояние 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 соответствие поля *Common Name* или поля *Subject Alternate Name* сертификата веб-сервера. Возможные значения:[1] NO (0, по умолчанию), YES (1).

Элемент	Тип	Описание
ssl_cert_file	string	Имя файла SSL-сертификата, используемого для аутентификации клиента (должен быть в формате PEM).
ssl_key_file	string	Имя файла закрытого SSL-ключа, используемого для аутентификации клиента (должен быть в формате PEM).
ssl_key_password	string	Пароль файла закрытого SSL-ключа.
steps		(обязательно) Корневой элемент для шагов веб-сценария шаблона.
tags		Корневой элемент для тегов веб-сценария.
tag	string	(обязательно) Имя тега.
value	string	Значение тега.

Note:
См. также: Объект веб-сценария (обратитесь к соответствующему свойству с совпадающим именем).

Шаблон веб-сценария шаги

Элемент	Тип	Описание
name	строка	(обязательно) Имя шага веб-сценария.
url	строка	(обязательно) URL для мониторинга.

::: noteclassicСм. также: Объект шага веб-сценария (ссылка на соответствующее свойство с соответствующим именем).:::

Настройка макросов 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:
            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	Type	Описание
type	string	(обязательный) 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	(обязательный для макросов TEXT, LIST, CHECKBOX) Метка макроса.
description	text	Текст справки для макроса. Поддерживается форматирование Markdown.
required	string	Отмечает 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 текстовом поле. Поддерживается для макросов TEXT.
options		Корневой элемент для элементов LIST или значений CHECKBOX.
value	string	(обязательный для макросов LIST) Значение элемента LIST.
text	string	(обязательный для макросов LIST) Текст элемента LIST.
checked	string	(обязательный для макросов CHECKBOX) Значение, представляющее отмеченное состояние. Возможные значения:[1] 0 (false), 1 (true). Поддерживается для макросов CHECKBOX.
unchecked	string	(обязательный для макросов CHECKBOX) Значение, представляющее неотмеченное состояние. Возможные значения:[1] 0 (false), 1 (true). Поддерживается для макросов CHECKBOX.

Note:
См. также: Macro configuration object (см. соответствующее свойство с совпадающим именем).

Дашборды шаблона

```
zabbix_export:
  (...)
  templates:
    (...)
      dashboards:
        - uuid: c689ad3115fd46a4b927d1f70ee2e5a4
          name: Filesystems
          pages:
            - name: Overview
              widgets: (see table below)
        (...)
```

Элемент	Тип	Описание
uuid	string	(обязательно) Уникальный идентификатор этого дашборда.
name	string	(обязательно) Имя дашборда шаблона.
display_pe-riod	integer	Период отображения страниц дашборда.
auto_start	string	Автоматический запуск слайд-шоу. Возможные значения:[1] NO (0), YES (1, по умолчанию).
pages		Корневой элемент для страниц дашборда шаблона.
name	string	Имя страницы.
display_period	integer	Период отображения страницы.
sortorder	integer	Порядок сортировки страницы.
widgets		Корневой элемент для виджетов дашборда шаблона.

Note:
См. также: Объект дашборда шаблона (см. соответствующее свойство с таким же именем).

Виджеты панели шаблона

```
zabbix_export:
  (...)
  templates:
    (...)
      dashboards:
        - uuid: c689ad3115fd46a4b927d1f70ee2e5a4
          name:
          pages:
            - name:
              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})]:                    (              )'
                    - 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})]:                  ,  % (                )'
                    - type: STRING
                      name: reference
                      value: AAAEC
        (...)
```

Элемент	Тип	Описание
type	string	(обязательный) Тип виджета.
name	string	Имя виджета.
x	integer	Положение по горизонтали от левой стороны панели шаблона. Возможные значения:[1] 0-71.
y	integer	Положение по вертикали 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	(обязательный) 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).
name	string	(обязательный) Имя поля виджета.

Элемент	Тип	Описание
value	mixed	(обязательный) Значение поля виджета, в зависимости от типа поля.

Note:
См. также: Объект виджета панели шаблона (обратитесь к соответствующему свойству с совпадающим именем).

Карты соответствия значений

```
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	(обязательный) Уникальный идентификатор для этой карты соответствия значений.
name	string	(обязательный) Имя карты соответствия значений.
mapping		Корневой элемент для сопоставлений.
type	string	Тип совпадения сопоставления. 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	(обязательный) Значение, в которое преобразуется исходное значение.

Note:
См. также: Объект карты соответствия значений (обратитесь к соответствующему свойству с совпадающим именем).

Сноски [1] Значения API типа integer в скобках, например ENABLED (0), упоминаются только для справки. Для получения дополнительной информации смотрите страницу связанного объекта API в записи таблицы или в конце каждого раздела.

4 Узлы сети

Обзор

Узлы сети экспортируются вместе со многими связанными объектами и связями между объектами.

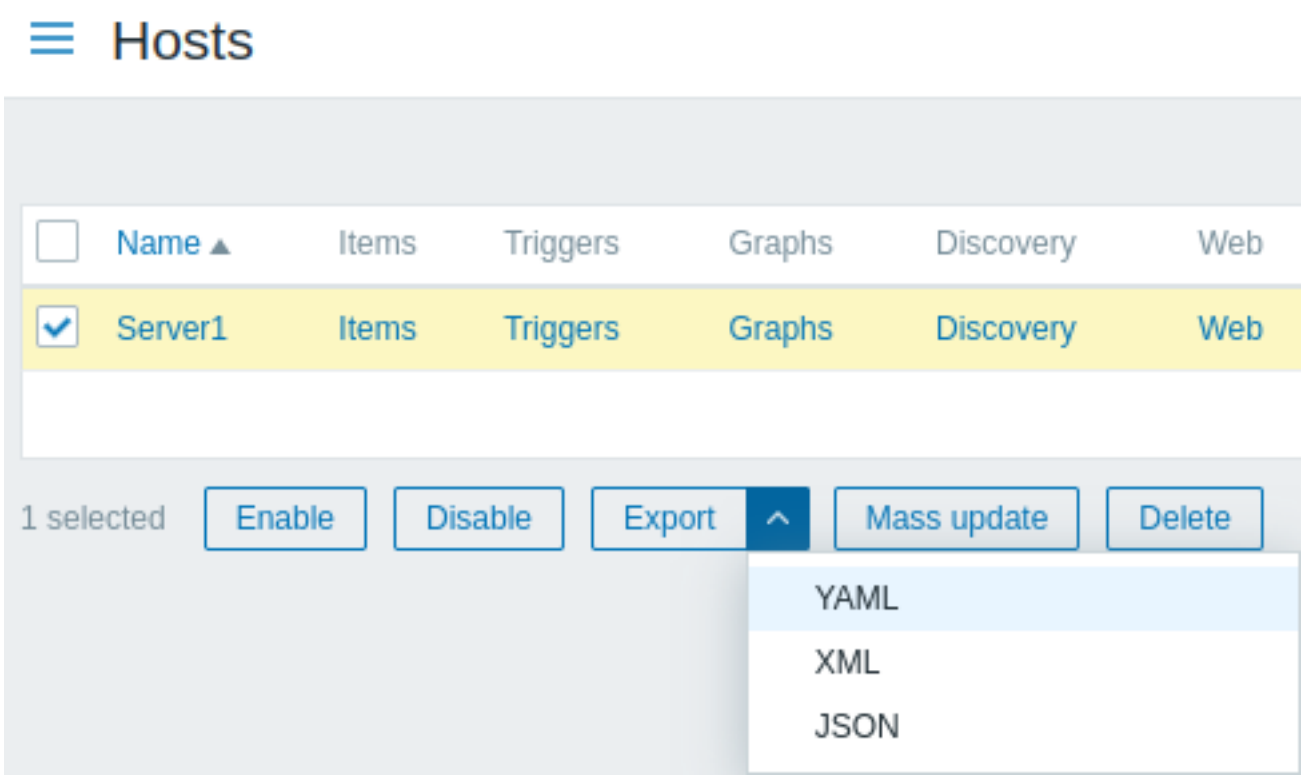
Экспорт узла сети содержит:

- Связанные группы узлов сети
- Данные узла сети
- Связи с шаблонами
- Связи с группами узлов сети
- Интерфейсы узла сети
- Напрямую связанные элементы данных
- Напрямую связанные триггеры
- Напрямую связанные правила обнаружения со всеми прототипами
- Напрямую связанные веб-сценарии
- Макросы узла сети
- Данные инвентаризации узла сети
- Карты соответствия значений
- Связанные графики

Экспорт

Чтобы экспортировать узлы сети, выполните следующие действия:

1. Перейдите в *Сбор данных* → *Узлы сети*.
2. Установите флажки у узлов сети, которые нужно экспортировать.
3. Нажмите *Экспорт* под списком.



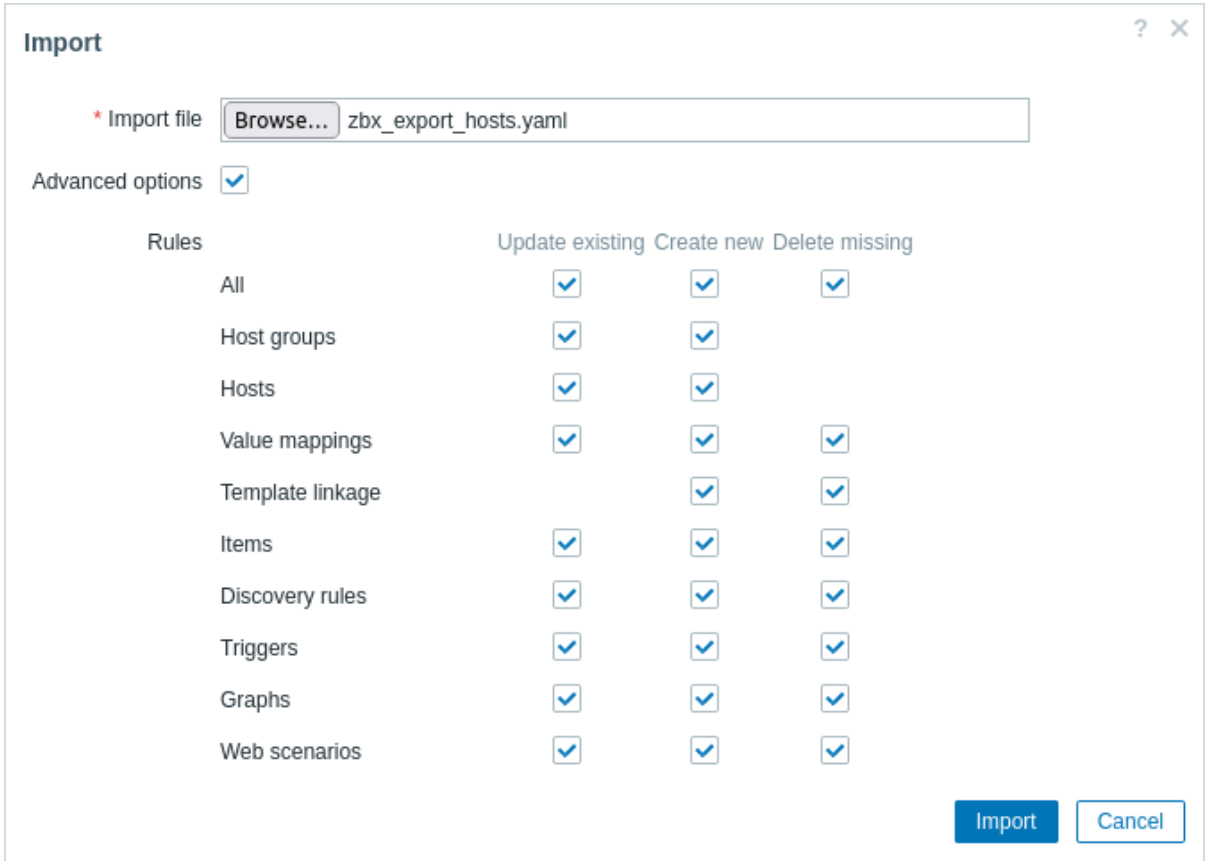
В зависимости от выбранного формата узлы сети экспортируются в локальный файл с именем по умолчанию:

- `zabbix_export_hosts.yaml` — при экспорте в YAML (вариант экспорта по умолчанию);
- `zabbix_export_hosts.xml` — при экспорте в XML;
- `zabbix_export_hosts.json` — при экспорте в JSON.

Импорт

Чтобы импортировать узлы сети, выполните следующие действия:

1. Перейдите в *Сбор данных* → *Узлы сети*.
2. Нажмите *Импорт* в правом верхнем углу.
3. Выберите файл импорта.
4. Нажмите *Импорт* в правом нижнем углу формы настройки.



Если вы установите флажок *Расширенные параметры*, будет отображён подробный список всех элементов, доступных для импорта, — установите или снимите отметку для каждого правила импорта по мере необходимости.

Если вы нажмёте флажок в строке *Все*, все элементы ниже будут отмечены/сняты.

Правила импорта:

Правило	Описание
Обновить существующие	Существующие элементы будут обновлены с использованием данных из файла импорта. В противном случае они не будут обновлены.
Создать новые	Новые элементы будут созданы с использованием данных из файла импорта. В противном случае они не будут созданы.
Удалить отсутствующие	Существующие элементы, отсутствующие в файле импорта, будут удалены. В противном случае они не будут удалены. Если для *Связь с шаблонами* отмечен параметр *Удалить отсутствующие*, текущая связь с шаблонами, отсутствующая в файле импорта, будет удалена. Сущности (элементы данных, триггеры, графики и т. д.), унаследованные от отключённых шаблонов, не будут удалены (если только параметр *Удалить отсутствующие* не выбран также для каждой сущности).

Сообщение об успешном или неуспешном импорте будет отображено в веб-интерфейсе.

Формат экспорта

При экспорте узла сети Zabbix генерирует структурированный формат YAML, JSON или XML. Экспорт включает элементы узла сети, такие как метаданные узла сети, элементы данных, макросы, триггеры и другие.

Каждый элемент служит определенной цели и может содержать вложенные элементы.

В следующих разделах описывается 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. Отвязка сохраняет все элементы в конфигурации (в отличие от отвязки с очисткой, которая удаляет их).

Многоточие `(...)` обозначает элементы, опущенные для краткости. Примечание `(see table below)` используется, когда элемент более подробно объясняется в следующих разделах.

```
zabbix_export:
  version: '8.0'
  host_groups:
    - uuid: 748ad4d098d447d492bb935c907f652f
      name: Databases
  hosts: ( .        )
  graphs: ( .        )
```

Метаданные узла сети

Элемент	Тип	Описание
version	string	(обязательно) Версия Zabbix.
host_groups		(обязательно) Корневой элемент для групп узлов сети.
uuid	string	(обязательно) Уникальный идентификатор этой группы узлов сети.
name	string	(обязательно) Имя группы узлов сети.
hosts		Корневой элемент для узлов сети.
graphs		Корневой элемент для графиков узла сети.

```
zabbix_export:
  (...)
  hosts:
    - host: 'MySQL server'
      name: 'MySQL server'
      groups:
        - name: Databases
      interfaces: ( .        )
      items: ( .        )
      discovery_rules: ( .        )
      tags:
        - tag: location
          value: Riga
      macros:
        (...)
        - macro: '{$MYSQL.DSN}'
          value: 192.0.2.0
        (...)
        - macro: '{$MYSQL.PASSWORD}'
          type: SECRET_TEXT
          description: '            MySQL.'
        (...)
      valuemaps: ( .        )
  (...)
```

Узлы сети

Element	Type	Описание
host	string	(обязательно) Уникальное имя узла сети.
name	string	Видимое имя узла сети.
description	text	Описание узла сети.
monitored_by	string	Как узел сети отслеживается. Возможные значения:[1] SERVER (0, по умолчанию), PROXY (1) или PROXY_GROUP (2).
proxy		Корневой элемент для прокси.
name	string	(обязательно) Имя прокси (если есть), который отслеживает узел сети.
proxy_group		Корневой элемент для группы прокси.
name	string	(обязательно) Имя группы прокси (если есть), которая используется для мониторинга узла сети.
status	string	Состояние узла сети. Возможные значения:[1] ENABLED (0, по умолчанию), DISABLED (1).

Element	Type	Описание
ipmi_authtype	string	Тип аутентификации сеанса IPMI. Возможные значения:[1] DEFAULT (-1, по умолчанию), NONE (0), MD2 (1), MD5 (2), STRAIGHT (4), OEM (5), RMCP_PLUS (6).
ipmi_privilege	string	Уровень привилегий сеанса IPMI. Возможные значения:[1] CALLBACK (1), USER (2, по умолчанию), OPERATOR (3), ADMIN (4), OEM (5).
ipmi_username	string	Имя пользователя для IPMI-проверок.
ipmi_password	string	Пароль для IPMI-проверок.
templates		Корневой элемент для связанных шаблонов.
name	string	(обязательно) Имя шаблона.
groups		Корневой элемент для групп узлов сети, к которым принадлежит узел сети.
name	string	(обязательно) 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	(обязательно) Имя тега.
value	string	Значение тега.
macros		Корневой элемент для макросов узла сети.
macro	string	(обязательно) Имя пользовательского макроса.
type	string	Тип пользовательского макроса. Возможные значения:[1] TEXT (0, по умолчанию), SECRET_TEXT (1), VAULT (2).
value	string	Значение пользовательского макроса.
description	string	Описание пользовательского макроса.
inventory		Корневой элемент для инвентарных данных узла сети.
<inventory_property>	string	Свойство инвентарных данных. Все свойства имеют соответствующий элемент (`type`, `name`, `os` и т. д.; пример см. в разделе Формат экспорта).
inventory_mode	string	Режим инвентаризации. Возможные значения:[1] DISABLED (-1), MANUAL (0, по умолчанию), AUTOMATIC (1).
valuemaps		Корневой элемент для карт соответствия значений узла сети.

Note:
См. также: Host object (обратитесь к соответствующему свойству с совпадающим именем).

Интерфейсы узла сети

```
zabbix_export:
  (...)
  hosts:
    - host: 'MySQL server'
      (...)
      interfaces:
        - ip: 192.0.2.0
          interface_ref: if1
      (...)
```

Элемент	Type	Описание
default	string	Является ли это основным интерфейсом узла сети. Обратите внимание, что на узле сети может быть только один основной интерфейс каждого типа. Возможные значения:[1] NO (0), YES (1, по умолчанию).
type	string	Тип интерфейса. Возможные значения:[1] ZABBIX (1, по умолчанию), 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, по умолчанию).
ip	string	(обязательно для IP-подключений) IP-адрес (IPv4 или IPv6).
dns	string	(обязательно для DNS-подключений) DNS-имя.
port	string	Номер порта.
details		Корневой элемент для сведений об интерфейсе.

Элемент	Type	Описание
version	string	Использовать эту версию SNMP. Возможные значения:[1] SNMPV1 (1), SNMP_V2C (2, по умолчанию), SNMP_V3 (3).
community	string	(обязательно для элементов данных SNMPv1 и SNMPv2) SNMP community.
max_repetitions	string	Максимальное значение повторений для собственных 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, по умолчанию), AUTHNOPRIV (1), AUTHPRIV (2).
authprotocol	string	Протокол аутентификации SNMPv3. Поддерживается для элементов данных SNMPv3. Возможные значения:[1] MD5 (0, по умолчанию), SHA1 (1), SHA224 (2), SHA256 (3), SHA384 (4), SHA512 (5).
authpassphrase	string	Парольная фраза аутентификации SNMPv3. Поддерживается для элементов данных SNMPv3.
privprotocol	string	Протокол приватности SNMPv3. Поддерживается для элементов данных SNMPv3. Возможные значения:[1] DES (0, по умолчанию), AES128 (1), AES192 (2), AES256 (3), AES192C (4), AES256C (5).
privpassphrase	string	Парольная фраза приватности SNMPv3. Поддерживается для элементов данных SNMPv3.
bulk	string	Использовать ли bulk-запросы для SNMP. Возможные значения:[1] NO (0), YES (1, по умолчанию).
interface_ref	string	Имя ссылки на интерфейс для использования в элементах данных (формат: `if<N>`).

Note:
См. также: Host interface object (обратитесь к соответствующему свойству с совпадающим именем).

Элементы данных узла сети

```
zabbix_export:
  (...)
  hosts:
    - host: 'MySQL server'
      (...)
      items:
        (...)
        - name: 'Binlog cache disk use'
          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: 'Buffer pool utilization'
          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: Uptime
        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	Type	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).
snmp_oid	string	(обязательно для элементов данных SNMP_AGENT) ID объекта SNMP.
key	string	(обязательно) Ключ элемента данных.
delay	string	Интервал обновления элемента данных. По умолчанию: 1m. Значение всегда будет 0 для элементов данных TRAP.
history	string	Период времени (с использованием суффикса времени, пользовательского макроса или LLD-макроса), в течение которого должны храниться исторические данные. По умолчанию: 31d.
trends	string	Период времени (с использованием суффикса времени, пользовательского макроса или LLD-макроса), 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 получаемого 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, которым разрешено отправлять данные для элемента данных, разделённых запятыми. Поддерживается для элементов данных TRAP и HTTP_AGENT.
units	string	Единицы измерения получаемого значения (bps, B и т. д.).
params	text	Дополнительные параметры в зависимости от типа элемента данных (выполняемый 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	Сенсор IPMI. Поддерживается для элементов данных IPMI.

Элемент	Type	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 SIMPLE, ODBC, JMX и HTTP_AGENT элементов данных. При использовании для элементов данных JMX также должен быть указан `password` (см. ниже), либо оба элемента должны быть оставлены пустыми.
password	string	(обязательно для элементов данных SSH и TELNET) Пароль для аутентификации. Поддерживается для SIMPLE, ODBC, JMX и HTTP_AGENT элементов данных. При использовании для элементов данных JMX также должен быть указан `username` (см. выше), либо оба элемента должны быть оставлены 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 этим элементом данных. Возможные значения:[1] NONE (0), ALIAS (4) и т. д. (поддерживаемые поля см. в Инвентаризации узла сети).
valuemap		Корневой элемент для карт значений элемента данных.
name	string	(обязательно) Имя карты значений, используемой для элемента данных.
logtimefmt	string	Формат времени в записях журнала. Поддерживается для элементов данных со значением типа LOG.
preprocessing		Корневой элемент для предобработки значений элемента данных.
step		Корневой элемент для шагов предобработки 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 элементов данных DEPENDENT) Корневой элемент для мастер-элемента данных зависимого элемента данных.
key	string	(обязательно) Ключ мастер-элемента данных зависимого элемента данных.
timeout	string	Тайм-аут запроса опроса данных элемента данных. Поддерживается для списка типов элементов данных Timeouts.
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 HTTP(S)-запроса. 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 опросе данных. Поддерживается для элементов данных HTTP_AGENT. Возможные значения:[1] NO (0), YES (1, по умолчанию).
post_type	string	Тип тела отправляемых данных. Поддерживается для элементов данных HTTP_AGENT. Возможные значения:[1] RAW (0, по умолчанию), JSON (2), XML (3).
http_proxy	string	Строка подключения к HTTP(S) прокси. Поддерживается для элементов данных HTTP_AGENT.
headers		Корневой элемент для заголовков HTTP(S)-запроса. Поддерживается для элементов данных HTTP_AGENT.

Элемент	Type	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 быть сохранена. 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	Как обрабатывать ответ. Поддерживается для элементов данных HTTP_AGENT. Возможные значения:[1] RAW (0, по умолчанию), JSON (1).
allow_traps	string	Разрешить заполнять значение 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	Следует ли проверять подлинность сертификата узла сети. Поддерживается для элементов данных HTTP_AGENT. Возможные значения:[1] NO (0, по умолчанию), YES (1).
verify_host	string	Следует ли проверять, что имя узла сети для подключения совпадает с именем в сертификате узла сети. Поддерживается для элементов данных HTTP_AGENT. Возможные значения:[1] NO (0, по умолчанию), YES (1).
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: Item object (обратитесь к соответствующему свойству с совпадающим именем).

Шаги предобработки 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
          (...)
```

Элемент	Type	Описание
type	string	(обязательно) Тип шага предобработки 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 как: полученное значение - предыдущее значение), CHANGE_PER_SECOND (10, вычисляется как: (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 предобработки значения элемента данных.
parameter	string	Отдельный параметр шага предобработки значения элемента данных.
error_handler	string	Тип действия, используемого в случае сбоя шага предобработки. 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 предобработки элемента данных (см. соответствующее свойство с совпадающим именем).

Триггеры элементов данных узла сети

```
zabbix_export:
  (...)
  hosts:
    - host: 'MySQL server'
      (...)
      items:
        (...)
        - name: Uptime
          (...)
          triggers:
            - expression: 'nodata(/MySQL server/mysql.uptime,30m)=1'
              name: 'MySQL: Failed to fetch info data'
              event_name: 'MySQL: Failed to fetch info data (or no data for 30m)'
              priority: INFO
              description: 'Zabbix has not received any data for items for the last 30 minutes.'
              dependencies:
                - name: 'MySQL: Service is down'
                  expression: 'last(/MySQL server/mysql.ping["{$MYSQL.DSN}","{$MYSQL.USER}","{$MYSQL.PASSW
              tags:
                - tag: scope
                  value: availability
            - expression: 'last(/MySQL server/mysql.uptime)<10m'
              name: 'MySQL: Service has been restarted'
              event_name: 'MySQL: Service has been restarted (uptime < 10m)'
              priority: INFO
              description: 'MySQL uptime is less than 10 minutes.'
              tags:
                - tag: scope
                  value: notice
        (...)
```

Элемент	Type	Описание
uuid	string	(обязательно) Уникальный идентификатор этого триггера.
expression	string	(обязательно) Выражение триггера.

Элемент	Type	Описание
recovery_mode	string	Основа для генерации OK событий. Возможные значения:[1] EXPRESSION (0, по умолчанию), 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, по умолчанию), TAG_VALUE (1).
correlation_tag	string	Имя тега, которое будет использоваться для корреляции событий.
name	string	(обязательно) 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	Состояние триггера. Возможные значения:[1] ENABLED (0, по умолчанию), DISABLED (1).
priority	string	Важность триггера. Возможные значения:[1] NOT_CLASSIFIED (0, по умолчанию), INFO (1), WARNING (2), AVERAGE (3), HIGH (4), DISASTER (5).
description	text	Описание триггера.
type	string	Тип генерации событий (одно событие о проблеме или несколько событий о проблеме). Возможные значения:[1] SINGLE (0, по умолчанию), MULTIPLE (1).
manual_close	string	Ручное закрытие событий о проблемах. Возможные значения:[1] NO (0, по умолчанию), YES (1).
dependencies		Корневой элемент для зависимостей.
name	string	(обязательно) Имя триггера зависимости.
expression	string	(обязательно) Выражение триггера зависимости.
recovery_expression	string	Выражение восстановления триггера зависимости.
tags		Корневой элемент для тегов триггера.
tag	string	(обязательно) Имя тега.
value	string	Значение тега.

Note:
См. также: Объект trigger (обратитесь к соответствующему свойству с совпадающим именем).

Правила низкоуровневого обнаружения узла сети

```
zabbix_export:
  (...)
  hosts:
    - host: 'MySQL server'
      (...)
      discovery_rules:
        - name: 'Database discovery'
          key: 'mysql.db.discovery["{$MYSQL.DSN}","{$MYSQL.USER}","{$MYSQL.PASSWORD}"]'
          delay: 1h
          filter: (see table below)
          description: 'Scanning databases in DBMS.'
          interface_ref: if1
          item_prototypes:
            - name: 'Size of database {#DATABASE}'
              key: 'mysql.db.size["{$MYSQL.DSN}","{$MYSQL.USER}","{$MYSQL.PASSWORD}","{#DATABASE}"]'
              delay: 5m
              units: B
              description: 'Database size.'
              preprocessing:
                - type: DISCARD_UNCHANGED_HEARTBEAT
                  parameters:
                    - 1h
              interface_ref: if1
              tags:
                - tag: component
```

```
                value: storage
              - tag: database
                value: '{#DATABASE}'
        lld_macro_paths:
          - lld_macro: '{#DATABASE}'
            path: $.Database
        preprocessing:
          - type: DISCARD_UNCHANGED_HEARTBEAT
            parameters:
              - 1d
    (...)
```

Attention:
Большинство элементов правил низкоуровневого обнаружения узла сети совпадают с элементами элементов данных узла сети. В таблице ниже описаны те элементы, которые отличаются от элементов данных узла сети.

Element	Type	Description
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), JMX (16), DEPENDENT (18), HTTP_AGENT (19), SNMP_AGENT (20), ITEM_TYPE_SCRIPT (21), ITEM_TYPE_BROWSER (22).
key	string	(обязательно) Ключ правила низкоуровневого обнаружения.
filter		Корневой элемент для фильтров правил низкоуровневого обнаружения узла сети.
lifetime	string	Период времени (в секундах, с использованием суффикса времени или пользовательского макроса), по истечении которого 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, с использованием суффикса времени или пользовательского макроса), по истечении которого 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 элементов данных узла сети, которые совпадают с элементами данных узла сети.
trigger_prototypes		Корневой элемент для элементов прототипов триггеров узла сети, которые совпадают с триггерами элементов данных узла сети.
graph_prototypes		Корневой элемент для прототипов графиков узла сети, которые совпадают с графиками узла сети.
host_prototypes		Корневой элемент для прототипов узлов сети, 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, являющийся прямым дочерним элементом указанного правила/прототипа правила.
key	string	(обязательно) Ключ родительского правила низкоуровневого обнаружения (или прототипа правила).
master_item	string	(обязательно для правил DEPENDENT) Корневой элемент для мастер-элемента данных зависимого правила.
lld_macro_paths		Корневой элемент для путей макросов правила низкоуровневого обнаружения.
lld_macro	string	(обязательно) Имя макроса правила низкоуровневого обнаружения.
path	string	(обязательно) Селектор значения, которое будет присвоено соответствующему макросу.
preprocessing		Корневой элемент для предобработки значений правила низкоуровневого обнаружения.
step		Корневой элемент для элементов шагов предобработки значений правила низкоуровневого обнаружения, которые совпадают с шагами предобработки значений элемента данных узла сети, за исключением меньшего количества возможных значений. См. также: объект предобработки правила LLD.

Element	Type	Description
overrides		Корневой элемент для правил переопределения правила низкоуровневого обнаружения.
name	string	(обязательно) Уникальное имя переопределения.
step	string	(обязательно) Уникальный порядковый номер переопределения.
stop	string	Остановить обработку следующих переопределений при совпадении.
filter		Корневой элемент для элементов фильтра правила переопределения правила низкоуровневого обнаружения, которые совпадают с фильтрами правил низкоуровневого обнаружения узла сети.
operations		Корневой элемент для операций переопределения правила низкоуровневого обнаружения узла сети.

Note:
См. также: объект правила LLD (обратитесь к соответствующему свойству с совпадающим именем).

Фильтры правил низкоуровневого обнаружения узла сети

```
zabbix_export:
  (...)
  hosts:
    - host: 'MySQL server'
      (...)
      discovery_rules:
        - name: '              '
          (...)
          filter:
            evaltype: AND
            conditions:
              - macro: '{#DATABASE}'
                value: '{$MYSQL.DBNAME.MATCHES}'
              - macro: '{#DATABASE}'
                value: '{$MYSQL.DBNAME.NOT_MATCHES}'
                operator: NOT_MATCHES_REGEX
          (...)
```

Элемент	Type	Описание
evaltype	string	Переопределяет метод вычисления 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	(обязательно) Имя макроса низкоуровневого обнаружения, по которому выполняется проверка.
value	string	Значение для сравнения.
operator	string	Оператор условия. Возможные значения:[1] MATCHES_REGEX (8, по умолчанию), NOT_MATCHES_REGEX (9).
formulaid	string	(обязательно) 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.

Note:
См. также: Объект фильтра правила LLD (обратитесь к соответствующему свойству с совпадающим именем).

Операции переопределения правила низкоуровневого обнаружения узла сети

Элемент	Тип	Описание
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 при выполнении операции переопределения.
discover	string	Определяет, добавляется ли объект в результате обнаружения.
delay	string	Интервал обновления, установленный для прототипа элемента данных при выполнении операции переопределения.
history	string	Период хранения истории, установленный для прототипа элемента данных при выполнении операции переопределения.
trends	string	Период хранения трендов, установленный для прототипа элемента данных при выполнении операции переопределения.
severity	string	Важность прототипа триггера, установленная при выполнении операции переопределения.
tags		Корневой элемент для тегов, установленных для объекта при выполнении операции переопределения.
tag	string	(обязательно) Имя тега.
value	string	Значение тега.
templates		Корневой элемент для шаблонов, привязанных к прототипу узла сети при выполнении операции переопределения.
name	string	(обязательно) Имя шаблона.
inventory_mode	string	Режим инвентаризации прототипа узла сети, установленный при выполнении операции переопределения.

Note:
См. также: Объект операции переопределения правила LLD (обратитесь к соответствующему свойству с совпадающим именем).

Веб-сценарии узла сети

Элемент	Тип	Описание
uuid	string	(обязательно) Уникальный идентификатор этого веб-сценария.
name	string	(обязательно) Имя веб-сценария.
delay	string	Частота выполнения веб-сценария (в секундах, с использованием суффикса времени или пользовательского макроса). По умолчанию: 1m.
attempts	integer	Количество попыток выполнения шагов веб-сценария. Возможные значения: 1-10 (по умолчанию: 1).
agent	string	Клиентский агент. Zabbix будет представляться выбранным браузером. Это полезно, если 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	(обязательно) Имя переменной.
value	text	(обязательно) Значение переменной.
headers		Корневой элемент для HTTP-заголовков, отправляемых при выполнении запроса. Заголовки должны быть перечислены с использованием того же синтаксиса, в котором они появляются в протоколе HTTP.
name	string	(обязательно) Имя заголовка.
value	text	(обязательно) Значение заголовка.
status	string	Состояние веб-сценария. Возможные значения:[1] ENABLED (0, по умолчанию), DISABLED (1).

Элемент	Тип	Описание
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 SSL-ключа, используемого для аутентификации клиента (должен быть в формате PEM).
ssl_key_password	string	Пароль файла закрытого SSL-ключа.
steps		(обязательно) Корневой элемент для шагов веб-сценария узла сети.
tags		Корневой элемент для тегов веб-сценария.
tag	string	(обязательно) Имя тега.
value	string	Значение тега.

Note:
См. также: Объект веб-сценария (см. соответствующее свойство с совпадающим именем).

Шаги веб-сценария узла сети

Элемент	Тип	Описание
name	string	(обязательно) Имя шага веб-сценария.
url	string	(обязательно) URL для мониторинга.
query_fields		Корневой элемент для параметров запроса (массив HTTP-полей, добавляемых к URL при выполнении запроса).
name	string	(обязательно) Имя параметра запроса.
value	string	Значение параметра запроса.
posts		Корневой элемент для переменных HTTP POST (строка (необработанные данные post) или массив HTTP-полей (данные полей формы)).
name	string	(обязательно) Имя поля post.
value	string	(обязательно) Значение поля post.
variables		Корневой элемент переменных уровня шага (макросов), которые должны применяться после этого шага. Если значение переменной имеет префикс 'regex:', то её значение извлекается из данных, возвращённых этим шагом, в соответствии с шаблоном регулярного выражения после префикса 'regex:'
name	string	(обязательно) Имя переменной.
value	text	(обязательно) Значение переменной.
headers		Корневой элемент для HTTP-заголовков, отправляемых при выполнении запроса.
name	string	(обязательно) Имя заголовка.
value	text	(обязательно) 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 выполнения шага (в секундах, с использованием суффикса времени или пользовательского макроса). По умолчанию: 15s.
required	string	Текст, который должен присутствовать в ответе (игнорируется, если пусто).
status_codes	string	Список допустимых HTTP-кодов состояния, разделённых запятыми (например, 200-201,210-299; игнорируется, если пусто).

Note:
См. также: Объект шага веб-сценария (обратитесь к соответствующему свойству с совпадающим именем).

Графики узла сети

```
zabbix_export:
  (...)
  graphs:
    - name: 'MySQL: Bandwidth'
      graph_items: ( .         )
    (...)
```

Элемент	Type	Описание
uuid	string	Уникальный идентификатор графика.
name	string	(обязательно) Имя графика.
width	integer	Ширина графика в пикселях. Используется для предпросмотра, а также для круговых/взрывных графиков. Возможные значения:[1] 20-65535 (по умолчанию: 900).
height	integer	Высота графика в пикселях. Используется для предпросмотра, а также для круговых/взрывных графиков. Возможные значения:[1] 20-65535 (по умолчанию: 900).
yaxismin	double	Минимальное значение оси Y. Поддерживается для FIXED минимального значения оси Y. По умолчанию: 0.
yaxismax	double	Максимальное значение 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 стиль 3D. 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 отдельных сведений об элементе данных.
host	string	(обязательно) Узел сети элемента данных.
key	string	(обязательно) Ключ элемента данных.
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 отдельных сведений об 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 (обратитесь к соответствующему свойству с совпадающим именем).

Элементы данных 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	Type	Описание
sortorder	integer	Порядок отрисовки. Меньшее значение отрисовывается первым. Может использоваться для отрисовки линий или областей позади (или перед) другими.
drawtype	string	Стиль отрисовки элемента данных 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 отрисована шкала Y элемента данных графика. Поддерживается для графиков NORMAL и STACKED graphs.
calc_fnc	string	Данные для отрисовки, если для элемента данных 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/разделённых графиков).
type	string	Тип элемента данных графика. Возможные значения:[1] SIMPLE (0, по умолчанию), GRAPH_SUM (2; значение элемента данных представляет всю круговую диаграмму; поддерживается для круговых/разделённых графиков).
item		(обязательно) Отдельный элемент данных.
host	string	(обязательно) Узел сети элемента данных.
key	string	(обязательно) Ключ элемента данных.

Note:
См. также: Graph item object (обратитесь к соответствующему свойству с совпадающим именем).

Карты значений узла сети

```
zabbix_export:
  (...)
  hosts:
    - host: 'MySQL server'
      (...)
      valuemaps:
        - name:
          mappings:
            - value: '1'
```

```
            newvalue:
(...)
```

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 совпадения сопоставления. 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 преобразуется исходное значение.

Note:
См. также: Объект карты значений (обратитесь к соответствующему свойству с совпадающим именем).

Сноски [1] Значения API типа integer в скобках, например ENABLED (0), упоминаются только для справки. Для получения дополнительной информации смотрите связанную страницу объекта API в записи таблицы или в конце каждого раздела.

5 Карты сети

Обзор

Экспорт карты сети содержит:

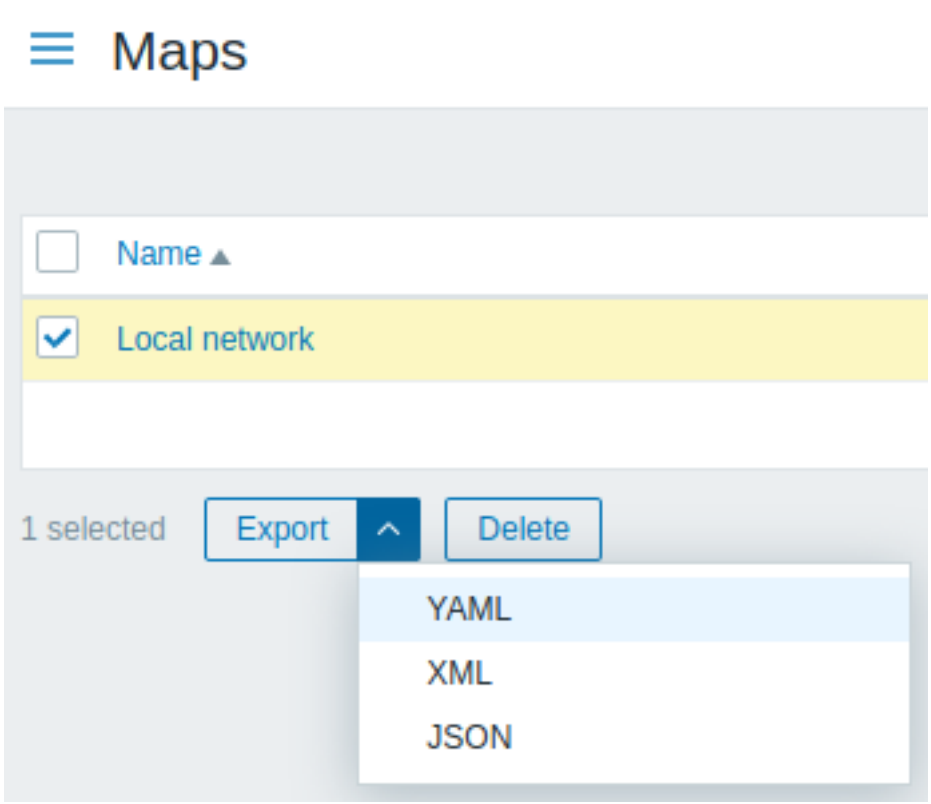
- Все связанные изображения
- Структуру карты (все настройки карты, все содержащиеся на ней элементы с их настройками, связи карты и индикаторы состояния связей карты)

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, её импорт завершится ошибкой.

Экспорт

Чтобы экспортировать карты сети, выполните следующие действия:

1. Перейдите в *Мониторинг* → *Карты сети*.
2. Установите флажки у карт сети, которые нужно экспортировать.
3. Нажмите *Экспорт* под списком.



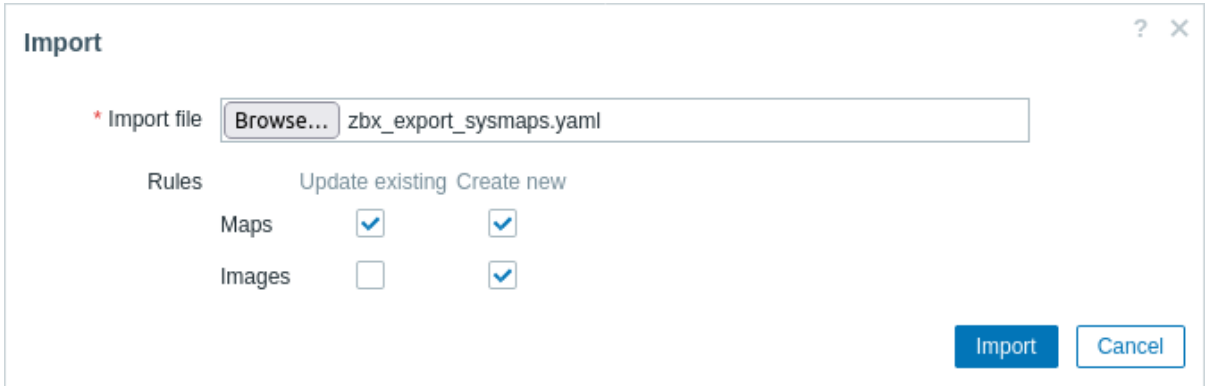
В зависимости от выбранного формата карты экспортируются в локальный файл с именем по умолчанию:

- `zabbix_export_maps.yaml` — при экспорте в YAML (вариант экспорта по умолчанию);
- `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 *Мониторинг* → *Карты*.
2. Нажмите *Импорт* в правом верхнем углу.
3. Выберите файл импорта.
4. Отметьте необходимые параметры в правилах импорта.
5. Нажмите *Импорт* в правом нижнем углу формы настройки.



Правила импорта:

Правило	Описание
Обновить существующие	Существующие карты будут обновлены с использованием данных из файла импорта. В противном случае они не будут обновлены.
Создать новые	Новые карты будут созданы с использованием данных из файла импорта. В противном случае они не будут созданы.

Если снять оба флажка с параметров карт и отметить соответствующие параметры для изображений, будут импортированы только изображения. Импорт изображений доступен только пользователям с ролью *Супер-администратор*.

Сообщение об успешном или неуспешном завершении импорта будет отображено в веб-интерфейсе.

Warning:
При замене существующего изображения это повлияет на все карты, использующие данное изображение.

Формат экспорта

Экспорт в YAML:

```yaml
zabbix_export:
  version: '8.0'
  images:
    - name: Zabbix_server_3D_(128)
      imagetype: '1'
      encodedImage: iVBOR...5CYII=
  maps:
    - name: '        '
      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: {  }
```

Экспортируемые элементы Экспортируемые элементы описаны в таблице ниже.

Элемент	Тип	Описание
images		Корневой элемент для изображений.
name	string	Уникальное имя изображения.
imagetype	integer	Тип изображения. Возможные значения: 1 - Изображение; 2 - Фон.
encodedImage	string	Изображение, закодированное в Base64.
maps		Корневой элемент для карт.

Карты

Элемент	Тип	Описание
name	string	Уникальное имя карты.
width	integer	Ширина карты в пикселях.
height	integer	Высота карты в пикселях.
label_type	integer	Тип подписи элемента карты. Возможные значения: 0 - Подпись; 1 - IP-адрес узла сети; 2 - Имя элемента; 3 - Только состояние; 4 - Ничего.
label_location	integer	Расположение подписи элемента карты по умолчанию. Возможные значения: 0 - Снизу; 1 - Слева; 2 - Справа; 3 - Сверху.
highlight	integer	Включить подсветку иконок для активных триггеров и состояний узлов сети. Возможные значения: 0 - Нет; 1 - Да.

Элемент	Тип	Описание
expandproblem	integer	Отображать проблемный триггер для элементов с одной проблемой. Возможные значения: 0 - Нет; 1 - Да.
markelements	integer	Подсвечивать элементы карты, у которых недавно изменилось состояние. Возможные значения: 0 - Нет; 1 - Да.
show_unack	integer	Отображение проблем. Возможные значения: 0 - Количество всех проблем; 1 - Количество неподтвержденных проблем; 2 - Количество подтвержденных и неподтвержденных проблем по отдельности.
severity_min	integer	Минимальная важность триггера, отображаемая на карте по умолчанию. Возможные значения: 0 - Не классифицировано; 1 - Информация; 2 - Предупреждение; 3 - Средняя; 4 - Высокая; 5 - Катастрофа.
show_suppressed	integer	Отображать проблемы, которые в противном случае были бы подавлены (не показаны) 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 конфигурации карты. Возможные значения: 0 - Да; 1 - Нет.
grid_align	integer	Автоматически выравнивать иконки в конфигурации карты. Возможные значения: 0 - Да; 1 - Нет.
label_format	integer	Использовать расширенную конфигурацию подписей. Возможные значения: 0 - Нет; 1 - Да.
label_type_host	integer	Отображать подпись как подпись узла сети. Поддерживается, если `label_format` установлено в 1. Возможные значения: 0 - Подпись; 1 - IP-адрес узла сети; 2 - Имя элемента; 3 - Только состояние; 4 - Ничего; 5 - Пользовательская подпись.
label_type_hostgroup	integer	Отображать подпись как подпись группы узлов сети. Поддерживается, если `label_format` установлено в 1. Возможные значения: 0 - Подпись; 2 - Имя элемента; 3 - Только состояние; 4 - Ничего; 5 - Пользовательская подпись.

Элемент	Тип	Описание
label_type_trigger	integer	Отображать подпись как подпись триггера. Поддерживается, если `label_format` установлено в 1. Возможные значения: 0 - Подпись; 2 - Имя элемента; 3 - Только состояние; 4 - Ничего; 5 - Пользовательская подпись.
label_type_map	integer	Отображать подпись как подпись карты. Поддерживается, если `label_format` установлено в 1. Возможные значения: 0 - Подпись; 2 - Имя элемента; 3 - Только состояние; 4 - Ничего; 5 - Пользовательская подпись.
label_type_image	integer	Отображать подпись как подпись изображения. Поддерживается, если `label_format` установлено в 1. Возможные значения: 0 - Подпись; 2 - Имя элемента; 4 - Ничего; 5 - Пользовательская подпись.
label_string_host	string	Пользовательская подпись для элементов узла сети. Поддерживается, если `label_type_host` установлено в 5.
label_string_hostgroup	string	Пользовательская подпись для элементов группы узлов сети. Поддерживается, если `label_type_hostgroup` установлено в 5.
label_string_trigger	string	Пользовательская подпись для элементов триггера. Поддерживается, если `label_type_trigger` установлено в 5.
label_string_map	string	Пользовательская подпись для элементов карты. Поддерживается, если `label_type_map` установлено в 5.
label_string_image	string	Пользовательская подпись для элементов изображения. Поддерживается, если `label_type_image` установлено в 5.
expand_macros	integer	Разворачивать макросы в подписях в конфигурации 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 иконок (если есть).
name	string	Имя сопоставления иконок.
urls		Корневой элемент для URL-адресов, 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.

Элемент		Тип	Описание
	width	integer	Ширина фигуры.
	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 связей между элементами карты.
	drawtype	integer	Стиль связи. Возможные значения: 0 - Линия; 2 - Жирная линия; 3 - Пунктирная линия; 4 - Штриховая линия.

Элемент	Тип	Описание
color	string	Цвет связи (6 символов, hex).
label	string	Подпись связи.
selementid1	id	ID одного из соединяемых элементов.
selementid2	id	ID другого соединяемого элемента.
linktriggers		Корневой элемент для индикаторов состояния связей.

Note:
См. также: Объект карты (обратитесь к соответствующему свойству с совпадающим именем).

Элементы карты

Элемент	Тип	Описание
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 представлены на карте. Все сущности имеют соответствующий им элемент (`host` и т. д.; см., например, Формат экспорта).
label	string	Метка значка.
label_location	integer	Расположение метки. Возможные значения: -1 - Использовать значение карты по умолчанию; 0 - Снизу; 1 - Слева; 2 - Справа; 3 - Сверху.
x	integer	Расположение по оси X.
y	integer	Расположение по оси Y.
elementsubtype	integer	Подтип элемента. Поддерживается, если `elementtype` установлен в 3. Возможные значения: 0 - Одна группа узлов сети; 1 - Все группы узлов сети.
areatype	integer	Размер области. Поддерживается, если `elementtype` установлен в 1. Возможные значения: 0 - Такой же, как у всей карты; 1 - Пользовательский размер.
width	integer	Ширина области. Поддерживается, если `areatype` установлен в 1.
height	integer	Высота области. Поддерживается, если `areatype` установлен в 1.
viewtype	integer	Алгоритм размещения области. Поддерживается, если `elementsubtype` установлен в 1. Возможные значения: 0 - Равномерно размещать в области.
use_iconmap	integer	Использовать сопоставление значков для этого элемента. Актуально только если сопоставление значков активировано 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`.

Элемент	Тип	Описание
name	string	Уникальное имя изображения.
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-адресов, используемых картами или каждым элементом карты.
name	string	Имя ссылки.
url	string	URL-адрес ссылки.
evaltype	integer	Тип вычисления для тегов.
tags		Корневой элемент для тегов проблем (для элементов узла сети и группы узлов сети). Если теги указаны, на карте будут отображаться только проблемы с этими тегами.
tag	string	Имя тега.
value	string	Значение тега.
operator	integer	Оператор.

Note:
См. также: Объект элемента карты (обратитесь к соответствующему свойству с совпадающим именем).

Индикаторы состояния ссылки карты

Элемент	Тип	Описание
drawtype	integer	Стиль ссылки, когда триггер находится в состоянии «Проблема». Возможные значения: 0 - Линия; 2 - Жирная линия; 3 - Пунктирная линия; 4 - Штриховая линия.
color	string	Цвет ссылки (6 символов, hex), когда триггер находится в состоянии «Проблема».
trigger		Корневой элемент для триггера, используемого для индикации состояния ссылки.
description	string	Имя триггера.
expression	string	Выражение триггера.
recovery_expression	string	Выражение восстановления триггера.

Note:
См. также: Объект триггера ссылки карты (обратитесь к соответствующему свойству с совпадающим именем).

6 Способы оповещения

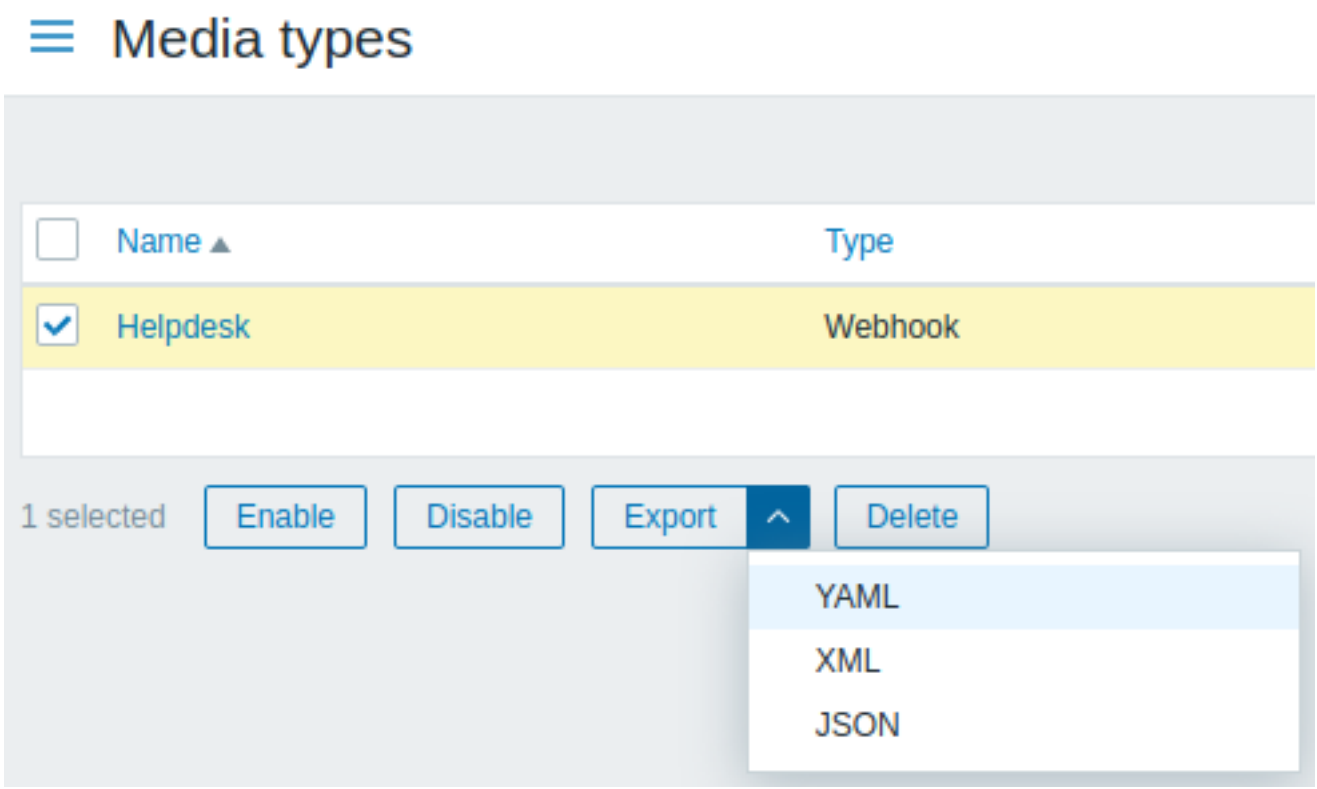
Обзор

Типы оповещений экспортируются вместе со всеми связанными объектами и связями между объектами.

Экспорт

Чтобы экспортировать типы медиа, выполните следующие действия:

1. Перейдите в *Оповещения → Типы медиа*.
2. Установите флажки у типов медиа, которые нужно экспортировать.
3. Нажмите *Экспорт* под списком.



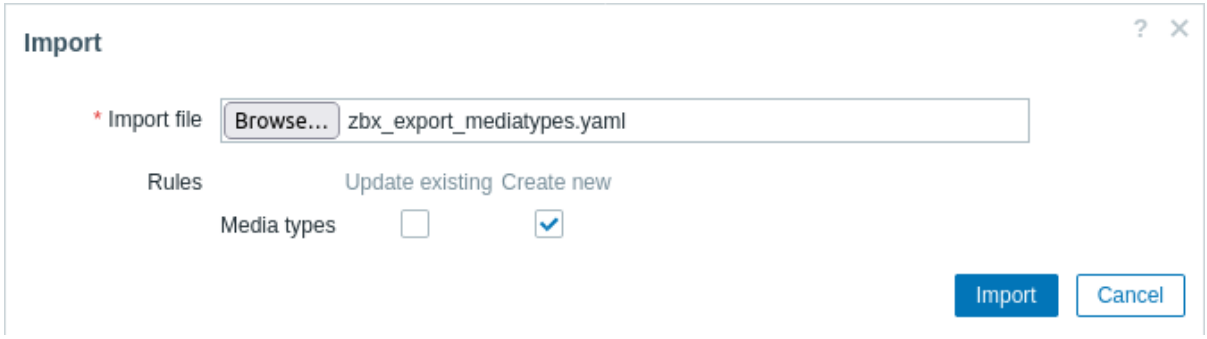
В зависимости от выбранного формата типы медиа экспортируются в локальный файл с именем по умолчанию:

- `zabbix_export_mediatypes.yaml` — при экспорте в YAML (вариант по умолчанию для экспорта);
- `zabbix_export_mediatypes.xml` — при экспорте в XML;
- `zabbix_export_mediatypes.json` — при экспорте в JSON.

Импорт

Чтобы импортировать способы оповещения, выполните следующие действия:

1. Перейдите в *Оповещения* → *Способы оповещения*.
2. Нажмите *Импортировать* в правом верхнем углу.
3. Выберите файл импорта.
4. Отметьте необходимые параметры в правилах импорта.
5. Нажмите *Импортировать* в правом нижнем углу формы настройки.



Правила импорта:

Rule	Description
Update existing	Существующие элементы будут обновлены с использованием данных из файла импорта. В противном случае они не будут обновлены.
Create new	Новые элементы будут созданы с использованием данных из файла импорта. В противном случае они не будут созданы.

Сообщение об успешном или неуспешном импорте будет отображено в веб-интерфейсе.

Формат экспорта

Экспорт в YAML:

```yaml
zabbix_export:
  version: '8.0'
  media_types:
    - name: Pushover
      type: WEBHOOK
      parameters:
        - name: endpoint
          value: 'https://api.pushover.net/1/messages.json'
        - name: eventid
          value: '{EVENT.ID}'
        - name: event_nseverity
          value: '{EVENT.NSEVERITY}'
        - name: event_source
          value: '{EVENT.SOURCE}'
        - name: event_value
          value: '{EVENT.VALUE}'
        - name: expire
          value: '1200'
        - name: message
          value: '{ALERT.MESSAGE}'
        - name: priority_average
          value: '0'
        - name: priority_default
          value: '0'
        - name: priority_disaster
          value: '0'
        - name: priority_high
          value: '0'
        - name: priority_information
          value: '0'
        - name: priority_not_classified
          value: '0'
        - name: priority_warning
          value: '0'
        - name: retry
          value: '60'
        - name: title
          value: '{ALERT.SUBJECT}'
        - name: token
          value: '<PUSHOVER TOKEN HERE>'
        - name: triggerid
          value: '{TRIGGER.ID}'
        - name: url
          value: '{$ZABBIX.URL}'
        - name: url_title
          value: Zabbix
        - name: user
          value: '{ALERT.SENDTO}'
      status: DISABLED
      max_sessions: '0'
      script: |
        try {
            var params = JSON.parse(value),
                request = new HttpRequest(),
                data,
                response,
                severities = [
                    {name: 'not_classified', color: '#97AAB3'},
                    {name: 'information', color: '#7499FF'},
                    {name: 'warning', color: '#FFC859'},
                    {name: 'average', color: '#FFA059'},
                    {name: 'high', color: '#E97659'},
```

```
            {name: 'disaster', color: '#E45959'},
            {name: 'resolved', color: '#009900'},
            {name: 'default', color: '#000000'}
        ],
        priority;

    if (typeof params.HTTPProxy === 'string' && params.HTTPProxy.trim() !== '') {
        request.setProxy(params.HTTPProxy);
    }

    if ([0, 1, 2, 3].indexOf(parseInt(params.event_source)) === -1) {
        throw 'Incorrect "event_source" parameter given: "' + params.event_source + '".\nMust be 0
    }

    if (params.event_value !== '0' && params.event_value !== '1'
        && (params.event_source === '0' || params.event_source === '3')) {
        throw 'Incorrect "event_value" parameter given: ' + params.event_value + '\nMust be 0 or 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" should be -2..2';
    }

    if (params.event_source === '0' && isNaN(params.triggerid)) {
        throw 'field "triggerid" is not a number';
    }

    if (isNaN(params.eventid)) {
        throw 'field "eventid" is not a number';
    }

    if (typeof params.message !== 'string' || params.message.trim() === '') {
        throw 'field "message" cannot be empty';
    }

    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 'field "retry" should be a number with value of at least 30 if "priority" is set
        }
```

```
            if (isNaN(params.expire) || params.expire > 10800) {
                throw 'field "expire" should be a number with value of at most 10800 if "priority" is
            }

            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/templat

             token                       Pushover.
              Pushover        Zabbix —                      send to.
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.

Элемент	Тип	Описание
name	string	(обязательно) Имя типа медиа.
type	string	(обязательно) Транспорт, используемый типом медиа. Возможные значения:[1] EMAIL (0), SMS (1), SCRIPT (2), WEBHOOK (4).
status	string	Включен ли тип медиа. Возможные значения:[1] ENABLED (0, по умолчанию), DISABLED (1)
max_sessions	integer	Максимальное количество оповещений, которые могут обрабатываться параллельно. Возможные значения для SMS:[1] 1 (по умолчанию). Возможные значения для других типов медиа:[1] 0-100 (где 0 — без ограничений).
attempts	integer	Максимальное количество попыток отправки оповещения. Возможные значения:[1] 1-10 (по умолчанию: 3).
attempt_interval	string	Интервал (в секундах или с использованием суффикса времени) между повторными попытками. Возможные значения:[1] 0-60s (по умолчанию: 10s).
description	string	Описание типа медиа.
message_templates		Корневой элемент для шаблонов сообщений типа медиа.
event_source	string	(обязательно) Источник события. Возможные значения:[1] TRIGGERS (0), DISCOVERY (1), AUTOREGISTRATION (2), INTERNAL (3), SERVICE (4).

Элемент	Тип	Описание
operation_mode	string	Режим операции. Возможные значения:[1] PROBLEM (0), RECOVERY (1), UPDATE (2).
subject	string	Тема сообщения.
message	string	Тело сообщения.

Note:
См. также: Объект типа медиа (обратитесь к соответствующему свойству с таким же именем).

Email

Следующие дополнительные элементы экспортируются только для типа медиа *Email*.

Элемент	Тип	Описание
provider	string	Провайдер 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 соединения SMTP, который следует использовать. Возможные значения:[1] NONE (0, по умолчанию), STARTTLS (1), SSL_OR_TLS (2).
smtp_verify_host	string	Проверка SSL-узла для SMTP. Возможные значения:[1] NO (0, по умолчанию), YES (1).
smtp_verify_peer	string	Проверка SSL-пира для SMTP. Возможные значения:[1] NO (0, по умолчанию), YES (1).
smtp_authentication	string	Метод аутентификации SMTP, который следует использовать. 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 перенаправления обратно после OAuth-авторизации.
client_id	string	Идентификатор клиента, зарегистрированный на сервере OAuth-авторизации.
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 типа медиа (см. соответствующее свойство с таким же именем).

SMS

Следующие дополнительные элементы экспортируются только для типа медиа *SMS*.

Элемент	Тип	Описание
gsm_modem	string	(обязательно) Имя последовательного устройства GSM-модема.

Note:
См. также: Объект типа медиа (см. соответствующее свойство с таким же именем).

Скрипт

Следующие дополнительные элементы экспортируются только для типа медиа *Скрипт*.

Элемент	Тип	Описание
имя скрипта	string	(обязательно) Имя скрипта.
parameters		Корневой элемент для параметров скрипта.
sortorder	string	(обязательно) Порядок параметров скрипта, передаваемых в скрипт как аргументы командной строки.
value	string	Значение параметра скрипта.

Note:
См. также: Объект типа медиа (см. соответствующее свойство с таким же именем).

Вебхук

Следующие дополнительные элементы экспортируются только для типа медиа *Вебхук*.

Элемент	Тип	Описание
script	string	Скрипт.
timeout	string	Интервал тайм-аута HTTP-запроса скрипта Javascript. Возможные значения:[1] 1-60s (по умолчанию: 30s).
process_tags	string	Следует ли обрабатывать возвращённые теги. Возможные значения:[1] NO (0, по умолчанию), YES (1).
show_event_menu	string	Указывает на наличие записи в меню события, если макрос {EVENT.TAGS.*} был успешно разрешён в полях `event_menu_url` и `event_menu_name`. Возможные значения:[1] NO (0, по умолчанию), YES (1).
event_menu_url	string	URL записи меню события. Поддерживает макрос {EVENT.TAGS.*}.
event_menu_name	string	Имя записи меню события. Поддерживает макрос {EVENT.TAGS.*}.
parameters		Корневой элемент для параметров типа медиа вебхук.
name	string	(обязательно) Имя параметра вебхука.
value	string	Значение параметра вебхука.

Note:
См. также: Объект типа медиа (обратитесь к соответствующему свойству с совпадающим именем).

Сноски

[1] Значения API типа integer в скобках, например ENABLED (0), указаны только для справки. Для получения дополнительной информации смотрите связанную страницу объекта API в записи таблицы или в конце каждого раздела.

13 Обнаружение

Пожалуйста, используйте боковую панель для доступа к содержимому раздела «Обнаружение».

1 Обнаружение сети

Обзор

Zabbix предлагает функциональность автоматического обнаружения сети, которая является эффективной и очень гибкой.

При правильной настройке обнаружения сети вы можете:

- ускорить развертывание Zabbix
- упростить администрирование
- использовать Zabbix в быстро меняющихся средах без избыточного администрирования

Обнаружение сети в Zabbix основано на следующей информации:

- диапазоны IP-адресов

- доступность внешних сервисов (FTP, SSH, WEB, POP3, IMAP, TCP и т. д.)
- информация, полученная от Zabbix агент (поддерживается только незашифрованный режим)
- информация, полученная от SNMP-агент

Оно НЕ предоставляет:

- обнаружение топологии сети

Обнаружение сети в основном состоит из двух фаз: обнаружение и действия.

Обнаружение

Zabbix периодически сканирует диапазоны IP-адресов, заданные в правилах обнаружения сети. Частота проверки настраивается отдельно для каждого правила.

Для каждого правила задан набор проверок сервисов, 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 воркерами обнаружения (их количество настраивается в веб-интерфейсе отдельно для каждого правила). 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 приблизительно 2000000 или 4 ГБ памяти. Если очередь переполнится, правило обнаружения будет пропущено, а в журнал будет выведено предупреждающее сообщение. 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 сервис (или завершаются ошибкой), остальные проверки все равно будут обработаны.

Note:
Если правило обнаружения изменено во время выполнения, текущее выполнение обнаружения будет прервано.

Каждая проверка сервиса и узла сети (IP), выполняемая модулем обнаружения сети, генерирует событие обнаружения.

Событие	Результат проверки сервиса
Сервис обнаружен	Сервис находится в состоянии 'up' после состояния 'down' или при обнаружении в первый раз.
Сервис доступен	Сервис находится в состоянии 'up' после того, как он уже был в состоянии 'up'.
Сервис потерян	Сервис находится в состоянии 'down' после состояния 'up'.
Сервис недоступен	Сервис находится в состоянии 'down' после того, как он уже был в состоянии 'down'.
Узел сети обнаружен	По крайней мере один сервис узла сети находится в состоянии 'up' после того, как все сервисы этого узла сети были в состоянии 'down', либо обнаружен сервис, принадлежащий незарегистрированному узлу сети.
Узел сети доступен	По крайней мере один сервис узла сети находится в состоянии 'up' после того, как по крайней мере один сервис уже был в состоянии 'up'.
Узел сети потерян	Все сервисы узла сети находятся в состоянии 'down' после того, как по крайней мере один из них был в состоянии 'up'.
Узел сети недоступен	Все сервисы узла сети находятся в состоянии 'down' после того, как они уже были в состоянии 'down'.

Действия

События обнаружения могут служить основой для соответствующих действий, таких как:

- Отправка уведомлений
- Добавление/удаление узлов сети
- Включение/отключение узлов сети
- Добавление узлов сети в группу
- Удаление узлов сети из группы
- Добавление тегов к узлу сети
- Удаление тегов с узла сети
- Привязка шаблона к узлам сети/отвязка шаблона от узлов сети
- Выполнение удалённых скриптов

Эти действия можно настроить с учётом типа устройства, IP-адреса, состояния, времени работы/простоя и т. д. Подробную информацию о настройке действий для событий на основе обнаружения сети см. на страницах операций и условий действия.

Поскольку действия обнаружения сети основаны на событиях, они будут срабатывать как тогда, когда обнаруженный узел сети находится в сети, так и тогда, когда он недоступен. Настоятельно рекомендуется добавить условие действия *Состояние обнаружения: доступен*, чтобы избежать срабатывания таких действий, как *Добавить узел сети*, при событиях *Сервис потерян*/*Сервис недоступен*. В противном случае, если обнаруженный узел сети удалён вручную, он всё равно будет генерировать события *Сервис потерян*/*Сервис недоступен* и будет создан заново во время следующего цикла обнаружения.

Note:
Привязка шаблонов к обнаруженному узлу сети завершится общей ошибкой, если любой из привязываемых шаблонов содержит уникальную сущность (например, ключ элемента данных), совпадающую с уникальной сущностью (например, ключом элемента данных), уже существующей на узле сети или в другом из привязываемых шаблонов.

Создание узла сети

Узел сети добавляется, если выбрана операция *Add host*. Узел сети также будет добавлен, даже если операция *Add host* отсутствует, если вы выберете операции, приводящие к действиям над узлом сети. Такими операциями являются:

- включить узел сети
- отключить узел сети
- добавить узел сети в группу узлов сети
- привязать шаблон к узлу 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 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 в качестве критерия для поиска узла сети в системе. Если узел сети с тем же IP-адресом, типом интерфейса и источником обнаружения уже существует, именно этот узел сети будет целевым 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 изменён или интерфейс удалён, при следующем обнаружении будет создан новый узел сети.

Именование узлов сети

При добавлении узлов сети имя узла сети определяется по результату обратного DNS-запроса или по IP-адресу, если обратное разрешение имени не удалось. Поиск выполняется с сервера Zabbix или прокси Zabbix, в зависимости от того, кто выполняет обнаружение. Если поиск не удался на прокси, на сервере он повторно не выполняется. Если узел сети с таким именем уже существует, к имени следующего узла сети будет добавлено **_2**, затем **_3** и так далее.

Также можно не использовать DNS/IP-поиск и вместо этого использовать значение элемента данных в качестве имени узла сети, например:

- Можно обнаружить несколько серверов с запущенным агентом Zabbix, используя элемент данных агента Zabbix для обнаружения, и автоматически назначить им корректные имена на основе строкового значения, возвращаемого этим элементом данных
- Можно обнаружить несколько сетевых устройств SNMP, используя элемент данных агента SNMP для обнаружения, и автоматически назначить им корректные имена на основе строкового значения, возвращаемого этим элементом данных

Если имя узла сети было задано с использованием значения элемента данных, оно не обновляется при последующих проверках обнаружения. Если задать имя узла сети с использованием значения элемента данных невозможно, используется значение по умолчанию (имя DNS).

Если узел сети с обнаруженным IP-адресом уже существует и источник обнаружения (сервер Zabbix, прокси или группа прокси) не изменился, новый узел сети не создается. Если источник обнаружения отличается, обнаруженный объект рассматривается как отдельный, и может быть создан новый узел сети. Однако если действие обнаружения содержит операции (привязка шаблона, добавление в группу узлов сети и т. д.), они выполняются для существующего узла сети, который совпадает по IP-адресу, типу интерфейса и источнику обнаружения.

Удаление узла сети

Узлы сети, обнаруженные правилом сетевого обнаружения, автоматически удаляются из *Monitoring* > *Discovery*, если обнаруженный объект больше не находится в диапазоне IP-адресов 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 сервисы — например, если проверка SNMP выполнена успешно, будет создан интерфейс SNMP.
- Если узел сети ответил как на запросы Zabbix агент, так и на запросы SNMP, будут созданы оба типа интерфейсов.
- Если критериями уникальности являются данные, возвращаемые Zabbix агент или SNMP, первый найденный для узла сети интерфейс будет создан как интерфейс по умолчанию. Другие IP-адреса будут добавлены как дополнительные интерфейсы. Условия действия (например, Host IP) не влияют на добавление интерфейсов. *Обратите внимание*, что 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 ответил только на проверки агента, он будет создан только с интерфейсом агента. Если позже он начнет отвечать на SNMP, будут добавлены дополнительные интерфейсы SNMP.
- Если изначально были созданы 3 отдельных узла сети, обнаруженные по критерию уникальности "IP", а затем правило обнаружения изменено так, что узлы сети A, B и C имеют одинаковый результат критерия уникальности, B и C будут созданы как дополнительные интерфейсы для A, первого узла сети. Отдельные узлы сети B и C сохранятся. В *Monitoring* > *Discovery* добавленные интерфейсы будут отображаться в столбце "Discovered device" черным шрифтом и с отступом, но в столбце "Monitored host" будет отображаться только A, первый созданный узел сети. Для IP-адресов, которые считаются дополнительными интерфейсами, "Uptime/Downtime" не измеряется.

Изменение настройки прокси

Узлы сети, обнаруженные разными прокси, не всегда рассматриваются как разные узлы 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 данного прокси, а не назначается самому прокси. Когда Zabbix проверяет уникальность IP-адресов во время обнаружения, он проверяет узлы сети, которые мониторятся родительской группой прокси. Узлы сети, мониторящиеся 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 мониторят пересекающиеся подсети.

Хотя такое поведение позволяет обнаружению работать в перекрывающихся диапазонах IP-адресов, используемых разными подсетями, изменение прокси, назначенного уже мониторящейся подсети, является более сложной задачей, поскольку изменения прокси должны последовательно применяться к обнаруженным узлам сети и к членству в родительской группе прокси, чтобы избежать дубликатов.

Например, шаги для замены прокси в правиле обнаружения:

1. отключить правило обнаружения
2. синхронизировать конфигурацию прокси
3. заменить прокси в правиле обнаружения
4. заменить прокси для всех узлов сети, обнаруженных этим правилом (убедитесь, что узлы сети в родительской группе прокси и все узлы сети, мониторящиеся отдельными прокси в этой группе, обновлены во избежание дубликатов)
5. включить правило обнаружения

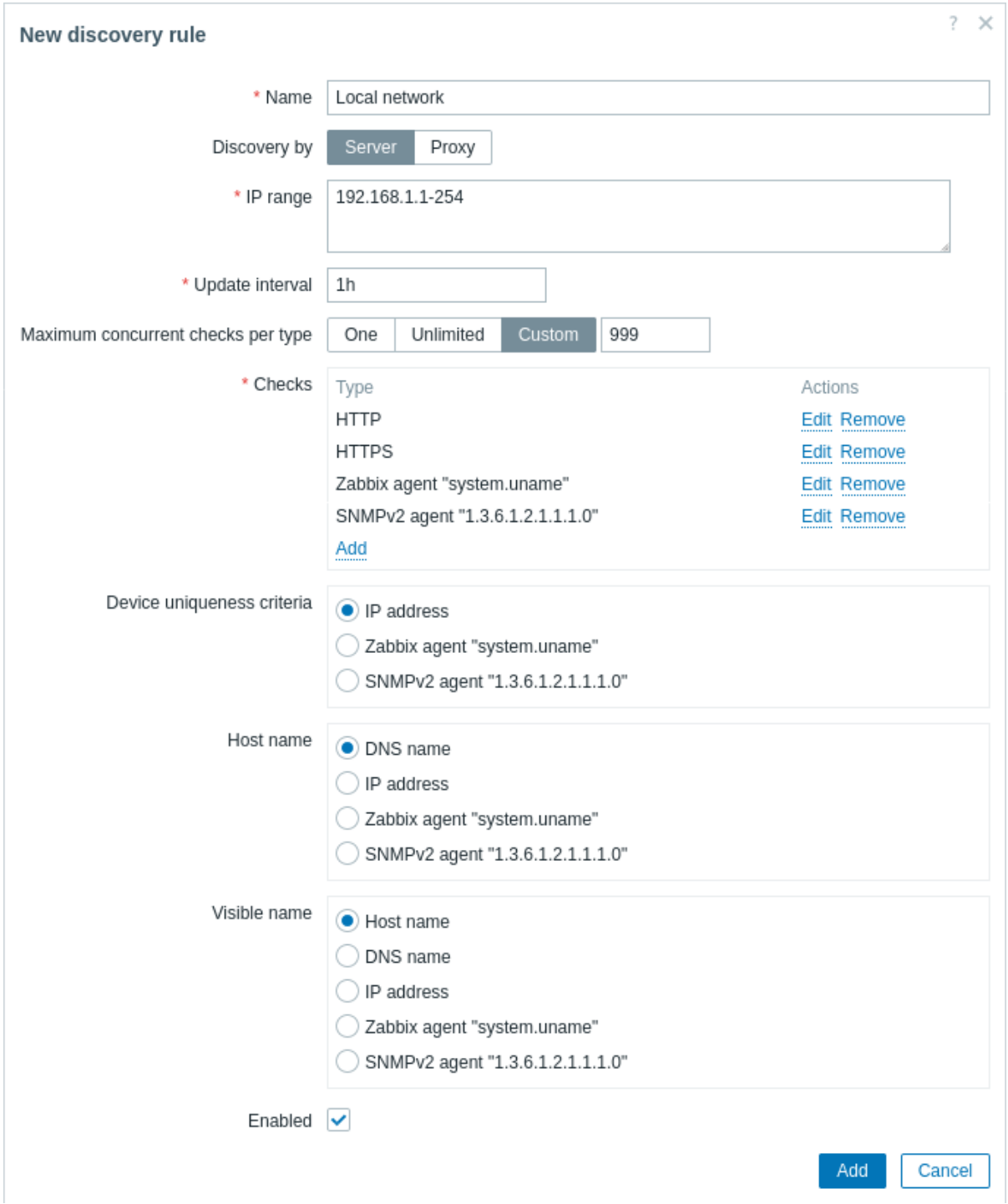
1 Настройка правила сетевого обнаружения

Обзор

Чтобы настроить правило обнаружения сети, используемое Zabbix для обнаружения узлов сети и сервисов:

- Перейдите в *Сбор данных* → *Обнаружение*
- Нажмите *Создать правило обнаружения* (или на имя правила, чтобы изменить существующее)
- Измените атрибуты правила обнаружения

Атрибуты правила



Все обязательные поля ввода отмечены красной звёздочкой.

Параметр	Описание
Имя	Уникальное имя правила. Например, ”Локальная сеть”.
Обнаружение через	Обнаружение выполняется: **Сервер** — сервером Zabbix **Прокси** — прокси Zabbix (выбирается в поле имени прокси)
Диапазон IP	Диапазон IP-адресов для обнаружения. Может иметь следующие форматы: Одиночный IP: 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.

Параметр	Описание
Интервал обновления	Этот параметр определяет, как часто Zabbix будет выполнять правило. Интервал отсчитывается после завершения выполнения предыдущего экземпляра обнаружения, поэтому пересечения не происходит. Поддерживаются суффиксы времени, например 30s, 1m, 2h, 1d. Поддерживаются пользовательские макросы. *Обратите внимание*, что если используется пользовательский макрос и его значение изменяется (например, 1w → 1h), следующая проверка будет выполнена в соответствии с предыдущим значением (в приведённом примере — ещё очень нескоро).
Максимальное количество одновременных проверок на тип	Установите максимальное количество потоков обнаружения (workers) на одну проверку сервиса для параллельной обработки проверок обнаружения: **Один** — один поток **Без ограничений** — неограниченное количество потоков (но не больше, чем указано в параметре StartDiscoverers) **Пользовательское** — задайте пользовательское количество потоков (0-999) Обратите внимание, что все правила обнаружения с асинхронными проверками сервисов SNMPv3 обрабатываются одним worker из-за особенностей реализации libsnmp, то есть увеличение количества workers не приведёт к увеличению скорости обнаружения.
Проверки	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, агент Zabbix, агент SNMPv1, агент SNMPv2, агент SNMPv3, ICMP ping. Обнаружение на основе протокола использует функциональность **net.tcp.service[]** для проверки каждого узла сети, кроме SNMP, который опрашивает SNMP OID. Агент Zabbix проверяется путём запроса элемента данных в незашифрованном режиме. Подробнее см. элементы данных агента. Параметр 'Ports' может иметь один из следующих форматов: Одиночный порт: 22 Диапазон портов: 22-45 Список: 22-45,55,60-70 Начиная с Zabbix 7.0. все проверки сервисов выполняются асинхронно, кроме проверок LDAP. Начиная с Zabbix 7.0 проверка HTTP/HTTPs выполняется через libcurl. Если сервер/прокси Zabbix скомпилирован без libcurl, тогда проверки HTTP будут работать как в предыдущих версиях (то есть как TCP-проверки), а проверки HTTPS работать не будут.
Критерии уникальности устройства	Критериями уникальности могут быть: **IP-адрес** — не обрабатывать несколько устройств с одним IP. Если устройство с таким же IP уже существует, оно будет считаться уже обнаруженным, и новый узел сети добавлен не будет. **<проверка обнаружения>** — либо проверка агента Zabbix, либо проверка агента SNMP. Обратите внимание, что критерий уникальности, используемый при обнаружении, не совпадает с идентификацией узла сети в системе при выполнении действий. Критерии уникальности при обнаружении определяют, являются ли два или более обнаруженных устройства одним и тем же устройством (или разными), тогда как в Zabbix критерием идентификации узла сети является только IP-адрес (см. Создание узла сети).
Имя узла сети	Задайте техническое имя создаваемого узла сети, используя: **DNS-имя** — DNS-имя (по умолчанию) **IP-адрес** — IP-адрес **<проверка обнаружения>** — полученное строковое значение проверки обнаружения (например, агент Zabbix, проверка агента SNMP) См. также: Именование узлов сети.
Видимое имя	Задайте видимое имя создаваемого узла сети, используя: **Имя узла сети** — техническое имя узла сети (по умолчанию) **DNS-имя** — DNS-имя **IP-адрес** — IP-адрес **<проверка обнаружения>** — полученное строковое значение проверки обнаружения (например, агент Zabbix, проверка агента SNMP) См. также: Именование узлов сети.
Включено	Если флажок установлен, правило активно и будет выполняться сервером Zabbix. Если флажок снят, правило неактивно. Оно не будет выполняться.

Превышение лимита файловых дескрипторов

В случае большого количества одновременных проверок возможно исчерпание лимита файловых дескрипторов для менеджера обнаружения.

Количество файловых дескрипторов, необходимых для обнаружения, равно числу процессов обнаружения * 1000. По умолчанию используется 5 процессов обнаружения, тогда как мягкий лимит системы составляет примерно 1024.

Если этот лимит приближается к исчерпанию, Zabbix уменьшит используемое по умолчанию количество одновременных проверок каждого типа для каждого процесса и запишет предупреждение в файл журнала. Однако если пользователь установил для *Maximum concurrent checks per type* значение выше, чем значение, вычисленное Zabbix, Zabbix будет использовать заданное пользователем значение для одного процесса.

Реальный сценарий

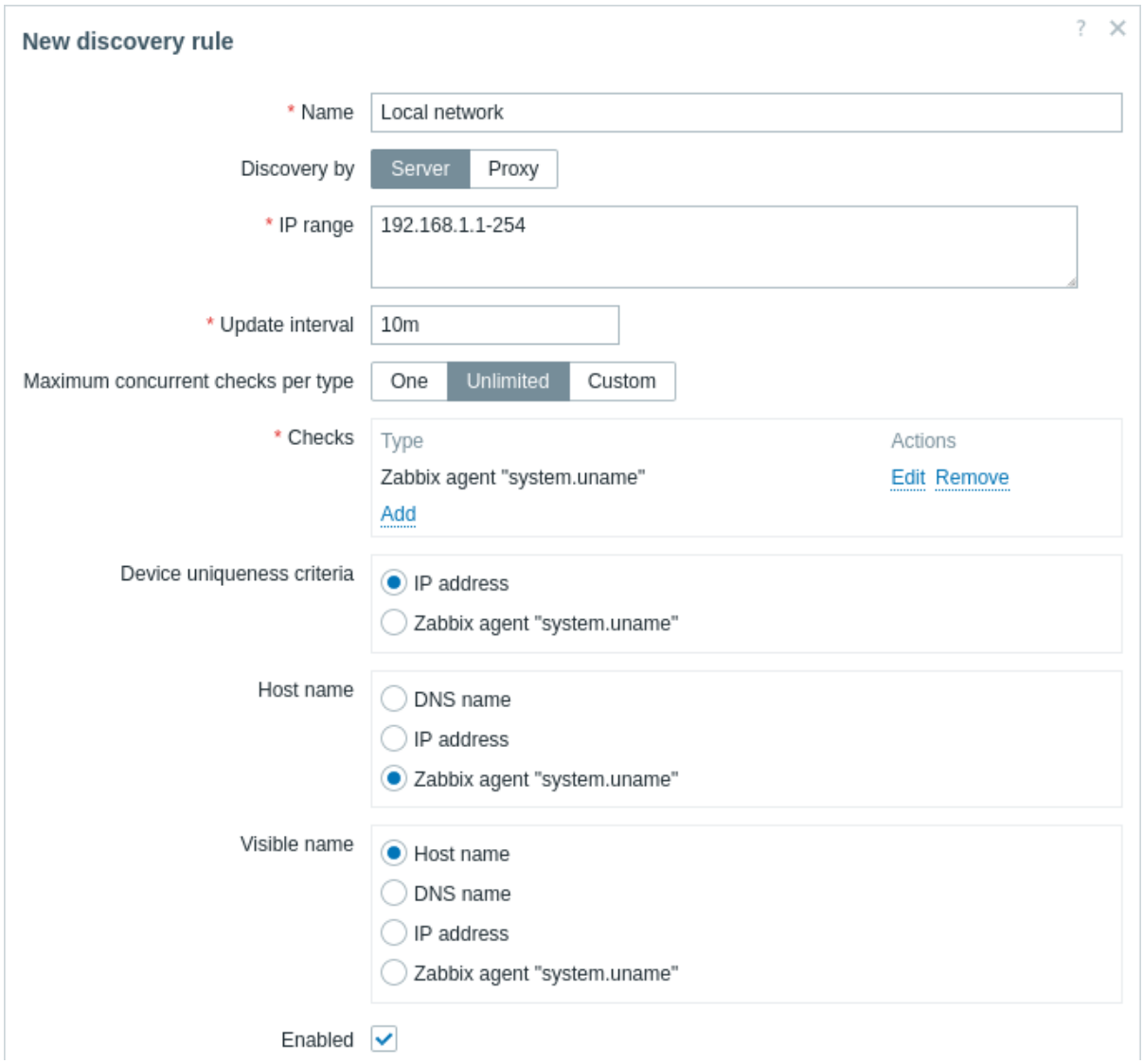
В этом примере мы хотим настроить обнаружение сети для локальной сети с диапазоном IP-адресов 192.168.1.1-192.168.1.254.

В нашем сценарии мы хотим:

- обнаруживать те узлы сети, на которых запущен агент Zabbix
- выполнять обнаружение каждые 10 минут
- добавлять узел сети в мониторинг, если время непрерывной работы узла сети превышает 1 час
- удалять узлы сети, если время простоя узла сети превышает 24 часа
- добавлять Linux-узлы сети в группу ”Linux servers”
- добавлять Windows-узлы сети в группу ”Windows servers”
- использовать шаблон *Linux* для Linux-узлов сети
- использовать шаблон *Windows* для Windows-узлов сети

Шаг 1

Определение правила обнаружения сети для нашего диапазона IP-адресов.



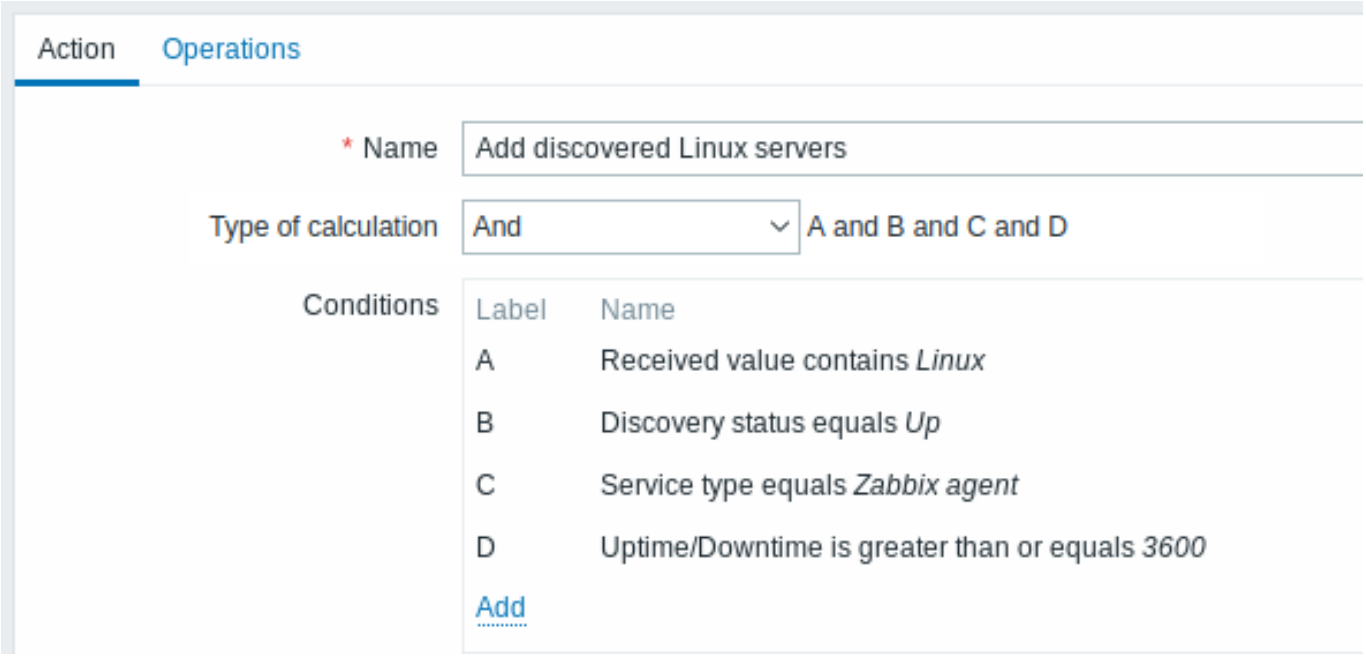
Zabbix попытается обнаружить узлы сети в диапазоне IP-адресов 192.168.1.1-192.168.1.254, подключаясь к агентам Zabbix и получая значение из ключа **system.uname**. Значение, полученное от агента, может использоваться для присвоения имен узлам сети, а также для применения различных действий к разным операционным системам. Например, связывать серверы Windows с шаблоном *Windows*, а серверы Linux — с шаблоном *Linux*.

Правило будет выполняться каждые 10 минут.

После добавления этого правила Zabbix автоматически начнет обнаружение и генерацию событий на основе обнаружения для дальнейшей обработки.

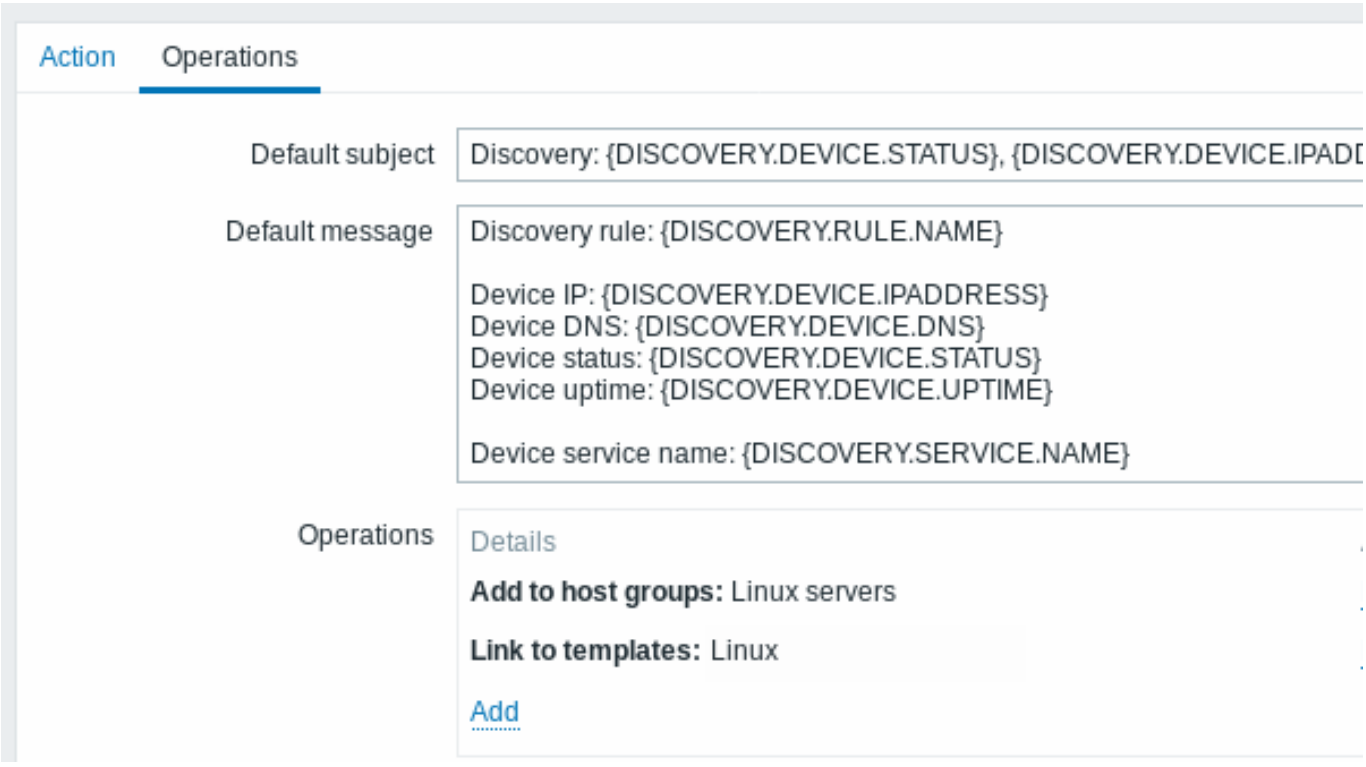
Шаг 2

Определение действия обнаружения для добавления обнаруженных Linux-серверов в соответствующую группу/шаблон.



Действие будет активировано, если:

- служба ”Zabbix агент” находится в состоянии ”up”
- значение system.uname (ключ Zabbix агента, который мы использовали при определении правила) содержит ”Linux”
- Uptime составляет 1 час (3600 секунд) или более

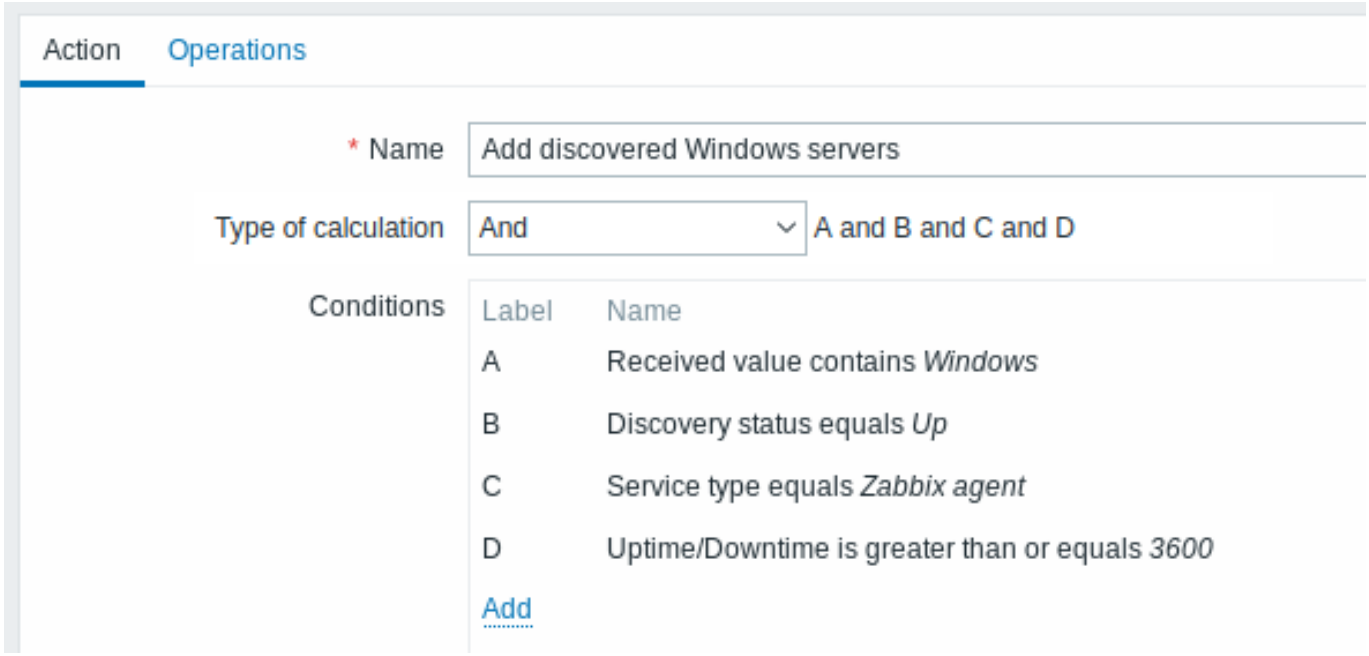


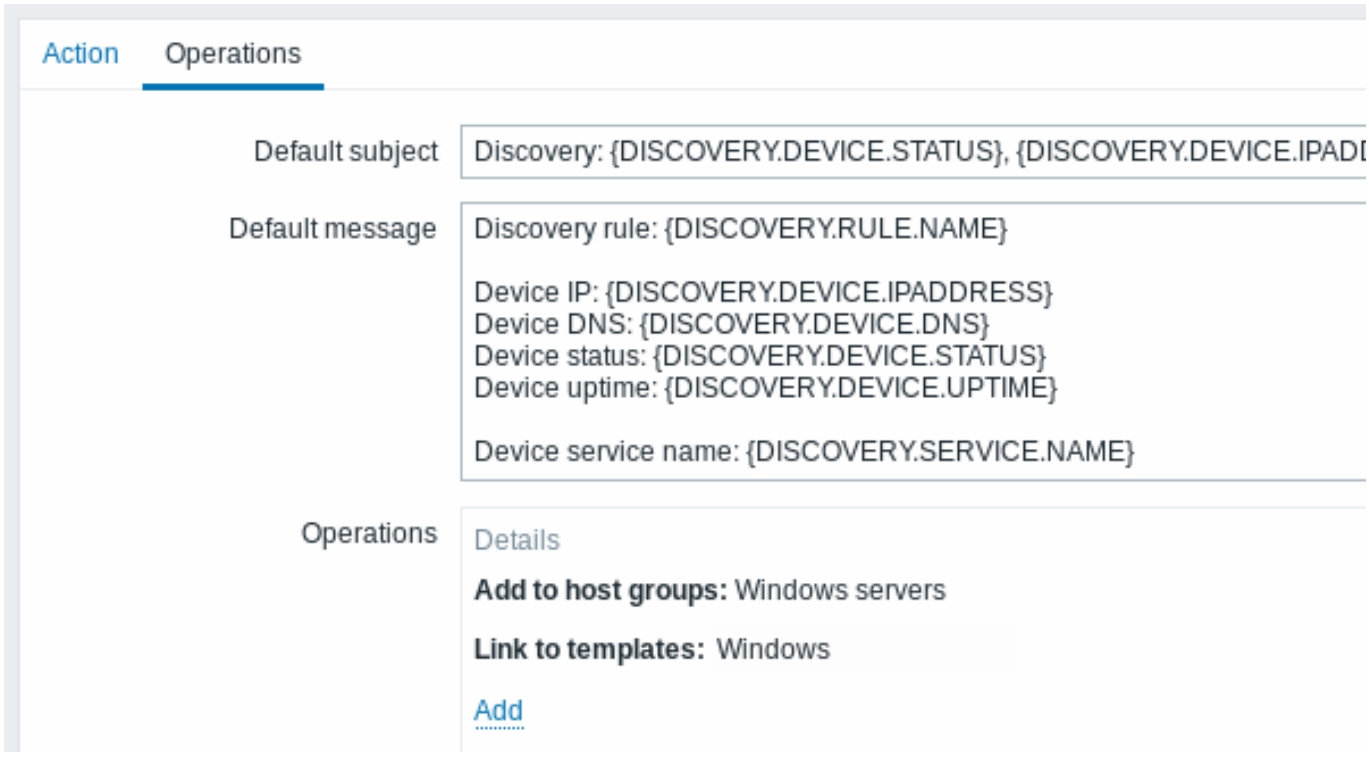
Действие выполнит следующие операции:

- добавить обнаруженный узел сети в группу ”Linux servers” (а также добавить узел сети, если он не был добавлен ранее)
- привязать узел сети к шаблону *Linux*. Zabbix автоматически начнет мониторинг узла сети, используя элементы данных и триггеры из шаблона ”Linux”.

Шаг 3

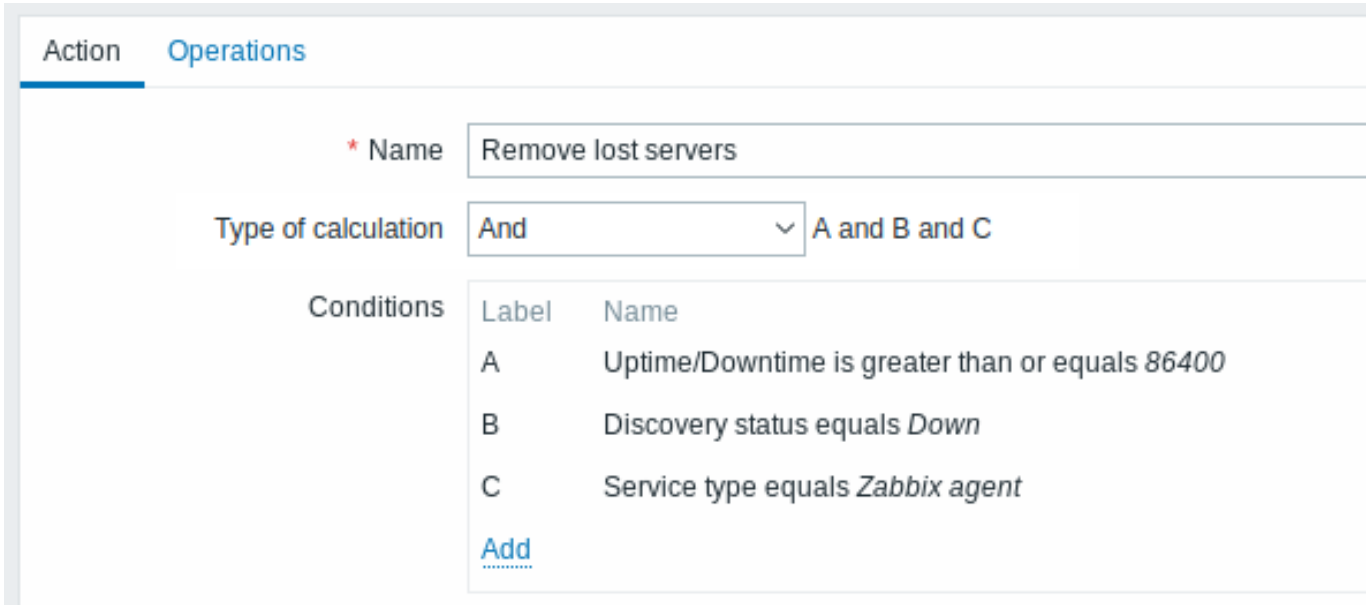
Определение действия обнаружения для добавления обнаруженных серверов Windows в соответствующую группу/шаблон.

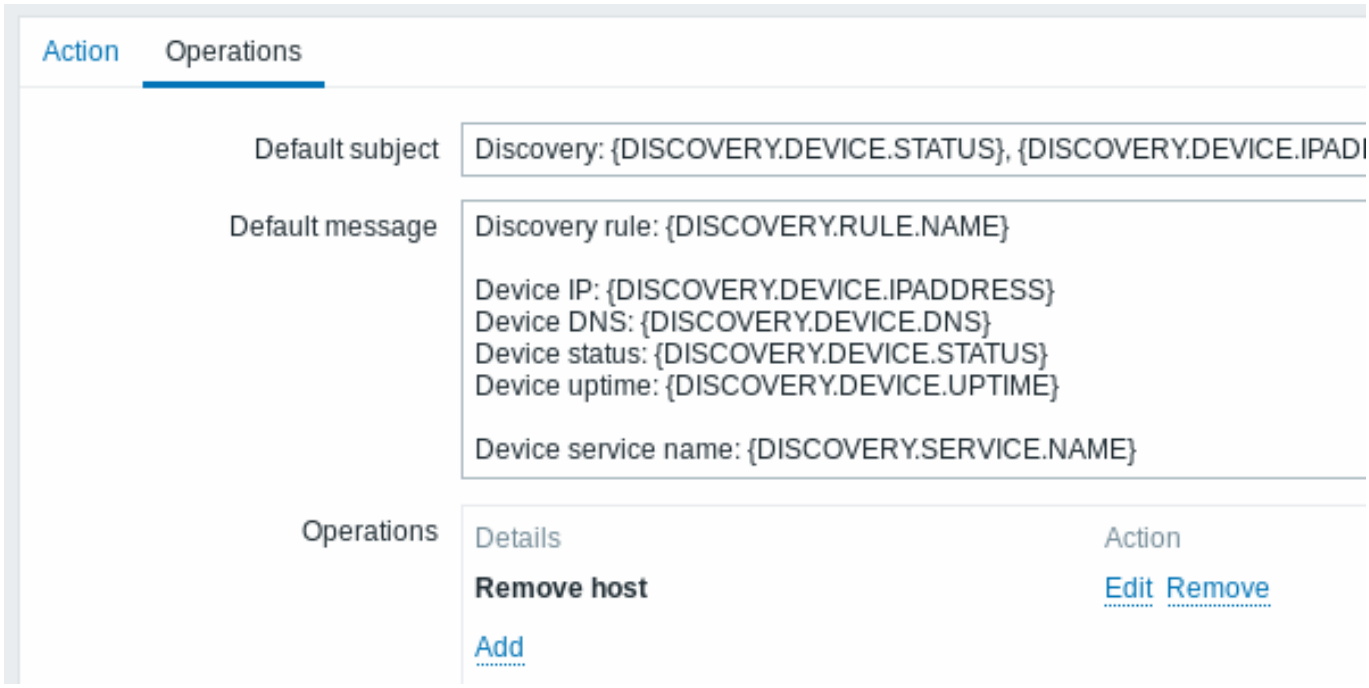




Шаг 4

Определение действия обнаружения для удаления потерянных серверов.





Сервер будет удалён, если служба ”Zabbix агент” находится в состоянии ’down’ более 24 часов (86400 секунд).

2 Авторегистрация активных агентов

Обзор

Можно разрешить авторегистрацию активного Zabbix агент, после чего сервер сможет начать их мониторинг. Таким образом новые узлы сети можно добавлять для мониторинга без их ручной настройки на сервере.

Авторегистрация может происходить, когда ранее неизвестный активный агент запрашивает проверки.

Эта возможность может быть очень полезна для автоматического мониторинга новых узлов Cloud. Как только у вас появляется новый узел в Cloud, Zabbix автоматически начнет сбор данных о производительности и доступности узла сети.

Авторегистрация активного агент также поддерживает мониторинг добавленных узлов сети с пассивными проверками. Когда активный агент запрашивает проверки, при условии, что в файле конфигурации определены параметры `ListenIP` или `ListenPort`, они также отправляются на сервер. Если указано несколько IP-адресов, на сервер отправляется первый из них.

При добавлении нового авторегистрируемого узла сети сервер использует полученные IP-адрес и порт для настройки агент.

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 авторегистрироваться с именем DNS в качестве интерфейса агент по умолчанию.

Авторегистрация выполняется повторно:

- если изменяется информация метаданных узла сети:
  - из-за изменения HostMetadata и перезапуска агент
  - из-за изменения значения, возвращаемого HostMetadataItem
- для созданных вручную узлов сети, у которых отсутствуют метаданные
- если узел сети вручную изменен так, чтобы мониториться другим Zabbix прокси
- если авторегистрация для того же узла сети поступает от нового Zabbix прокси

Интервал heartbeat авторегистрации активного агент для Zabbix сервер и Zabbix прокси составляет 120 секунд. Поэтому, если обнаруженный узел сети удален, авторегистрация будет выполнена повторно через 120 секунд.

Настройка

Укажите сервер

Убедитесь, что вы указали сервер Zabbix в файле конфигурации агента — `zabbix_agentd.conf`:

```
ServerActive=10.0.0.1
```

Если вы явно не определили *Hostname* в `zabbix_agentd.conf`, сервер будет использовать системное имя узла сети, на котором расположен агент, для именования узла сети. Системное имя узла сети в Linux можно получить, выполнив команду `hostname`.

Если *Hostname* определен в конфигурации агента Zabbix как список узлов сети, разделенный запятыми, узлы сети будут созданы для всех перечисленных имен узлов сети.

После внесения любых изменений в файл конфигурации перезапустите агент.

Действие для авторегистрации активного агента

Когда сервер получает запрос на авторегистрацию от агента, он вызывает действие. Для авторегистрации агента должно быть настроено действие с источником событий "Autoregistration".

Note:
Настройка обнаружения сети не требуется для того, чтобы активные агенты проходили авторегистрацию.

В веб-интерфейсе Zabbix перейдите в *Оповещения* → *Действия*, выберите *Действия авторегистрации* и нажмите *Создать действие*:

- На вкладке Действие задайте имя вашему действию
- При необходимости укажите условия. В условиях можно использовать поиск по подстроке или сопоставление по регулярному выражению для имени узла сети/метаданных узла сети. Если вы собираетесь использовать условие "Метаданные узла сети", см. следующий раздел.
- На вкладке Операции добавьте соответствующие операции, например: 'Добавить узел сети', 'Добавить в группу узлов сети' (например, *Discovered hosts*), 'Привязать шаблоны' и т. д.

Note:
Если узлы сети, которые будут проходить авторегистрацию, скорее всего будут поддерживаться только для активного мониторинга (например, узлы сети, доступ к которым с вашего сервера Zabbix ограничен межсетевым экраном), то можно создать специальный шаблон, например *Template_Linux-active*, для привязки.

Созданные узлы сети добавляются в группу *Discovered hosts* (по умолчанию; настраивается в *Администрирование* > *Общие* > *Прочее*). Если вы хотите, чтобы узлы сети добавлялись в другую группу, добавьте операцию *Удалить из группы узлов сети* (указав "Discovered hosts"), а также операцию *Добавить в группу узлов сети* (указав другую группу узлов сети), поскольку узел сети должен принадлежать к группе узлов сети.

Безопасная авторегистрация

Безопасный способ авторегистрации возможен при настройке аутентификации на основе PSK с использованием зашифрованных соединений.

Уровень шифрования настраивается глобально в *Administration* > *General* > *Autoregistration*. Можно выбрать отсутствие шифрования, TLS-шифрование с аутентификацией PSK или оба варианта одновременно (чтобы одни узлы сети могли регистрироваться без шифрования, а другие — через шифрование).

Аутентификация по PSK проверяется сервером Zabbix перед добавлением узла сети. Если проверка успешна, узел сети добавляется, а в *Connections from/to host* устанавливается только значение 'PSK' с identity/pre-shared key, совпадающими с глобальной настройкой авторегистрации.

Attention:
Чтобы обеспечить безопасность авторегистрации в установках, использующих прокси, следует включить шифрование между сервером Zabbix и прокси.

Использование DNS в качестве интерфейса по умолчанию

Параметры конфигурации HostInterface и HostInterfaceItem configuration parameters позволяют указать пользовательское значение для интерфейса узла сети во время авторегистрации.

В частности, они полезны, если узел сети должен быть авторегистрирован с DNS-именем в качестве интерфейса агента по умолчанию, а не с его IP-адресом. В этом случае DNS-имя должно быть указано или возвращено в качестве значения параметров HostInterface или HostInterfaceItem. Если значение одного из этих параметров изменится — например, с IP-адреса на DNS-имя или наоборот, — интерфейс по умолчанию авторегистрированного узла сети будет соответственно обновлен. Это обновление применяется к существующему узлу сети, а не путем создания нового. Чтобы отправить новое значение, агент необходимо перезапустить, чтобы он повторно инициировал процесс авторегистрации.

Note:
Если параметры HostInterface или HostInterfaceItem не настроены, вместо них используется параметр listen_dns. Это значение определяется путем выполнения обратного DNS-запроса для IP-адреса агента. Если обратный DNS не настроен должным образом или возвращает недопустимое имя, это может привести к некорректной или неудачной авторегистрации из-за недопустимого значения интерфейса.

Использование метаданных узла сети

Когда агент отправляет запрос авторегистрации на сервер, он передаёт своё 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 при необходимости использовать для передачи другой информации от агента на сервер.

Метаданные узла сети настраиваются в файле конфигурации агента — `zabbix_agentd.conf`. Существует 2 способа указать метаданные узла сети в файле конфигурации:

```
HostMetadata
HostMetadataItem
```

См. описание этих параметров 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 многобайтовые символы. Например, значение, содержащее только 3-байтовые символы, будет ограничено в общей сложности 21844 символами, а значение, содержащее только 4-байтовые символы, будет ограничено 16383 символами.

Attention:
Попытка авторегистрации выполняется каждый раз, когда активный агент отправляет на сервер запрос на обновление активных проверок. Задержка между запросами задаётся параметром RefreshActiveChecks агента. Первый запрос отправляется сразу после перезапуска агента.

Примеры

Авторегистрация по ОС с использованием HostMetadata

Предположим, вы хотите, чтобы узлы сети автоматически регистрировались на сервере Zabbix. В вашей сети используются активные агенты Zabbix (см. раздел "Configuration" выше). В вашей сети есть узлы сети Windows и Linux, и в вашем веб-интерфейсе Zabbix доступны шаблоны "Linux by Zabbix agent" и "Windows by Zabbix agent". Поэтому при регистрации узла сети вы хотите, чтобы к регистрируемому узлу сети применялся соответствующий шаблон Linux/Windows. По умолчанию при авторегистрации на сервер отправляется только имя узла сети, чего может быть недостаточно. Чтобы убедиться, что к узлу сети применяется правильный шаблон, следует использовать метаданные узла сети.

Настройка веб-интерфейса

Первое, что нужно сделать, — настроить веб-интерфейс. Создайте 2 действия. Первое действие:

- Имя: Авторегистрация Linux-узла сети
- Условия: Метаданные узла сети содержат *Linux*
- Операции: Привязать шаблоны: Linux by Zabbix агент

> **Note:**
> В этом случае можно пропустить операцию "Добавить узел сети". Привязка шаблона к узлу сети требует сначала добавить узел сети, поэтому сервер сделает это автоматически.

Второе действие:

- Имя: Авторегистрация Windows-узла сети
- Условия: Метаданные узла сети содержат *Windows*
- Операции: Привязать шаблоны: Windows by Zabbix агент

Настройка агента

Теперь необходимо настроить агенты. Добавьте следующую строку в файлы конфигурации агента:

```
HostMetadataItem=system.uname
```

Таким образом вы убедитесь, что метаданные узла сети будут содержать "Linux" или "Windows" в зависимости от узла сети, на котором запущен 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 агент после внесения любых изменений в файл конфигурации.

Использование HostMetadata для управления авторегистрацией и шаблонами

Шаг 1 — Защитите регистрацию с помощью HostMetadata

Использование метаданных узла сети позволяет обеспечить базовую защиту от регистрации нежелательных узлов сети.

Настройка веб-интерфейса

Создайте действие в веб-интерфейсе, используя некоторый трудноугадываемый секретный код, чтобы запретить нежелательным узлам сети регистрацию:

- Имя: Действие авторегистрации Linux
- Условия:
  - Тип вычисления: И
  - Условие (A): Метаданные узла сети содержат //Linux//
  - Условие (B): Метаданные узла сети содержат //21df83bf21bf0be663090bb8d4128558ab9b95fba66a6dbf834f8b91ae5e08ae//
- Операции:
  - Отправить сообщение пользователям: Admin через все способы оповещения
  - Добавить в группы узлов сети: Linux servers
  - Привязать шаблоны: Linux by Zabbix агент

Обратите внимание, что сам по себе этот метод не обеспечивает надежной защиты, поскольку данные передаются в открытом виде. Для немедленного применения изменений требуется перезагрузка кэша конфигурации.

Настройка агента

Добавьте следующую строку в файл конфигурации агента:

```
HostMetadata=Linux    21df83bf21bf0be663090bb8d4128558ab9b95fba66a6dbf834f8b91ae5e08ae
```

где "Linux" — это платформа, а остальная часть строки — секретный текст, который сложно угадать.

Не забудьте перезапустить агент после внесения любых изменений в файл конфигурации.

Шаг 2 — Добавление шаблона к зарегистрированному узлу сети

Можно добавить дополнительные шаблоны для уже зарегистрированного узла сети. В этом случае шаблон MySQL by Zabbix agent будет привязан только к тем узлам сети, у которых HostMetadata содержит токен `MySQL`.

Настройка веб-интерфейса

Обновите действие в веб-интерфейсе:

- Имя: Действие авторегистрации Linux
- Условия:
  - Тип вычисления: И

  - Условие (A): Метаданные узла сети содержат Linux
  - Условие (B): Метаданные узла сети содержат 21df83bf21bf0be663090bb8d4128558ab9b95fba66a6dbf834f8b91ae5e08ae
  - Условие (C): Метаданные узла сети содержат MySQL
- Операции:
  - Отправить сообщение пользователям: Admin через все способы оповещения
  - Добавить в группы узлов сети: Linux servers
  - Привязать шаблоны: Linux by Zabbix агент
  - Привязать шаблоны: MySQL by Zabbix Agent

Настройка агента

Обновите следующую строку в файле конфигурации агента:

```
HostMetadata=MySQL on Linux 21df83bf21bf0be663090bb8d4128558ab9b95fba66a6dbf834f8b91ae5e08ae
```

Не забудьте перезапустить агент после внесения любых изменений в файл конфигурации.

3 Низкоуровневое обнаружение (LLD)

Обзор Обнаружение низкого уровня (LLD) предоставляет способ автоматически создавать элементы данных, триггеры и графики для различных сущностей на узле сети. Например, Zabbix может автоматически начать мониторинг файловых систем или сетевых интерфейсов на вашей машине без необходимости вручную создавать элементы данных для каждой файловой системы или сетевого интерфейса. LLD также может создавать узлы сети, например, чтобы заполнять виртуальные машины, обнаруженные на гипервизоре, а также вложенные правила обнаружения, обеспечивая многоуровневое обнаружение. Кроме того, можно настроить Zabbix на автоматическое удаление ненужных сущностей на основе фактических результатов периодически выполняемого обнаружения.

Пользователь может определить собственные типы обнаружения при условии соблюдения определенного JSON-протокола.

Общая архитектура процесса обнаружения выглядит следующим образом.

Сначала пользователь создает правило обнаружения в *Сбор данных* > *Шаблоны*, в колонке *Обнаружение*. Правило обнаружения состоит из (1) элемента данных, который обнаруживает необходимые сущности (например, файловые системы или сетевые интерфейсы), и (2) прототипов элементов данных, триггеров и графиков, которые должны быть созданы на основе значения этого элемента данных.

Элемент данных, который обнаруживает необходимые сущности, похож на обычный элемент данных, встречающийся в других местах: сервер запрашивает у Zabbix агент значение этого элемента данных (или у другого источника, в зависимости от того, какой тип задан для элемента данных), агент отвечает текстовым значением. Разница в том, что значение, возвращаемое агентом, должно содержать список обнаруженных сущностей в формате JSON. Хотя детали этого формата важны только для разработчиков пользовательских проверок обнаружения, необходимо знать, что возвращаемое значение содержит список пар макрос → значение. Например, элемент данных ”net.if.discovery” может вернуть две пары: ”{#IFNAME}” → ”lo” и ”{#IFNAME}” → ”eth0”.

Эти макросы используются в именах, ключах и других полях прототипов, где затем подставляются полученные значения для создания реальных элементов данных, триггеров, графиков или даже узлов сети для каждой обнаруженной сущности. См. полный список параметров использования макросов LLD.

Когда сервер получает значение для элемента данных обнаружения, он просматривает пары макрос → значение и для каждой пары генерирует реальные элементы данных, триггеры и графики на основе их прототипов. В приведенном выше примере с ”net.if.discovery” сервер сгенерирует один набор элементов данных, триггеров и графиков для loopback-интерфейса ”lo” и другой набор для интерфейса ”eth0”.

Обратите внимание, что начиная с **Zabbix 4.2** формат JSON, возвращаемого правилами обнаружения низкого уровня, был изменен. Больше не ожидается, что JSON будет содержать объект ”data”. Теперь обнаружение низкого уровня принимает обычный JSON, содержащий массив, чтобы поддерживать новые возможности, такие как предобработка значений элементов данных и пользовательские пути к значениям макросов обнаружения низкого уровня в JSON-документе.

Встроенные ключи обнаружения были обновлены и теперь возвращают массив строк LLD в корне JSON-документа. Zabbix автоматически извлечет макрос и значение, если поле массива использует синтаксис {#MACRO} в качестве ключа. Любые новые встроенные проверки обнаружения будут использовать новый синтаксис без элементов ”data”. При обработке значения обнаружения низкого уровня сначала определяется корень (массив по `$.` или `$.data`).

Хотя элемент ”data” был удален из всех встроенных элементов данных, связанных с обнаружением, для обратной совместимости Zabbix по-прежнему будет принимать JSON-нотацию с элементом ”data”, хотя ее использование не

рекомендуется. Если JSON содержит объект только с одним элементом массива "data", то содержимое этого элемента будет автоматически извлечено с использованием JSONPath `$.data`. Обнаружение низкого уровня теперь принимает необязательные пользовательские макросы LLD с пользовательским путем, заданным в синтаксисе JSONPath.

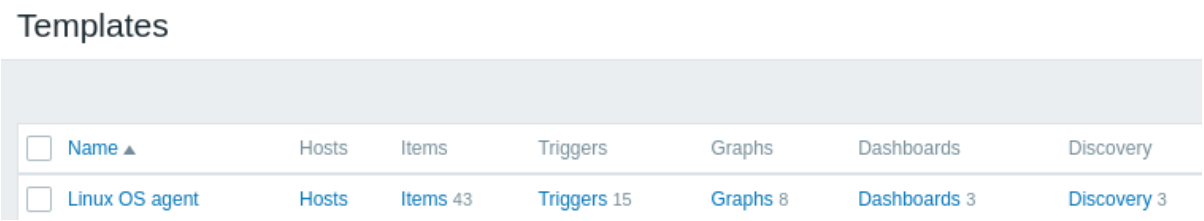
Warning:
В результате описанных выше изменений более новые агенты больше не смогут работать со старым сервером Zabbix.

См. также: Обнаруженные сущности

Настройка низкоуровневого обнаружения Мы проиллюстрируем низкоуровневое обнаружение на примере обнаружения файловых систем.

Чтобы настроить обнаружение, выполните следующие действия:

- Перейдите в: *Сбор данных* > *Шаблоны* или *Узлы сети*.
- Нажмите *Обнаружение* в строке соответствующего шаблона/узла сети.



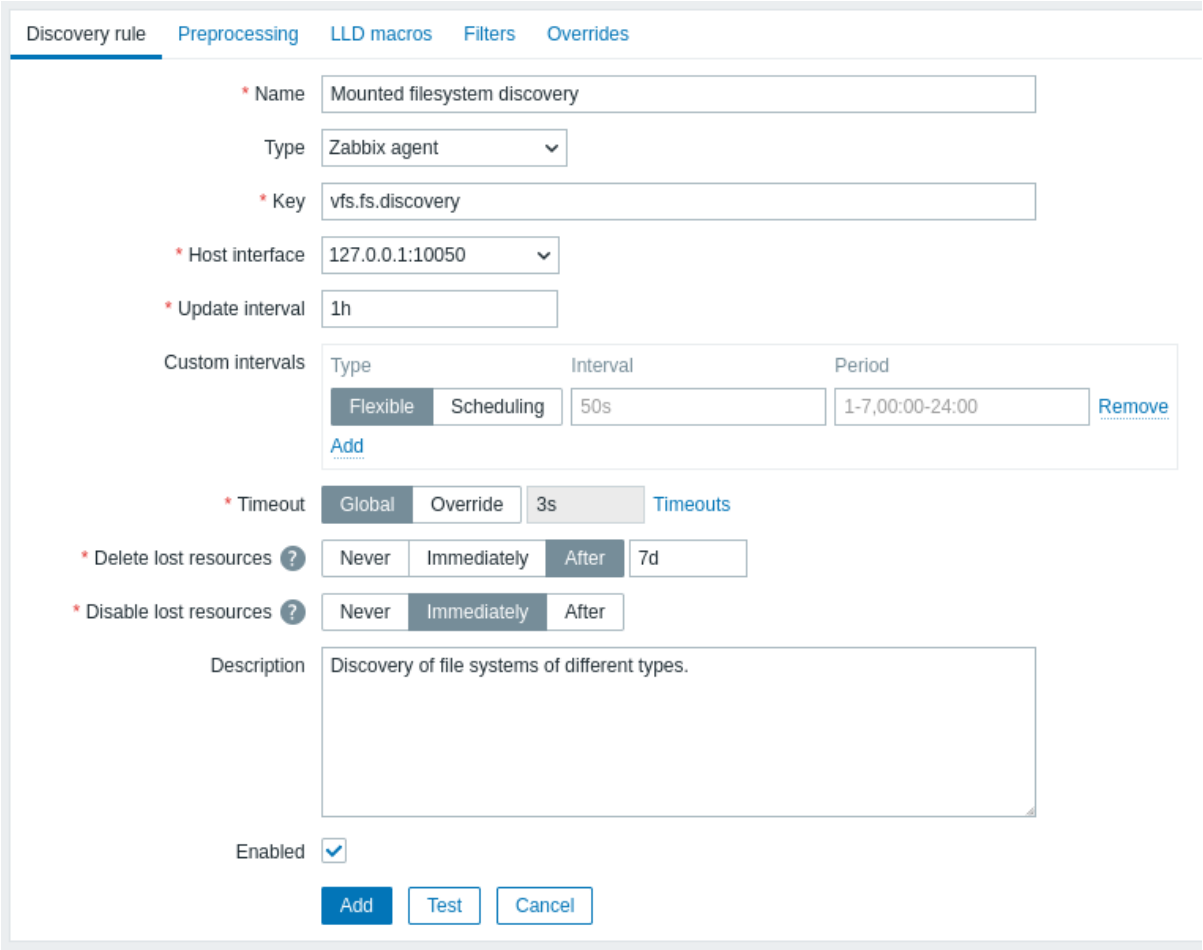
- Нажмите *Создать правило обнаружения* в правом верхнем углу экрана.
- Заполните форму правила обнаружения необходимыми данными.

Правило обнаружения

Форма правила обнаружения содержит пять вкладок, которые слева направо отражают поток данных в процессе обнаружения:

- *Правило обнаружения* — задает, прежде всего, встроенный элемент данных или пользовательский скрипт для получения данных обнаружения.
- *Предобработка* — применяет предобработку к обнаруженным данным.
- *LLD-макросы* — позволяет извлекать значения макросов для использования в обнаруженных элементах данных, триггерах и т. д.
- *Фильтры* — позволяет фильтровать обнаруженные значения.
- *Переопределения* — позволяет 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.

Parameter	Description
Name	Имя правила обнаружения.
Type	Тип проверки для выполнения обнаружения. В этом примере используется тип элемента данных *Zabbix агент*. 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 основной элемент данных в поле 'Master item'. Основной элемент данных должен существовать.
Key	Введите ключ элемента данных обнаружения (до 2048 символов). Например, можно использовать встроенный ключ элемента данных "vfs.fs.discovery", чтобы вернуть JSON-строку со списком файловых систем, имеющихся на компьютере, их типами и параметрами монтирования. Обратите внимание, что другим вариантом обнаружения файловых систем является использование результатов обнаружения по ключу агента "vfs.fs.get" (см. пример).
Update interval	Это поле задает, как часто Zabbix выполняет обнаружение. На начальном этапе, когда вы только настраиваете обнаружение файловых систем, можно установить небольшой интервал, но когда вы убедитесь, что все работает, можно установить 30 минут или больше, поскольку файловые системы обычно меняются не очень часто. Поддерживаются суффиксы времени, например 30s, 1m, 2h, 1d. Поддерживаются пользовательские макросы. *Примечание*: Интервал обновления может быть установлен в '0' только при наличии пользовательских интервалов с ненулевым значением. Если установлено '0' и существует пользовательский интервал (гибкий или по расписанию) с ненулевым значением, элемент данных будет опрашиваться в течение действия пользовательского интервала. Новые правила обнаружения будут проверены в течение 60 секунд после создания, если только для них не задано расписание или гибкий интервал обновления, а *Update interval* не установлен в 0. *Обратите внимание*, что для существующего правила обнаружения обнаружение можно выполнить немедленно, нажав кнопку *Execute now* button.

Parameter	Description
Custom intervals	Можно создать пользовательские правила для проверки элемента данных: **Flexible** — создать исключение для *Update interval* (интервал с другой частотой) **Scheduling** — создать пользовательское расписание опроса. Подробную информацию см. в разделе Пользовательские интервалы.
Timeout	Задайте тайм-аут проверки обнаружения. Выберите вариант тайм-аута: **Global** — используется тайм-аут прокси/глобальный тайм-аут (отображается в неактивном поле *Timeout*); **Override** — используется пользовательский тайм-аут (задается в поле *Timeout*; допустимый диапазон: 1 - 600s). Поддерживаются суффиксы времени, например 30s, 1m, а также пользовательские макросы. Нажатие на ссылку *Timeouts* позволяет настроить тайм-ауты прокси или глобальные тайм-ауты (если прокси не используется). Обратите внимание, что ссылка *Timeouts* видна только пользователям типа *Super admin* с правами на разделы веб-интерфейса *Administration* > *General* или *Administration* > *Proxies*.
Delete lost resources	Укажите, через какое время обнаруженная сущность будет удалена после того, как ее статус обнаружения станет ”Not discovered anymore”: *Never* — не удалять; *Immediately* — удалить немедленно; *After* — удалить по истечении указанного периода времени. Значение должно быть больше значения *Disable lost resources*. Поддерживаются суффиксы времени, например 2h, 1d. Поддерживаются пользовательские макросы. *Примечание*: Использовать ”Immediately” не рекомендуется, поскольку даже ошибочное изменение фильтра может привести к удалению сущности вместе со всеми историческими данными. Обратите внимание, что ресурсы, отключенные вручную, не будут удалены низкоуровневым обнаружением.
Disable lost resources	Укажите, через какое время обнаруженная сущность будет отключена после того, как ее статус обнаружения станет ”Not discovered anymore”: *Never* — не отключать; *Immediately* — отключить немедленно; *After* — отключить по истечении указанного периода времени. Значение должно быть больше интервала обновления правила обнаружения. Обратите внимание, что автоматически отключенные ресурсы будут снова включены, если будут повторно обнаружены низкоуровневым обнаружением. Ресурсы, отключенные вручную, не будут снова включены при повторном обнаружении. Это поле не отображается, если *Delete lost resources* установлено в ”Immediately”. Поддерживаются суффиксы времени, например 2h, 1d. Поддерживаются пользовательские макросы.
Description	Введите описание.
Enabled	Если установлен флажок, правило будет обрабатываться.

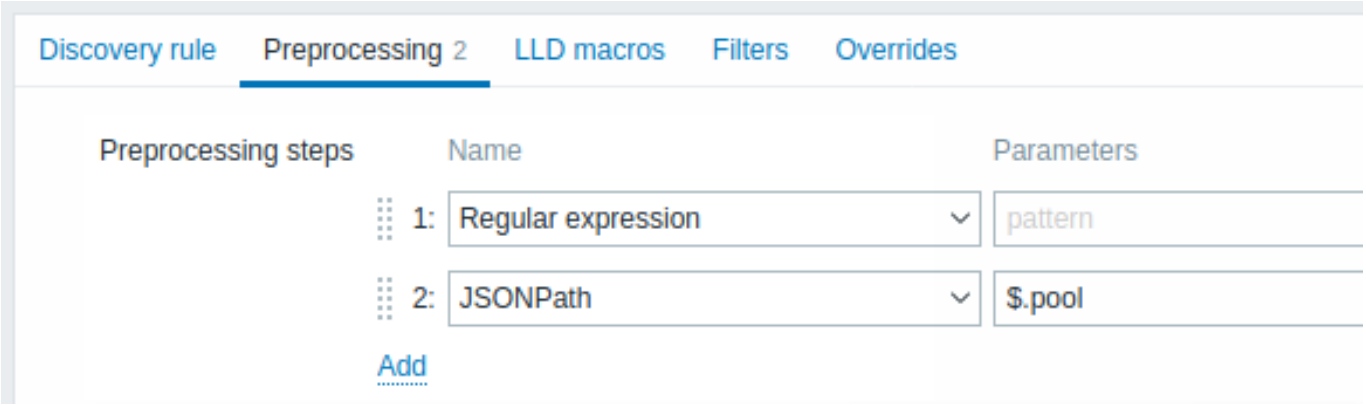
Note:
История правила обнаружения не сохраняется.

Предобработка

Вкладка **Предобработка** позволяет определить правила преобразования, применяемые к результату обнаружения. На этом шаге возможно одно или несколько преобразований. Преобразования выполняются в том порядке, в котором они определены. Вся предобработка выполняется сервером Zabbix.

См. также:

- Подробности предобработки
- Тестирование предобработки



Type		
	Transformation	Описание
Text		
	Regular expression	Сопоставить полученное значение с регулярным выражением <pattern> и заменить значение извлечённым <output>. Регулярное выражение поддерживает извлечение максимум 10 захваченных групп с помощью последовательности \N. Параметры: **pattern** - регулярное выражение **output** - шаблон форматирования вывода. Управляющая последовательность \N (где N=1...9) заменяется N-й совпавшей группой. Управляющая последовательность \0 заменяется совпавшим текстом. Если установить флажок *Custom on fail*, можно указать пользовательские варианты обработки ошибок: отбросить значение, установить указанное значение или установить указанное сообщение об ошибке.
	Replace	Найти строку поиска и заменить её другой строкой (или ничем). Все вхождения строки поиска будут заменены. Параметры: **search string** - строка для поиска и замены, с учётом регистра (обязательно) **replacement** - строка, на которую будет заменена строка поиска. Строка замены также может быть пустой, что фактически позволяет удалить найденную строку поиска. Можно использовать управляющие последовательности для поиска или замены переводов строк, возврата каретки, табуляции и пробелов: "\n \r \t \s"; обратную косую черту можно экранировать как "\\"; управляющие последовательности можно экранировать как "\\n". Экранирование переводов строк, возврата каретки и табуляции выполняется автоматически во время обнаружения низкого уровня.
Structured data		
	JSONPath	Извлечь значение или фрагмент из данных JSON с помощью функциональности JSONPath. Если установить флажок *Custom on fail*, можно указать пользовательские варианты обработки ошибок: отбросить значение, установить указанное значение или установить указанное сообщение об ошибке.

	XML XPath	Извлечь значение или фрагмент из данных XML с помощью функциональности XPath. Чтобы этот вариант работал, сервер Zabbix должен быть скомпилирован с поддержкой libxml.
		Примеры:
		`number(/document/item/value)` извлечёт 10 из `<document><item><value>10</value></item></document>`
		`number(/document/item/@attribute)` извлечёт 10 из `<document><item attribute="10"></item></document>`
		`/document/item` извлечёт `<item><value>10</value></item>` из `<document><item><value>10</value></item></document>`
		Обратите внимание, что пространства имён не поддерживаются.
		Если установить флажок *Custom on fail*, можно указать пользовательские варианты обработки ошибок: отбросить значение, установить указанное значение или установить указанное сообщение об ошибке.

	CSV to JSON	Преобразовать данные CSV-файла в формат JSON. Подробнее см.: Предобработка CSV в JSON.

Type	
XML to JSON	Преобразовать данные в формате XML в JSON. Подробнее см.: Правила сериализации. Если установить флажок *Custom on fail*, можно указать пользовательские варианты обработки ошибок: отбросить значение, установить указанное значение или установить указанное сообщение об ошибке.
SNMP	
SNMP walk value	Извлечь значение по указанному имени OID/MIB и применить параметры форматирования: **Unchanged** - вернуть Hex-STRING как неэкранированную шестнадцатеричную строку (*обратите внимание*, что подсказки отображения всё равно применяются); **UTF-8 from Hex-STRING** - преобразовать Hex-STRING в строку UTF-8; **MAC from Hex-STRING** - преобразовать Hex-STRING в строку MAC-адреса (в которой `' '` будет заменён на `':'`); **Integer from BITS** - преобразовать первые 8 байт битовой строки, представленной как последовательность шестнадцатеричных символов (например, ”1A 2B 3C 4D”), в 64-битное беззнаковое целое число. В битовых строках длиннее 8 байт последующие байты будут проигнорированы. Если установить флажок *Custom on fail*, можно указать пользовательские варианты обработки ошибок: отбросить значение, установить указанное значение или установить указанное сообщение об ошибке.
SNMP walk to JSON	Преобразовать значения SNMP в JSON. Укажите имя поля в JSON и соответствующий путь SNMP OID. Значения полей будут заполняться значениями из указанного пути SNMP OID. Этот шаг предобработки можно использовать для обнаружения SNMP OID. Доступны параметры форматирования значений, аналогичные шагу *SNMP walk value*. Если установить флажок *Custom on fail*, можно указать пользовательские варианты обработки ошибок: отбросить значение, установить указанное значение или установить указанное сообщение об ошибке.
SNMP get value	Применить параметры форматирования к значению, полученному через SNMP get: **UTF-8 from Hex-STRING** - преобразовать Hex-STRING в строку UTF-8; **MAC from Hex-STRING** - преобразовать Hex-STRING в строку MAC-адреса (в которой `' '` будет заменён на `':'`); **Integer from BITS** - преобразовать первые 8 байт битовой строки, представленной как последовательность шестнадцатеричных символов (например, ”1A 2B 3C 4D”), в 64-битное беззнаковое целое число. В битовых строках длиннее 8 байт последующие байты будут проигнорированы. Если установить флажок *Custom on fail*, можно указать пользовательские варианты обработки ошибок: отбросить значение, установить указанное значение или установить указанное сообщение об ошибке.
Custom scripts	
JavaScript	Введите код JavaScript в модальном редакторе, который открывается при щелчке в поле параметра или по значку карандаша рядом с ним. Обратите внимание, что доступная длина JavaScript зависит от используемой базы данных. Подробнее см.: Предобработка Javascript
Validation	
Does not match regular expression	Укажите регулярное выражение, которому значение не должно соответствовать. Например: `Error:(.*?)\.` Если установить флажок *Custom on fail*, можно указать пользовательские варианты обработки ошибок: отбросить значение, установить указанное значение или установить указанное сообщение об ошибке.

Type	
Check for error in JSON	Проверить наличие сообщения об ошибке уровня приложения, расположенного по JSONPath. Если проверка успешна и сообщение не пустое, обработка останавливается; в противном случае обработка продолжается со значением, которое было до этого шага предобработки. Обратите внимание, что такие ошибки внешних сервисов показываются пользователю как есть, без добавления информации о шаге предобработки. Например: `$.errors`. Если получен JSON вида `{"errors":"e1"}`, следующий шаг предобработки выполнен не будет. Если установить флажок *Custom on fail*, можно указать пользовательские варианты обработки ошибок: отбросить значение, установить указанное значение или установить указанное сообщение об ошибке.
Check for error in XML	Проверить наличие сообщения об ошибке уровня приложения, расположенного по XPath. Если проверка успешна и сообщение не пустое, обработка останавливается; в противном случае обработка продолжается со значением, которое было до этого шага предобработки. Обратите внимание, что такие ошибки внешних сервисов показываются пользователю как есть, без добавления информации о шаге предобработки. При невозможности разобрать некорректный XML ошибка выведена не будет. Если установить флажок *Custom on fail*, можно указать пользовательские варианты обработки ошибок: отбросить значение, установить указанное значение или установить указанное сообщение об ошибке.
Matches regular expression	Укажите регулярное выражение, которому значение должно соответствовать. Если установить флажок *Custom on fail*, можно указать пользовательские варианты обработки ошибок: отбросить значение, установить указанное значение или установить указанное сообщение об ошибке.
Throttling	
Discard unchanged with heartbeat	Отбросить значение, если оно не изменялось в течение заданного периода времени (в секундах). Для указания секунд поддерживаются положительные целые значения (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 вариант ограничения частоты. Например: `1m`. Если в это правило дважды передан одинаковый текст в течение 60 секунд, он будет отброшен. *Примечание*: изменение прототипов элементов данных не сбрасывает ограничение частоты. Ограничение частоты сбрасывается только при изменении шагов предобработки.
Prometheus	
Prometheus to JSON	Преобразовать требуемые метрики Prometheus в JSON. Подробнее см. Проверки Prometheus.

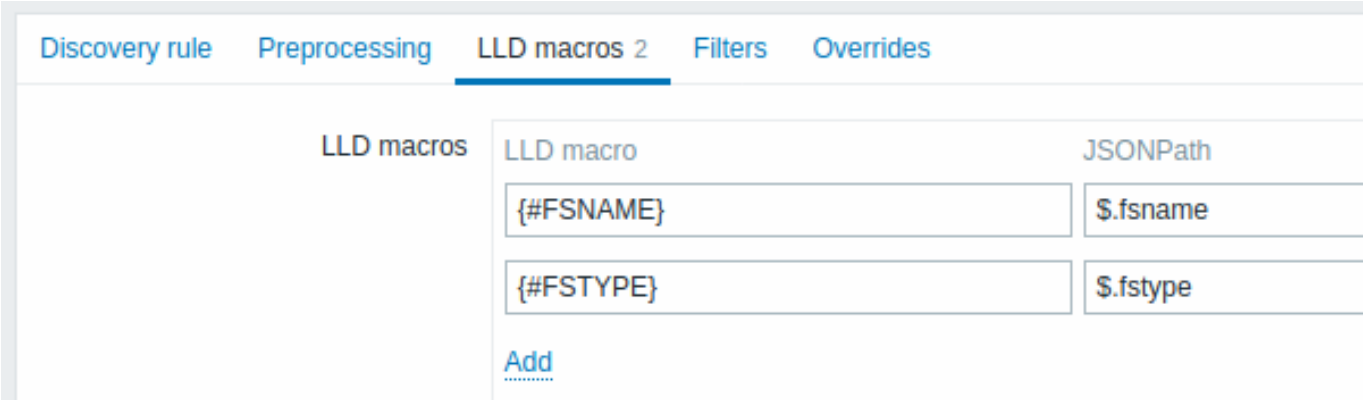
Обратите внимание, что если правило обнаружения было применено к узлу сети через шаблон, содержимое этой вкладки доступно только для чтения.

Пользовательские макросы

Вкладка **Макросы LLD** позволяет указать пользовательские макросы низкоуровневого обнаружения.

Пользовательские макросы полезны в случаях, когда возвращаемый JSON уже не содержит определённых требуемых макросов. Так, например:

- Встроенный ключ `vfs.fs.discovery` для обнаружения файловых систем возвращает JSON с некоторыми предопределёнными макросами LLD, такими как {#FSNAME}, {#FSTYPE}. Эти макросы можно использовать непосредственно в прототипах элементов данных и триггеров (см. последующие разделы страницы); определять пользовательские макросы не требуется;
- Элемент данных агента `vfs.fs.get` также возвращает JSON с данными файловых систем, но без каких-либо предопределённых макросов LLD. В этом случае вы можете определить макросы самостоятельно и сопоставить их со значениями в JSON с помощью JSONPath:

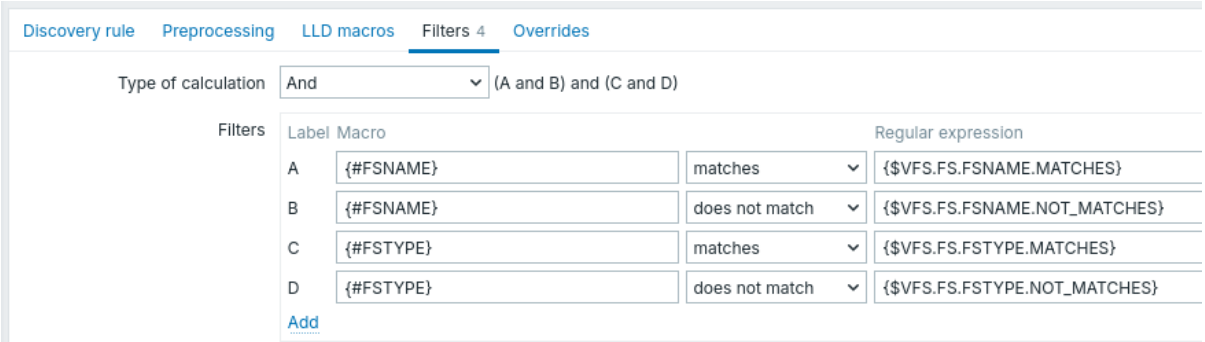


Извлечённые значения можно использовать в обнаруженных элементах данных, триггерах и т. д. Обратите внимание, что значения будут извлекаться из результата обнаружения и всех шагов предобработки, выполненных к этому моменту.

Параметр	Описание
Макрос LLD	Имя макроса низкоуровневого обнаружения с использованием следующего синтаксиса: {#MACRO}.
JSONPath	Путь, используемый для извлечения значения макроса LLD из строки LLD с использованием синтаксиса JSONPath. Значения, извлечённые из возвращаемого JSON, используются для замены макросов LLD в полях прототипов элементов данных, триггеров и т. д. JSONPath можно указывать с использованием точечной нотации или нотации в квадратных скобках. Нотацию в квадратных скобках следует использовать в случае наличия специальных символов и Unicode, например `$['unicode + special chars #1']['unicode + special chars #2']`. Например, `$.foo` извлечёт "bar" и "baz" из этого JSON: `[{"foo":"bar"}, {"foo":"baz"}]` Обратите внимание, что `$.foo` также извлечёт "bar" и "baz" из этого JSON: `{"data":[{"foo":"bar"}, {"foo":"baz"}]}`, поскольку одиночный объект "data" обрабатывается автоматически (для обратной совместимости с реализацией низкоуровневого обнаружения в версиях Zabbix до 4.2).

Фильтр

Фильтр можно использовать для создания реальных элементов данных, триггеров и графиков только для сущностей, соответствующих критериям. Вкладка **Фильтры** содержит определения фильтров правила обнаружения, позволяющие фильтровать значения обнаружения:



Параметр	Описание
Тип вычисления	Доступны следующие варианты вычисления фильтров: **И** - должны пройти все фильтры; **Или** - достаточно, чтобы прошел один фильтр; **И/Или** - использует *И* для разных имен макросов и *Или* для одинаковых имен макросов; **Пользовательское выражение** - предоставляет возможность задать пользовательское вычисление фильтров. Формула должна включать все фильтры в списке. Ограничено 255 символами.

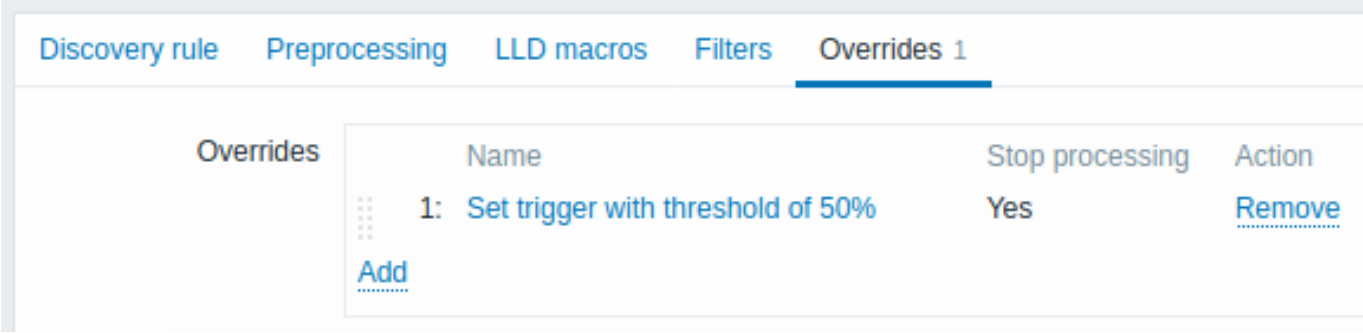
Параметр	Описание
Фильтры	Доступны следующие операторы условий фильтра: *совпадает*, *не совпадает*, *существует*, *не существует*. Операторы *совпадает* и *не совпадает* ожидают Perl Compatible Regular Expression (PCRE). Например, если вас интересуют только файловые системы C:, D: и E:, вы можете указать {#FSNAME} в поле "Макрос", а регулярное выражение "^C|^D|^E" — в поле "Регулярное выражение". Фильтрация также возможна по типам файловых систем с использованием макроса {#FSTYPE} (например, "^ext|^reiserfs") и по типам дисков с использованием макроса {#FSDRIVETYPE} (например, "fixed"; поддерживается только агентом Windows). В поле "Регулярное выражение" можно ввести регулярное выражение или сослаться на глобальное регулярное выражение. Чтобы проверить регулярное выражение, можно использовать "grep -E", например: `for f in ext2 nfs reiserfs smbfs; do echo $f | grep -E '^ext|^reiserfs' || echo "SKIP: $f"; done` Операторы *существует* и *не существует* позволяют фильтровать сущности на основе наличия или отсутствия указанного макроса LLD в ответе. Обратите внимание, что если макрос из фильтра отсутствует в ответе, найденная сущность будет проигнорирована, если только для этого макроса не указано условие "не существует". Будет показано предупреждение, если отсутствие макроса влияет на результат выражения. Например, если {#B} отсутствует в: {#A} matches 1 and {#B} matches 2 - будет выдано предупреждение {#A} matches 1 or {#B} matches 2 - предупреждения не будет

Warning:
Ошибка или опечатка в регулярном выражении, используемом в правиле LLD (например, некорректное регулярное выражение "File systems for discovery"), может привести к удалению тысяч элементов конфигурации, исторических значений и событий для множества узлов сети.

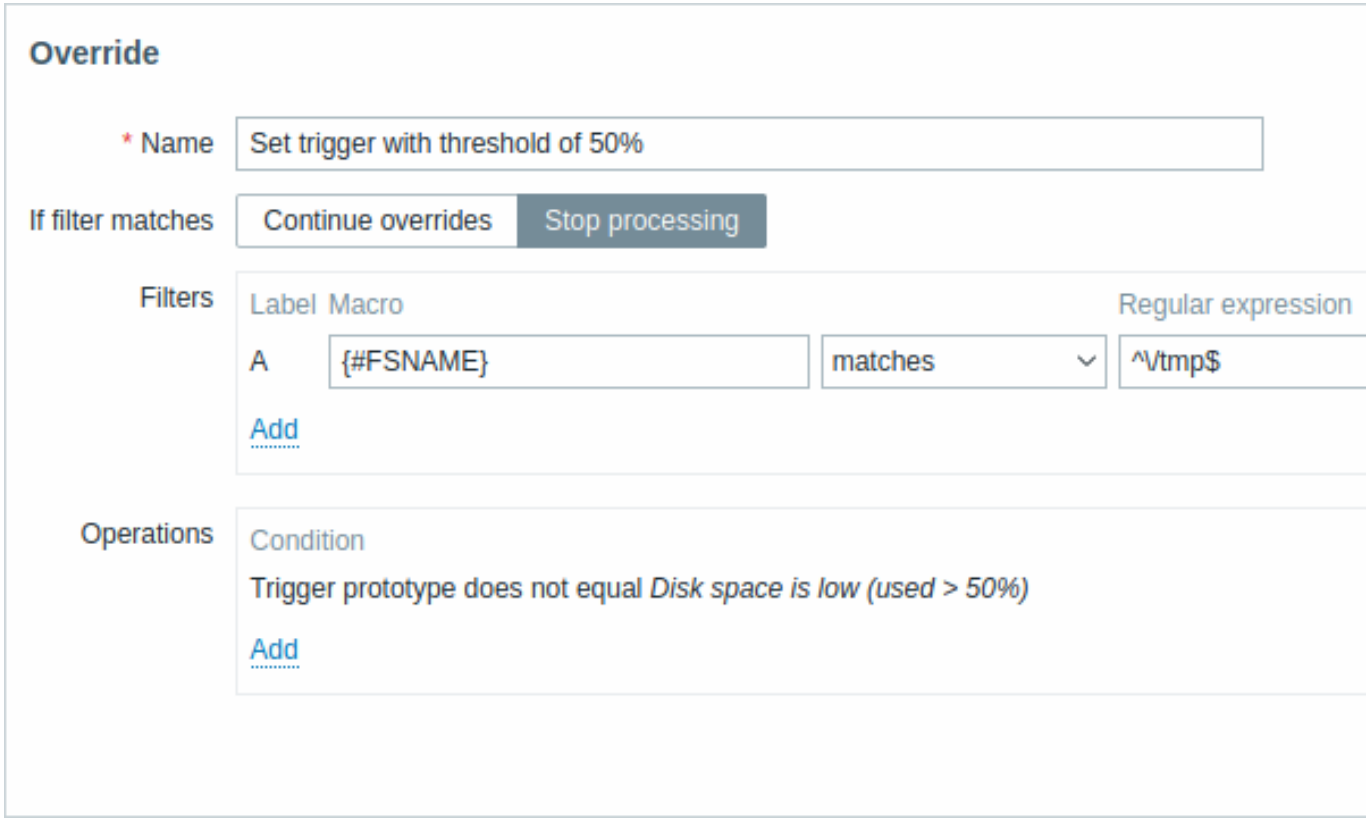
Attention:
База данных Zabbix в MySQL должна быть создана с учетом регистра, чтобы имена файловых систем, различающиеся только регистром, обнаруживались корректно.

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 виде переупорядочиваемого списка с перетаскиванием и выполняются в том порядке, в котором они определены. Чтобы настроить параметры нового переопределения, нажмите Add в блоке *Переопределения*. Чтобы изменить существующее переопределение, нажмите на его имя. Откроется всплывающее окно, в котором можно изменить параметры правила переопределения.

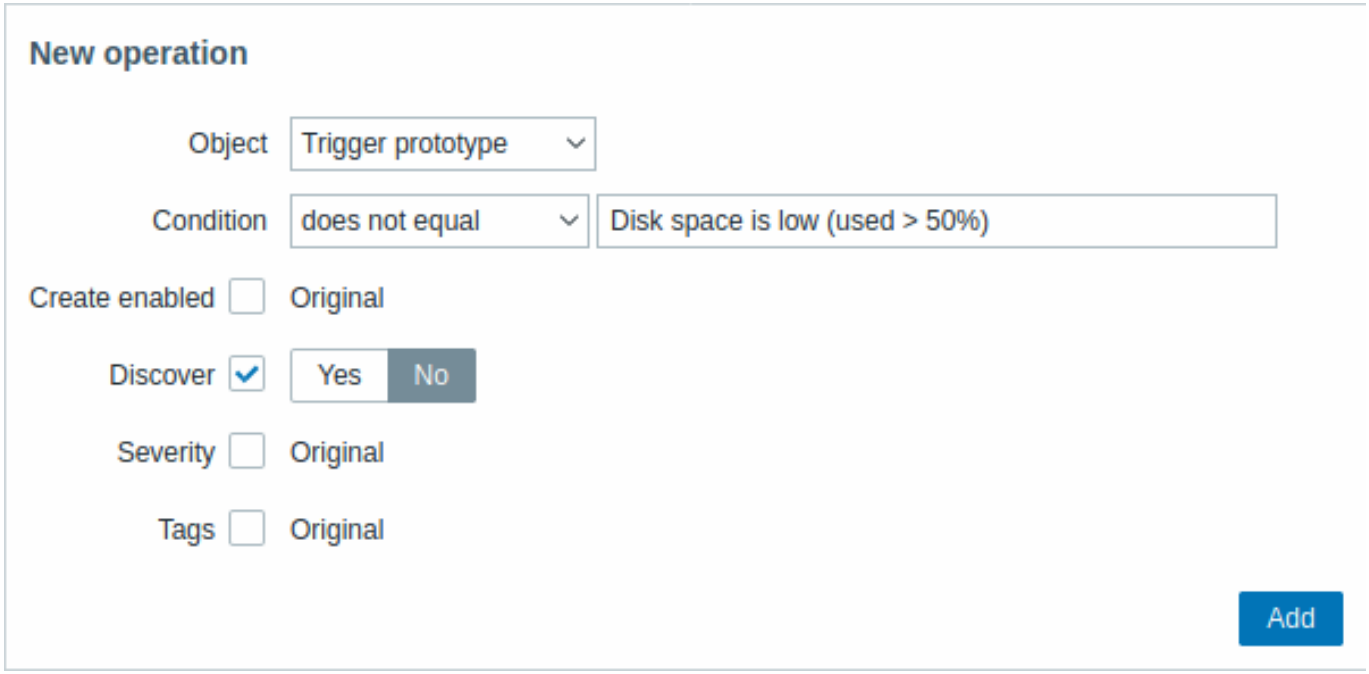


Все обязательные параметры отмечены красными звездочками.

Parameter	Description
Name	Уникальное (в пределах правила LLD) имя переопределения.
If filter matches	Определяет, следует ли обрабатывать следующие переопределения при выполнении условий фильтра: **Continue overrides** — последующие переопределения будут обработаны. **Stop processing** — операции из предыдущих (если есть) и данного переопределения будут выполнены, последующие переопределения будут проигнорированы для совпавших строк LLD.
Filters	Определяет, к каким обнаруженным сущностям должно применяться переопределение. Фильтры переопределения обрабатываются после фильтров правила обнаружения и имеют ту же функциональность.
Operations	Операции переопределения отображаются со следующими сведениями: **Condition** — тип объекта и условие, которому должно соответствовать имя объекта; например: Прототип триггера не равно *Disk space is low (used > 50%)*. **Actions** — отображаются ссылки для редактирования и удаления операции.

Настройка операции

Чтобы настроить параметры новой операции, нажмите Add в блоке Operations. Чтобы изменить существующую операцию, нажмите Edit рядом с операцией. Откроется всплывающее окно, в котором можно изменить параметры операции.



Parameter	Description
Object	Доступно пять типов объектов: Прототип элемента данных Прототип триггера Прототип графика Прототип узла сети Прототип обнаружения
Condition	Позволяет фильтровать сущности, к которым должна применяться операция.
Operator	Поддерживаемые операторы: **equals** — применить к этому прототипу **does not equal** — применить ко всем прототипам, кроме этого **contains** — применить, если имя прототипа содержит эту строку **does not contain** — применить, если имя прототипа не содержит эту строку **matches** — применить, если имя прототипа соответствует регулярному выражению **does not match** — применить, если имя прототипа не соответствует регулярному выражению
Pattern	Регулярное выражение или строка для поиска.
Object: *Item pro-to-type*	
Create enabled	Если флажок установлен, появятся переключатели, позволяющие переопределить исходные настройки прототипа элемента данных: *Yes* — элемент данных будет добавлен во включенном состоянии. *No* — элемент данных будет добавлен к обнаруженной сущности, но в отключенном состоянии.
Discover	Если флажок установлен, появятся переключатели, позволяющие переопределить исходные настройки прототипа элемента данных: *Yes* — элемент данных будет добавлен. *No* — элемент данных не будет добавлен.

Parameter		Description
	Update interval	Если флажок установлен, появятся две опции, позволяющие задать другой интервал для элемента данных: *Delay* — интервал обновления элемента данных. Поддерживаются пользовательские макросы и суффиксы времени (например, 30s, 1m, 2h, 1d). Должен быть установлен в 0, если используется *Custom interval*. *Custom interval* — нажмите Add, чтобы указать гибкие интервалы/интервалы по расписанию. Подробную информацию смотрите в разделе Пользовательские интервалы.
	History	Если флажок установлен, появятся переключатели, позволяющие задать другой период хранения истории для элемента данных: *Do not store* — если выбрано, история не будет храниться. *Store up to* — если выбрано, справа появится поле ввода для указания периода хранения. Поддерживаются пользовательские макросы и макросы LLD.
	Trends	Если флажок установлен, появятся переключатели, позволяющие задать другой период хранения трендов для элемента данных: *Do not store* — если выбрано, тренды не будут храниться. *Store up to* — если выбрано, справа появится поле ввода для указания периода хранения. Поддерживаются пользовательские макросы и макросы LLD.
	Tags	Если флажок установлен, появится новый блок, позволяющий указать пары тег-значение. Эти теги будут добавлены к тегам, указанным в прототипе элемента данных, даже если имена тегов совпадают.
Object: *Trigger prototype*		
	Create enabled	Если флажок установлен, появятся переключатели, позволяющие переопределить исходные настройки прототипа триггера: *Yes* — триггер будет добавлен во включенном состоянии. *No* — триггер будет добавлен к обнаруженной сущности, но в отключенном состоянии.
	Discover	Если флажок установлен, появятся переключатели, позволяющие переопределить исходные настройки прототипа триггера: *Yes* — триггер будет добавлен. *No* — триггер не будет добавлен.
	Severity	Если флажок установлен, появятся кнопки важности триггера, позволяющие изменить важность триггера.
	Tags	Если флажок установлен, появится новый блок, позволяющий указать пары тег-значение. Эти теги будут добавлены к тегам, указанным в прототипе триггера, даже если имена тегов совпадают.
Object: *Graph prototype*		
	Discover	Если флажок установлен, появятся переключатели, позволяющие переопределить исходные настройки прототипа графика: *Yes* — график будет добавлен. *No* — график не будет добавлен.
Object: *Host prototype*		
	Create enabled	Если флажок установлен, появятся переключатели, позволяющие переопределить исходные настройки прототипа узла сети: *Yes* — узел сети будет создан во включенном состоянии. *No* — узел сети будет создан в отключенном состоянии.

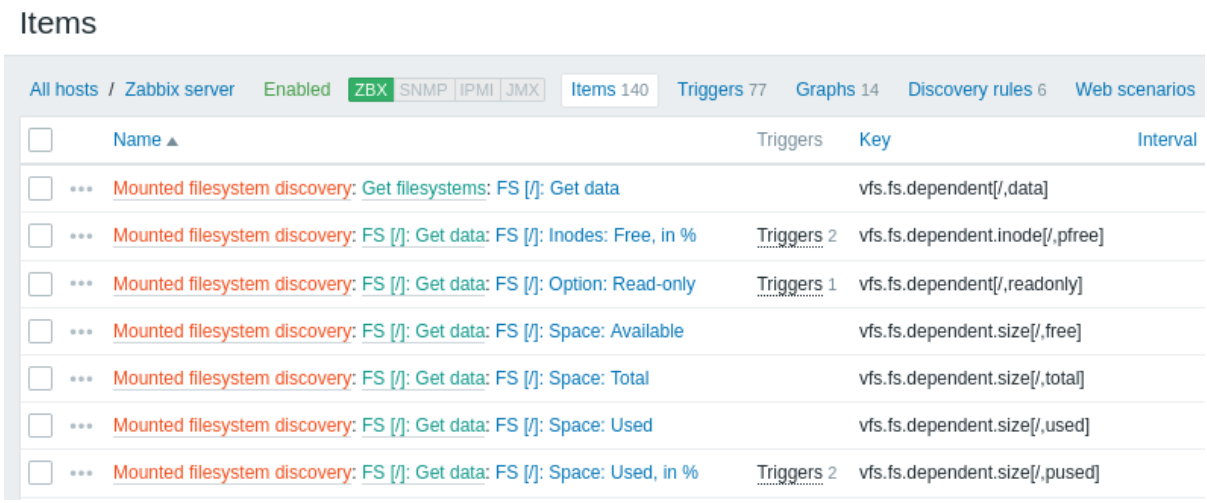
Parameter	Description
Discover	Если флажок установлен, появятся переключатели, позволяющие переопределить исходные настройки прототипа узла сети: *Yes* — узел сети будет обнаружен. *No* — узел сети не будет обнаружен.
Link templates	Если флажок установлен, появится поле ввода для указания шаблонов. Начните вводить имя шаблона или нажмите *Select* рядом с полем и выберите шаблоны из списка во всплывающем окне. Шаблоны из этого переопределения добавляются ко всем шаблонам, уже привязанным к прототипу узла сети.
Tags	Если флажок установлен, появится новый блок, позволяющий указать пары тег-значение. Эти теги будут добавлены к тегам, указанным в прототипе узла сети, даже если имена тегов совпадают.
Host inventory	Если флажок установлен, появятся переключатели, позволяющие выбрать другой режим инвентаризации для прототипа узла сети: *Disabled* — не заполнять инвентаризацию узла сети *Manual* — указывать сведения вручную *Automated* — автоматически заполнять данные инвентаризации узла сети на основе собранных метрик.

Кнопки формы

Кнопки в нижней части формы позволяют выполнять 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. *Обратите внимание*, что при немедленном выполнении обнаружения кэш конфигурации не обновляется, поэтому результат не будет отражать самые последние изменения в конфигурации правила обнаружения.
Test	Проверить конфигурацию правила обнаружения. Используйте эту кнопку, чтобы проверить настройки конфигурации (например, доступность соединения и корректность параметров) без постоянного применения каких-либо изменений.
Delete	Удалить правило обнаружения.
Cancel	Отменить редактирование свойств правила обнаружения.

Обнаруженные сущности На снимках экрана ниже показано, как обнаруженные элементы данных, триггеры и графики выглядят в конфигурации узла сети. Обнаруженные сущности имеют префикс в виде оранжевой ссылки на правило обнаружения, из которого они получены.



Обратите внимание, что обнаруженные сущности не будут созданы, если уже существуют сущности с теми же критериями уникальности, например элемент данных с тем же ключом или график с тем же именем. В этом случае в веб-интерфейсе будет отображено сообщение об ошибке о том, что правило обнаружения низкого уровня не смогло создать определённые сущности. Однако само правило обнаружения не станет неподдерживаемым из-за того, что какая-либо сущность не была создана и её пришлось пропустить. Правило обнаружения продолжит создавать/обновлять другие сущности.

Если обнаруженная сущность (узел сети, файловая система, интерфейс и т. д.) перестаёт обнаруживаться (или больше не проходит фильтр), сущности, созданные на её основе, могут быть автоматически отключены, а затем удалены.

Потерянные ресурсы могут быть автоматически отключены в зависимости от значения параметра *Disable lost resources*. Это влияет на потерянные узлы сети, элементы данных и триггеры.

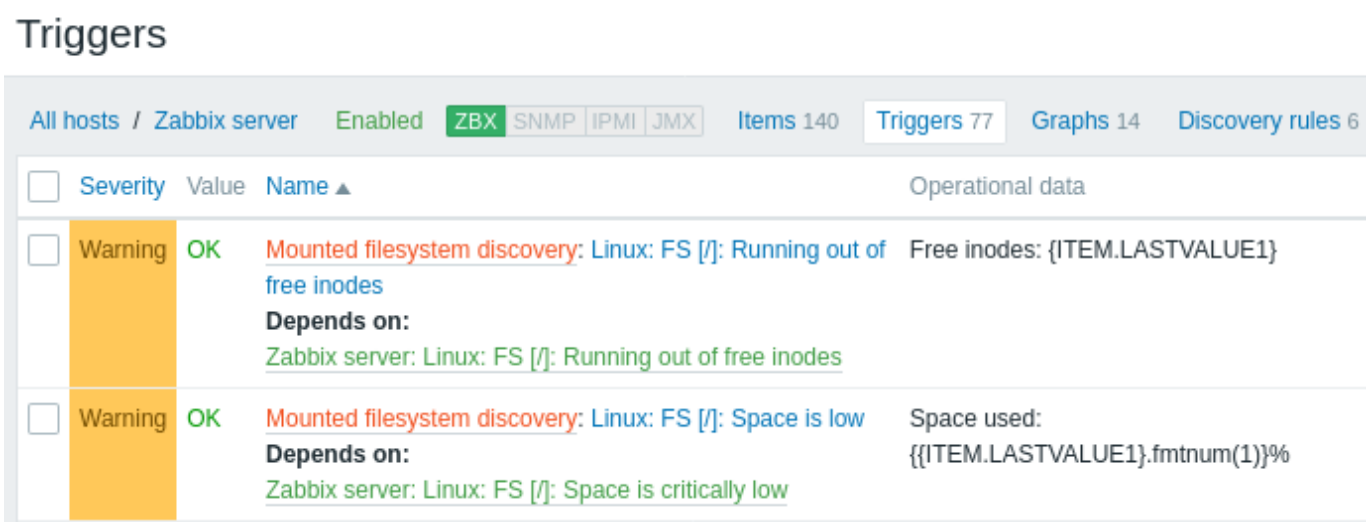
Потерянные ресурсы могут быть автоматически удалены в зависимости от значения параметра *Delete lost resources*. Это влияет на потерянные узлы сети, группы узлов сети, элементы данных, триггеры и графики.

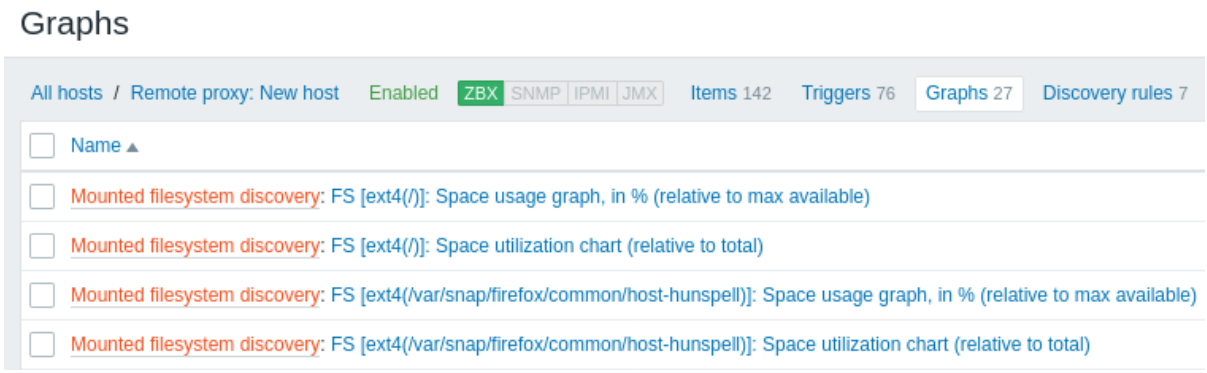
Когда обнаруженные сущности получают статус «Больше не обнаруживаются», в списке сущностей отображается индикатор времени жизни. Наведите на него указатель мыши, и будет показано сообщение с подробностями о его состоянии.



Если сущности были помечены для удаления, но не были удалены в ожидаемое время (из-за отключённого правила обнаружения или узла сети элемента данных), они будут удалены при следующей обработке правила обнаружения.

Сущности, содержащие другие сущности, помеченные для удаления, не будут обновляться при изменении на уровне правила обнаружения. Например, триггеры на основе LLD не будут обновляться, если они содержат элементы данных, помеченные для удаления.





Другие типы обнаружения Более подробная информация и инструкции по другим типам встроенного обнаружения доступны в следующих разделах:

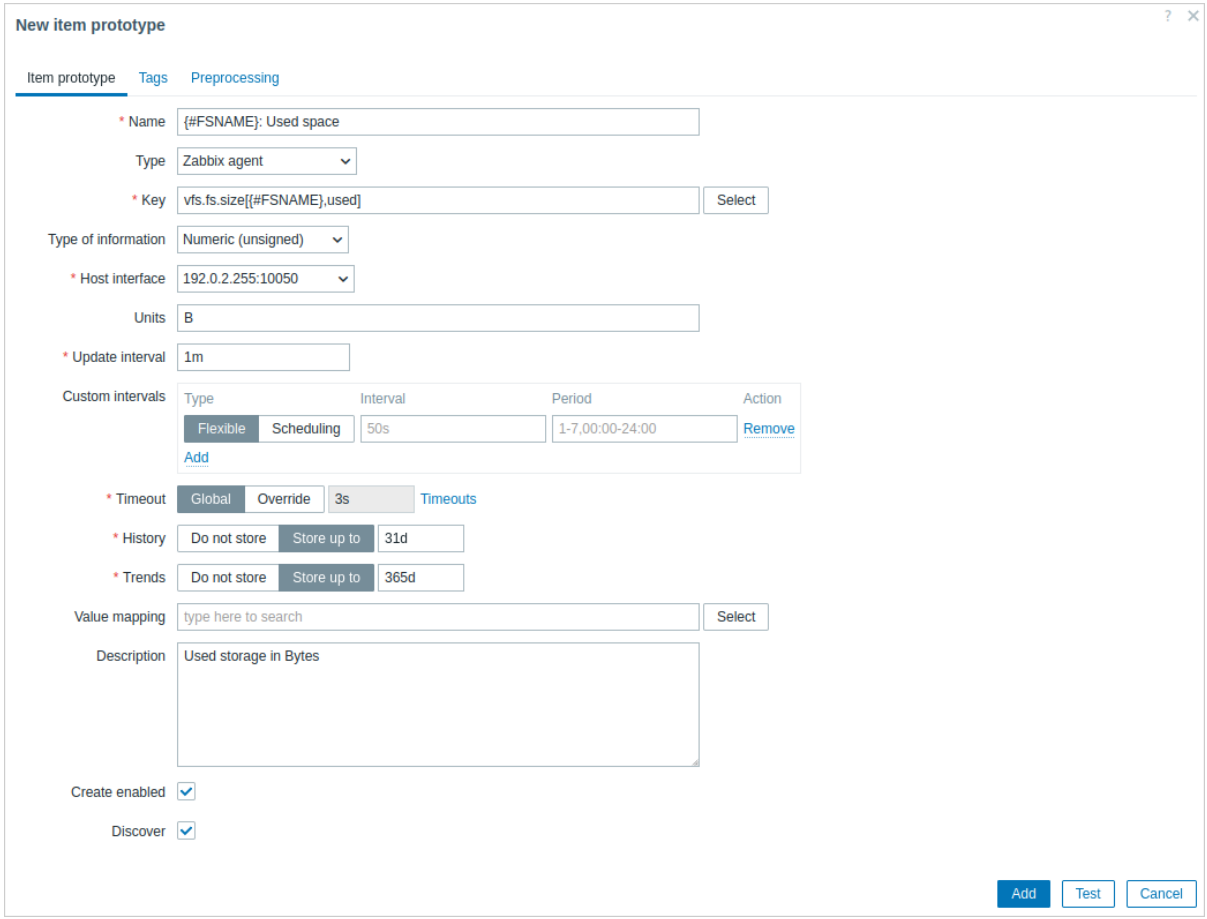
- обнаружение сетевых интерфейсов
- обнаружение процессоров и ядер CPU
- обнаружение SNMP OID
- обнаружение объектов JMX;
- обнаружение с использованием SQL-запросов ODBC
- обнаружение служб Windows
- обнаружение интерфейсов узла сети в Zabbix

Более подробную информацию о формате JSON для элементов данных обнаружения и пример реализации собственного средства обнаружения файловых систем в виде Perl-скрипта см. в разделе создание пользовательских правил LLD.

1 Прототипы элементов данных

После того как правило создано, перейдите к элементам данных этого правила и нажмите ”Создать прототип элемента данных”, чтобы создать прототип элемента данных.

Обратите внимание, как макрос {#FSNAME} используется там, где требуется имя файловой системы. Использование макроса низкоуровневого обнаружения является обязательным в ключе элемента данных, чтобы гарантировать корректную обработку обнаружения. Когда правило обнаружения будет обработано, этот макрос будет заменен обнаруженной файловой системой.



Note:
Если параметры ключа элемента данных не используются, поместите макрос LLD внутрь квадратных скобок ключа элемента данных `[...]` (например, `v_[{#MACRO}]`).

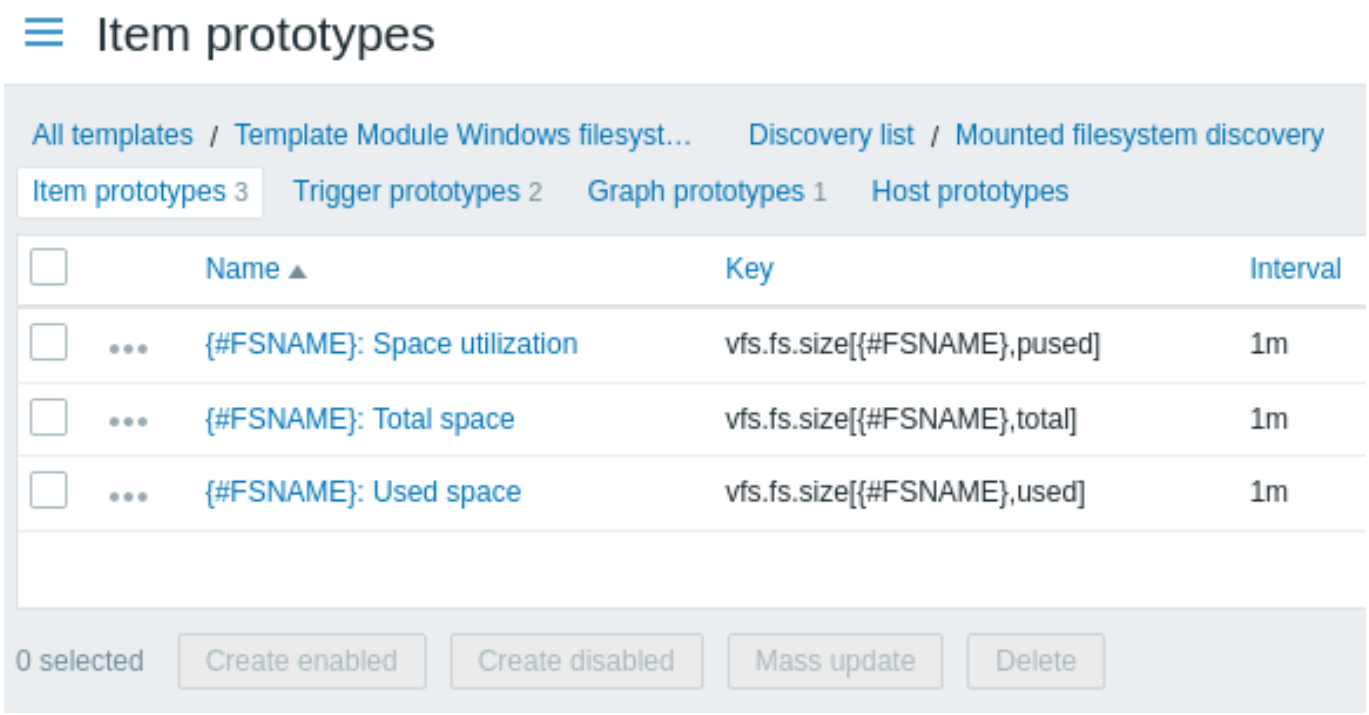
Макросы низкоуровневого обнаружения macros и пользовательские макросы поддерживаются в конфигурации прототипа элемента данных и параметрах предобработки значения элемента данных. Обратите внимание, что при использовании в интервалах обновления один макрос должен заполнять все поле целиком. Несколько макросов в одном поле или макросы, смешанные с текстом, не поддерживаются.

Note:
Для безопасного использования в параметрах предобработки регулярных выражений и XPath выполняется экранирование макросов низкоуровневого обнаружения с учетом контекста.

Атрибуты, специфичные для прототипов элементов данных:

Параметр	Описание
Создавать включенным	Если флажок установлен, элемент данных будет добавлен во включенном состоянии. Если флажок снят, элемент данных будет добавлен к обнаруженной сущности, но в отключенном состоянии.
Обнаруживать	Если флажок установлен (по умолчанию), элемент данных будет добавлен к обнаруженной сущности. Если флажок снят, элемент данных не будет добавлен к обнаруженной сущности, если только этот параметр не переопределен в правиле обнаружения.

Можно создать несколько прототипов элементов данных для каждой интересующей нас метрики файловой системы:



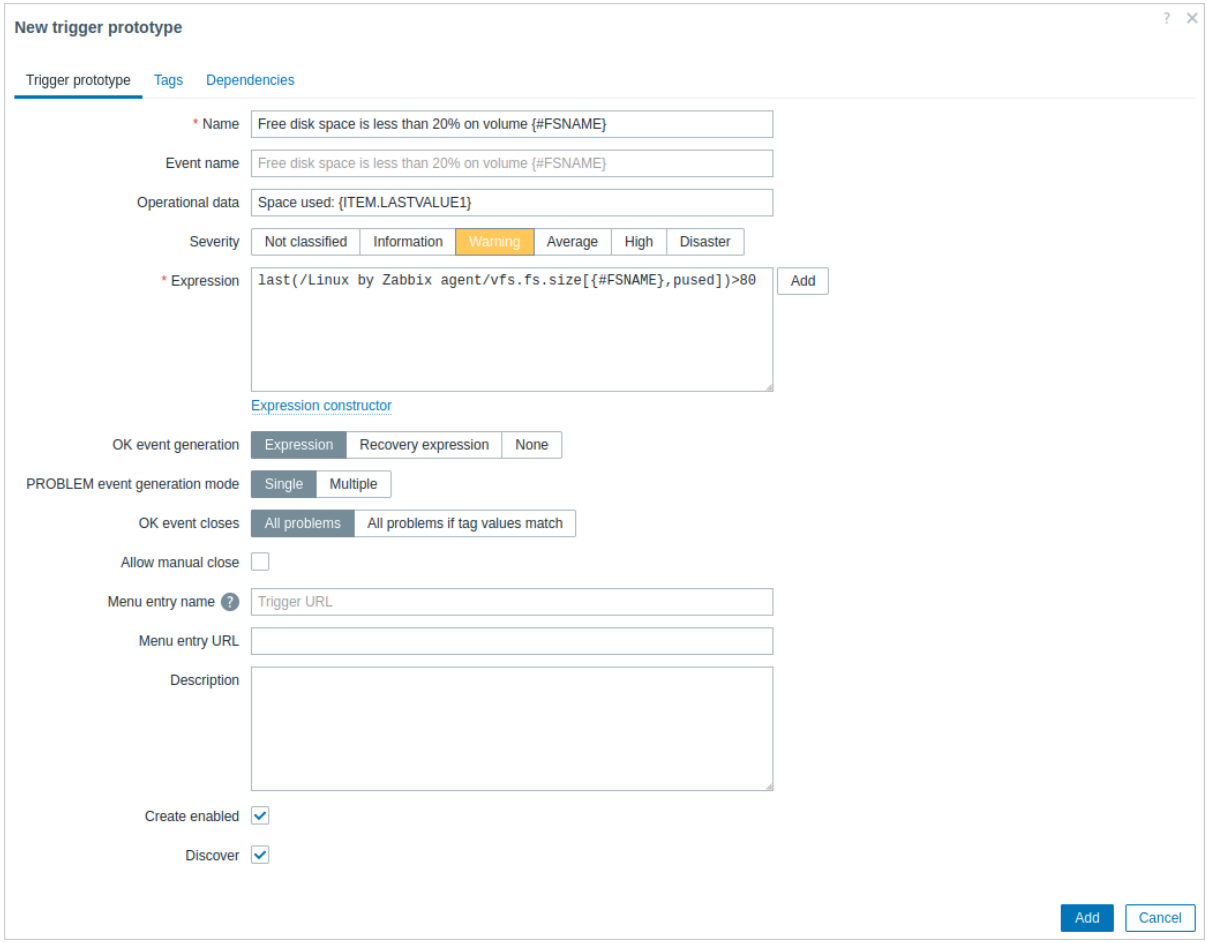
Нажмите на значок с тремя точками, чтобы открыть меню для конкретного прототипа элемента данных со следующими параметрами:

- *Создать прототип триггера* - создать прототип триггера на основе этого прототипа элемента данных
- *Прототипы триггеров* - нажмите, чтобы увидеть список со ссылками 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 вы хотите обновить свойства нескольких прототипов элементов данных одновременно.

2 Прототипы триггеров

Мы создаём прототипы триггеров аналогично прототипам элементов данных:

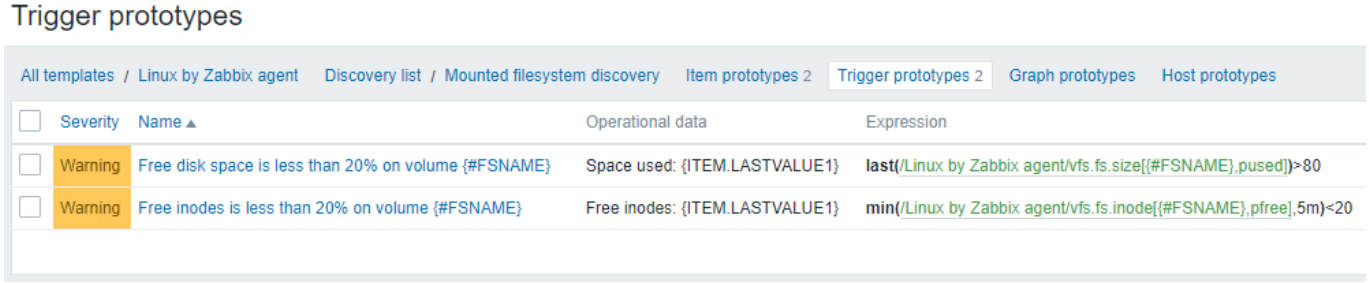


Атрибуты, специфичные для прототипов триггеров:

Parameter	Description
Create enabled	Если отмечено, триггер будет добавлен во включённом состоянии. Если не отмечено, триггер будет добавлен к обнаруженному объекту, но в отключённом состоянии.
Discover	Если отмечено (по умолчанию), триггер будет добавлен к обнаруженному объекту. Если не отмечено, триггер не будет добавлен к обнаруженному объекту, если только этот параметр не переопределён в правиле обнаружения.

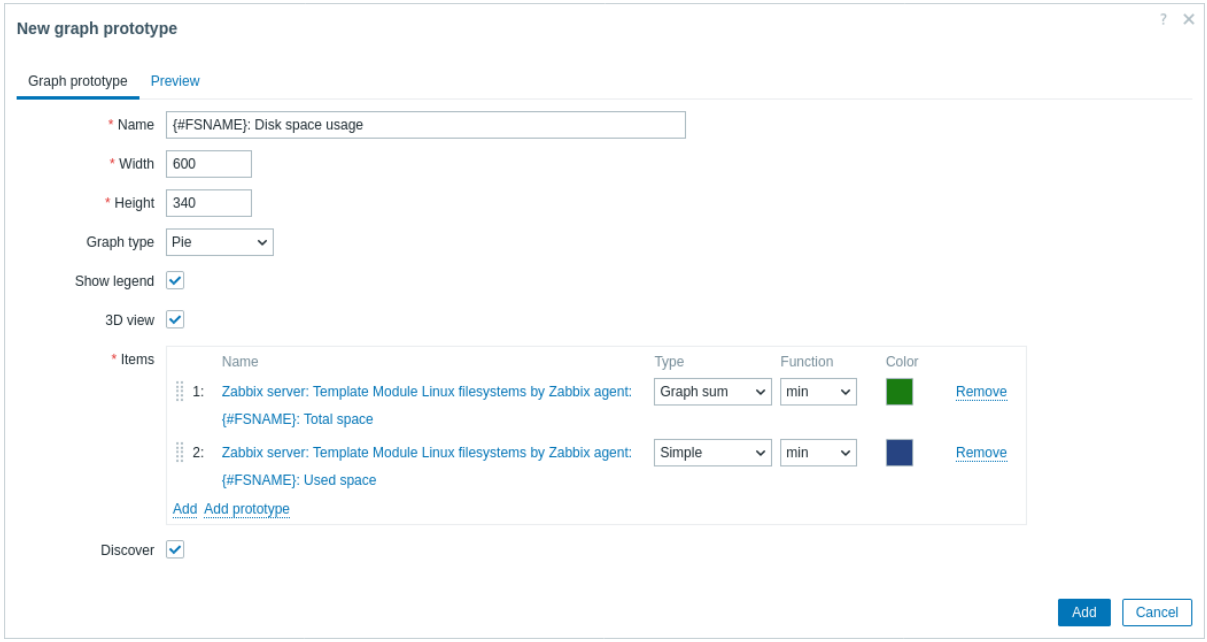
Когда реальные триггеры создаются из прототипов, может возникнуть необходимость в гибкой настройке того, какая константа ('20' в нашем примере) используется для сравнения в выражении. Посмотрите, как пользовательские макросы с контекстом могут быть полезны для достижения такой гибкости.

Вы можете определить зависимости между прототипами триггеров. Для этого перейдите на вкладку *Dependencies*. Прототип триггера может зависеть от другого прототипа триггера из того же правила обнаружения низкого уровня (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.



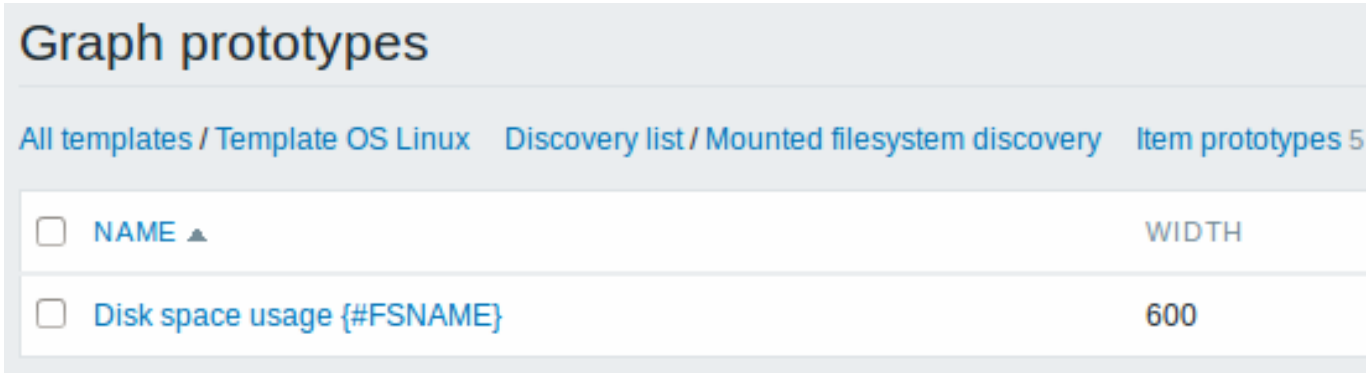
3 Прототипы графиков

Мы также можем создавать прототипы графиков:

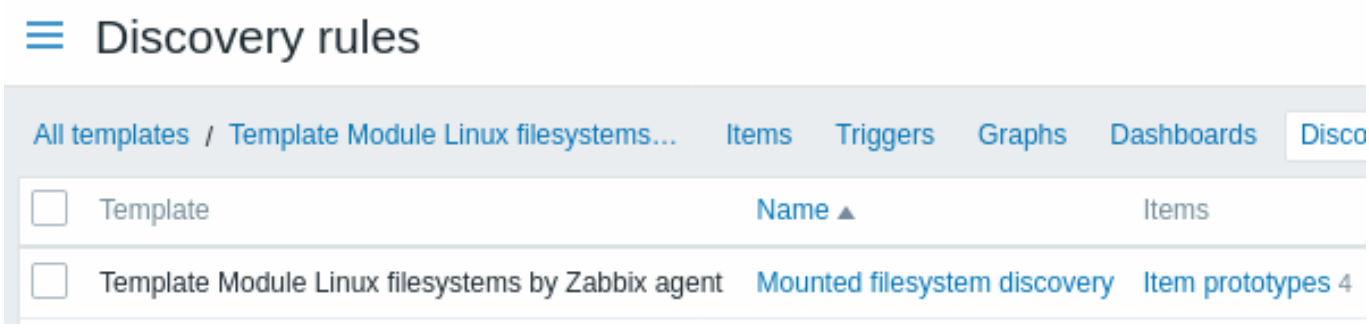


Атрибуты, специфичные для прототипов графиков:

Параметр	Описание
Discover	Если отмечено (по умолчанию), график будет добавлен к обнаруженной сущности. Если не отмечено, график не будет добавлен к обнаруженной сущности, если только этот параметр не переопределён в правиле обнаружения.



Наконец, мы создали правило обнаружения, которое выглядит, как показано ниже. Оно содержит пять прототипов элементов данных, два прототипа триггеров и один прототип графика.



4 Прототипы узлов сети

Прототипы узлов сети — это шаблоны для создания узлов сети с помощью правил обнаружения низкого уровня. До обнаружения в качестве узлов сети эти прототипы не могут иметь элементы данных и триггеры, кроме тех, которые связаны из шаблонов.

Конфигурация

Прототипы узлов сети настраиваются в правилах обнаружения низкого уровня.

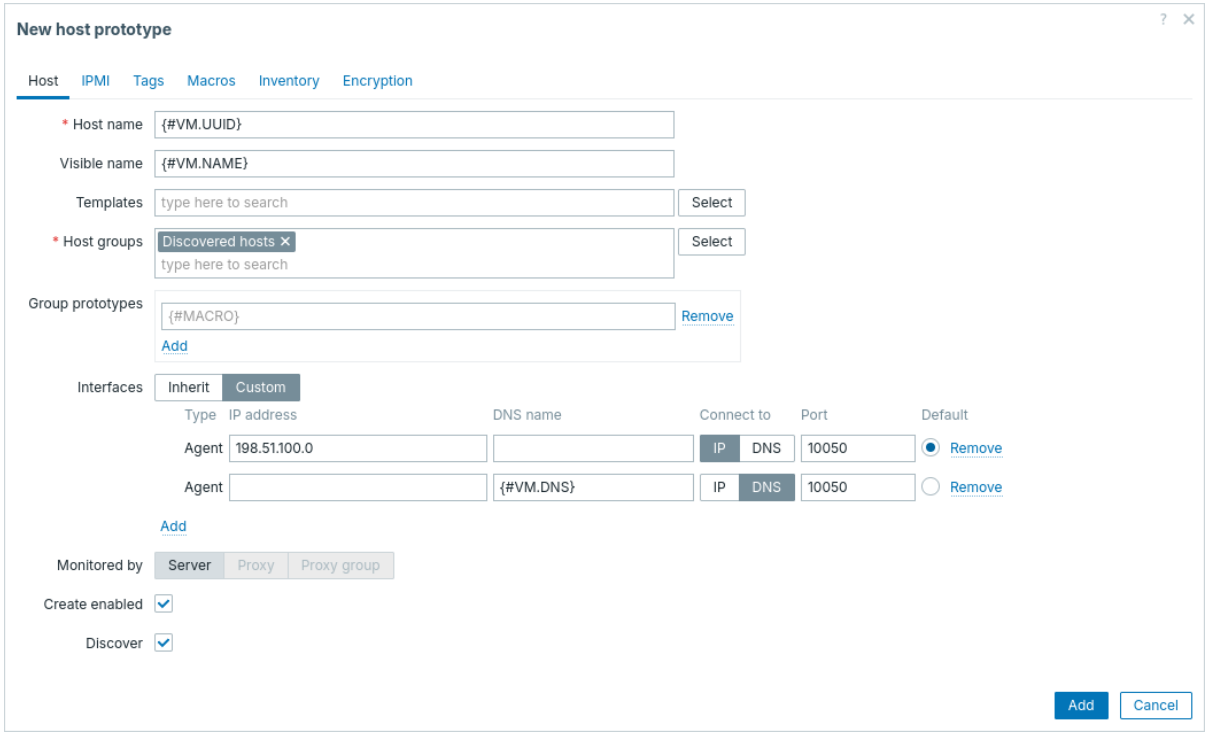
Чтобы создать прототип узла сети:

1. Перейдите в *Сбор данных → Узлы сети*.

2. Нажмите *Обнаружение* для нужного узла сети, чтобы перейти к списку правил обнаружения низкого уровня, 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 конфигурации.



Все обязательные поля ввода отмечены красной звёздочкой.

Параметр	Описание
Имя узла сети	Этот параметр должен содержать как минимум один макрос обнаружения низкого уровня, чтобы обеспечить уникальные имена узлов сети для создаваемых узлов сети. Поддерживаются макросы обнаружения низкого уровня.
Видимое имя	
Прототипы групп	Позволяет указывать прототипы групп узлов сети с использованием макросов обнаружения низкого уровня. На основе указанных прототипов групп будут обнаружены, созданы и привязаны к создаваемым узлам сети группы узлов сети; обнаруженные группы, которые уже были созданы другими правилами обнаружения низкого уровня, также будут привязаны к создаваемым узлам сети. Однако обнаруженные группы узлов сети, совпадающие с созданными вручную группами узлов сети, не будут привязаны к создаваемым узлам сети.
Интерфейсы	Укажите, будут ли обнаруженные узлы сети наследовать IP-адрес от узла сети, которому принадлежит правило обнаружения (по умолчанию), или получать пользовательские интерфейсы. Поддерживаются макросы обнаружения низкого уровня и пользовательские макросы.
Контролируется	Это поле автоматически синхронизируется с соответствующим полем в конфигурации узла сети.
Создавать включённым	Установите статус обнаруженных узлов сети; если флажок снят, узлы сети будут созданы как отключённые.
Обнаруживать	Укажите, будут ли узлы сети создаваться из прототипа узла сети; если флажок снят, узлы сети не будут создаваться из прототипа узла сети (если только этот параметр не переопределён в правиле обнаружения низкого уровня).

Note:
Макросы обнаружения низкого уровня также поддерживаются для значений тегов и значений пользовательских макросов прототипа узла сети.
 Карты соответствия 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 *Interfaces* с "Inherit" на "Custom". Нажмите Add и выберите тип интерфейса — Zabbix агент, SNMP, JMX, IPMI.

Note:
Если выбрано *Custom*, но интерфейсы не заданы, узлы сети будут созданы без интерфейсов.
 Если выбрано *Inherit* и прототип узла сети принадлежит шаблону, все обнаруженные узлы сети унаследуют интерфейс узла сети от узла сети, к которому привязан шаблон.

Если указано несколько пользовательских интерфейсов, основной интерфейс можно задать в столбце *Default*.

Пример настройки пользовательского интерфейса узла сети см. в разделе *Пример настройки мониторинга VMware*.

Warning:
Узел сети будет создан только в том случае, если интерфейс узла сети содержит корректные данные.

Обнаруженные узлы сети

В списке узлов сети обнаруженные узлы сети имеют префикс в виде имени правила обнаружения, которое их создало.

Обнаруженные узлы сети наследуют большинство параметров от прототипов узлов сети в режиме *только для чтения*. Можно настраивать только следующие параметры обнаруженного узла сети:

- *Шаблоны* — привязать новые шаблоны или отвязать добавленные вручную. Шаблоны, унаследованные от прототипа узла сети, отвязать нельзя.
- *Описание* — добавить или изменить описание узла сети.
- *Состояние* — включить или отключить узел сети.
- *Теги* — добавить новые теги или удалить добавленные вручную. Теги, унаследованные от прототипа узла сети, удалить нельзя. Обратите внимание, что теги не могут дублироваться (теги с одинаковыми именем и значением). Если в прототип узла сети добавлен новый тег, совпадающий с тегом, вручную добавленным в обнаруженный узел сети, то во время обнаружения вручную добавленный тег будет заменён унаследованным тегом.
- *Макросы* — добавить новые макросы узла сети или удалить добавленные вручную, а также унаследованные от прототипа узла сети; изменить значения макросов и типы.

Note:
Обнаруженные узлы сети также наследуют пользовательские макросы от узла сети, на котором настроено правило обнаружения. Эти унаследованные макросы можно удалить, а их значения и типы можно изменить.

Обнаруженные узлы сети также могут обнаруживать другие узлы сети. Например, если к прототипу узла сети (используемому для обнаружения гипервизоров) привязан шаблон, который включает правило низкоуровневого обнаружения с собственным прототипом узла сети (используемым для обнаружения виртуальных машин), Zabbix обнаружит как гипервизоры, так и их виртуальные машины. Чтобы дополнительно обнаруживать контейнеры на этих ВМ, вы можете создать правила низкоуровневого обнаружения или привязать новый шаблон с прототипами узлов сети к обнаруженным ВМ, либо заранее настроить прототип узла сети ВМ, привязав шаблон, который сам включает прототипы узлов сети.

Обнаруженные узлы сети можно удалить вручную. Однако обратите внимание, что они будут обнаружены снова, если для них включено обнаружение.

Узлы сети, которые больше не обнаруживаются, могут быть:

- автоматически отключены (в зависимости от значения *Отключать потерянные ресурсы* в правиле обнаружения)
- автоматически удалены (в зависимости от значения *Удалять потерянные ресурсы* в правиле обнаружения).

5 Прототипы обнаружения

Обзор

Прототипы обнаружения — это **вложенные правила обнаружения низкого уровня** внутри "родительского" правила обнаружения, которые позволяют создавать многоуровневое обнаружение объектов с их собственными элементами данных, триггерами и т. д. Например, может потребоваться обнаружить все экземпляры баз данных на сервере баз данных, затем обнаружить табличные пространства для каждого экземпляра, а затем обнаружить таблицы для каждого табличного пространства.

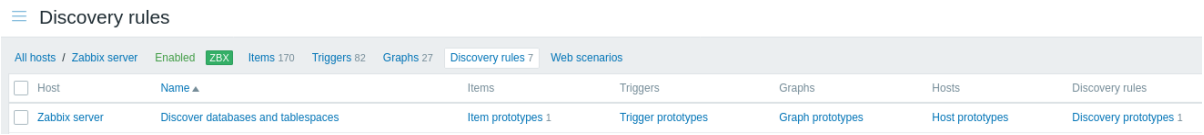
Прототипы обнаружения имеют собственные прототипы элементов данных, триггеров, графиков, узлов сети и обнаружения. Вложенный прототип обнаружения будет использовать то же значение JSON, что и родительское правило, если вы укажете тип *Nested*.

Количество уровней вложенности для прототипов обнаружения не ограничено.

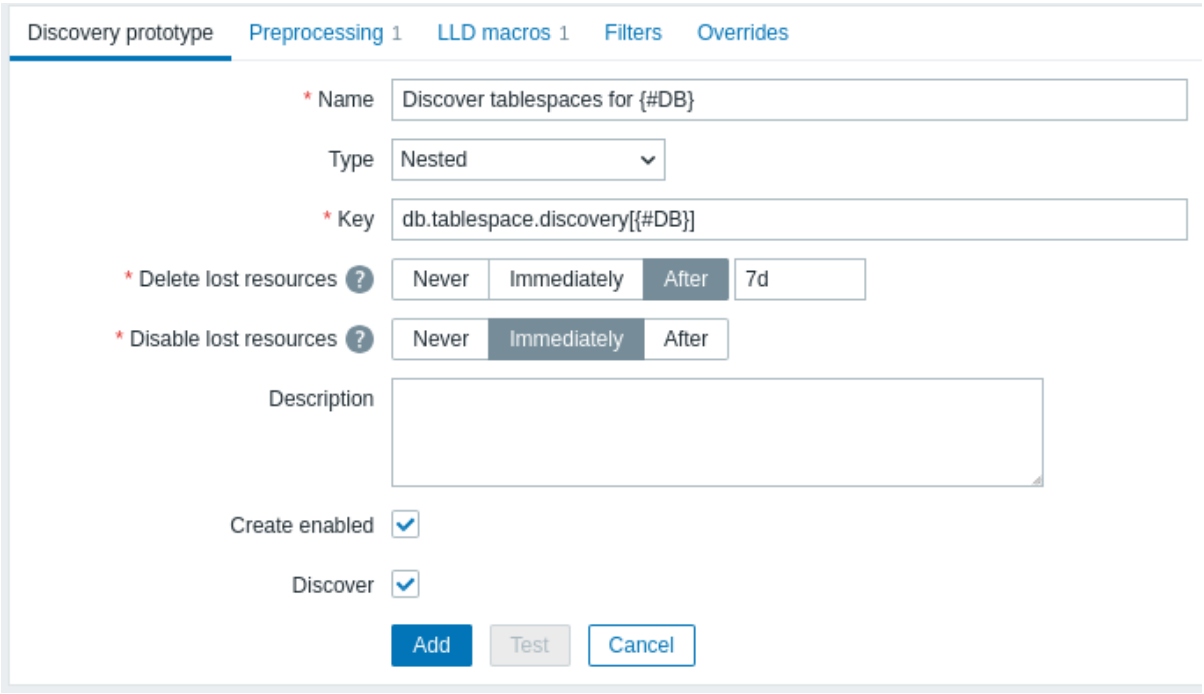
Настройка

Чтобы создать прототип обнаружения:

- Нажмите *Прототипы обнаружения* в строке существующего правила обнаружения



- Нажмите *Создать прототип обнаружения*



Поля настройки этой формы являются общими с обычным обнаружением низкого уровня.

Если в открывшейся форме прототипа обнаружения выбрать "Nested" в поле *Тип*, тогда правила обнаружения (из прототипа обнаружения) будут созданы на основе JSON-объекта из того же значения JSON, что и родительское правило обнаружения. Например, если исходный JSON имеет вид [<object A>, <object B>] и существует один прототип вложенного правила обнаружения, тогда будут созданы два правила обнаружения на основе данных объекта A и объекта B соответственно.

В этом случае прототип обнаружения активируется одновременно с родительским правилом. Таким образом, вложенное правило может использовать предобработку для работы с другим "срезом" тех же данных, уже полученных родительским правилом.

LLD-макросы из родительского правила LLD 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, а 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, 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 *Active connections to {#DB}* (Тип: Агент, Ключ: db.connections[{#DB}]).
4. Создайте правило LLD для этого шаблона с именем *Discover tablespaces* (Тип: Вложенный).
5. Перейдите на вкладку *Preprocessing* этого правила и добавьте шаг JSONPath=$.tablespaces.
6. Перейдите на вкладку *LLD Macros* этого правила, добавьте макрос {#TSNAME}=$.name.
7. Создайте прототип элемента данных для этого правила с именем *Size of tablespace {#TSNAME} for {#DB}* (Тип: Агент, Ключ: db.ts.size[{#DB}, {#TSNAME}]).
8. Вернувшись к корневому узлу сети, создайте правило LLD для этого узла сети с именем *Discover databases and tablespaces*.
9. Перейдите на вкладку *LLD Macros* этого правила, добавьте макрос {#DB}=$.database.
10. Добавьте прототип узла сети для этого правила с именем *Host for database {#DB}*.
11. Перейдите на вкладку *Macros* этого прототипа узла сети, добавьте макрос {$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 ее табличным пространствам:

Host	Items
Host for database db1	Active connections to db1
	Size of tablespace ts1 for db1
	Size of tablespace ts2 for db1
	Size of tablespace ts3 for db1
Host for database db2	Active connections to db2
	Size of tablespace ts1 for db2
	Size of tablespace ts2 for db2
	Size of tablespace ts3 for db2
Host for database db3	Active connections to db3
	Size of tablespace ts1 for db3
	Size of tablespace ts2 for db3
	Size of tablespace ts3 for db3

6 Примечания по обнаружению низкого уровня

Использование LLD макросов в контекстах пользовательских макросов

LLD макросы могут использоваться внутри контекста пользовательского макроса, например, в прототипах триггера.

Несколько правил LLD для одного и того же элемента

Можно определить несколько правил низкоуровневого обнаружения с одним и тем же элементом данных для обнаружения.

Чтобы сделать это, вам необходимо определить параметр агента Alias, чтобы разрешить использовать отличающиеся ключи элементов данных обнаружений в различных правилах обнаружения, например: vfs.fs.discovery[foo], vfs.fs.discovery[bar] и т.д.

Ограничения данных для возвращаемых значений

Для JSON-данных правила низкоуровневого обнаружения ограничение отсутствует, если они получены напрямую сервером Zabbix. Это связано с тем, что возвращаемые значения обрабатываются без сохранения в базе данных.

Для пользовательских правил низкоуровневого обнаружения ограничение также отсутствует. Однако, если данные пользовательского правила низкоуровневого обнаружения получаются с использованием пользовательского параметра, применяется ограничение возвращаемого значения пользовательского параметра.

Если данные должны проходить через прокси Zabbix, ему необходимо сохранять эти данные в базе данных. В таком случае применяются ограничения базы данных.

7 Правила обнаружения

Пожалуйста, используйте боковую панель, чтобы посмотреть примеры настройки правил обнаружения для различных случаев.

1 Обнаружение примонтированных файловых систем

Обзор

Можно обнаруживать смонтированные файловые системы и их свойства:

- имя точки монтирования
- тип файловой системы
- размер файловой системы
- статистику inode
- параметры монтирования

Для этого можно использовать сочетание:

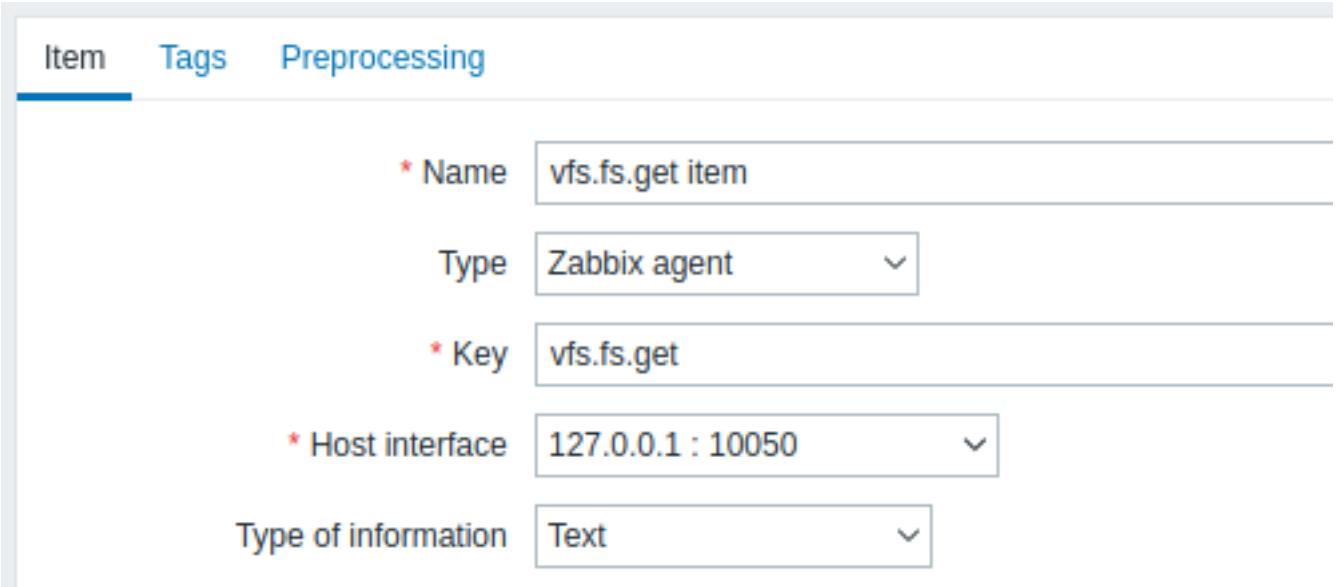
- элемента данных агента `vfs.fs.get` в качестве мастер-элемента данных
- зависимого правила низкоуровневого обнаружения и прототипов элементов данных

Настройка

Основной элемент данных

Создайте элемент данных агента Zabbix, используя следующий ключ:

`vfs.fs.get`



Установите тип информации в ”Текст” для потенциально больших данных JSON.

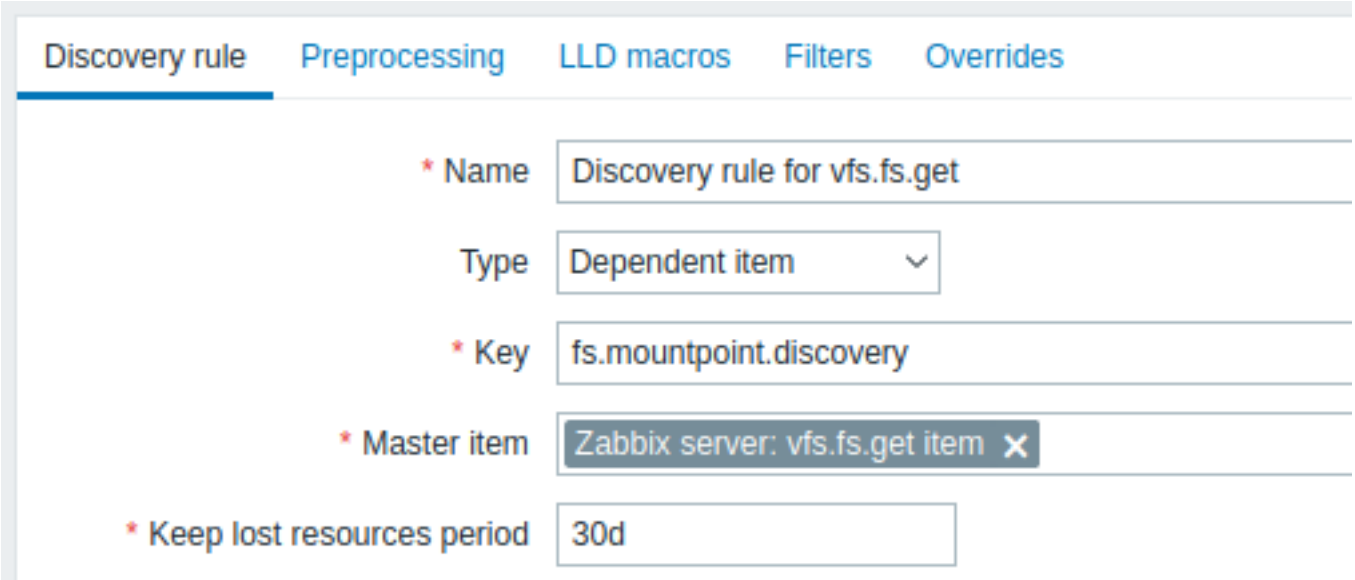
Данные, возвращаемые этим элементом данных, будут содержать примерно следующее для смонтированной файловой системы:

```
[
  {
    "fsname": "/",
    "fstype": "ext4",
    "bytes": {
      "total": 249405239296,
      "free": 24069537792,
```

```
      "used": 212595294208,
      "pfree": 10.170306,
      "pused": 89.829694
    },
    "inodes": {
      "total": 15532032,
      "free": 12656665,
      "used": 2875367,
      "pfree": 81.487503,
      "pused": 18.512497
    },
    "options": "rw,noatime,errors=remount-ro"
  }
]
```

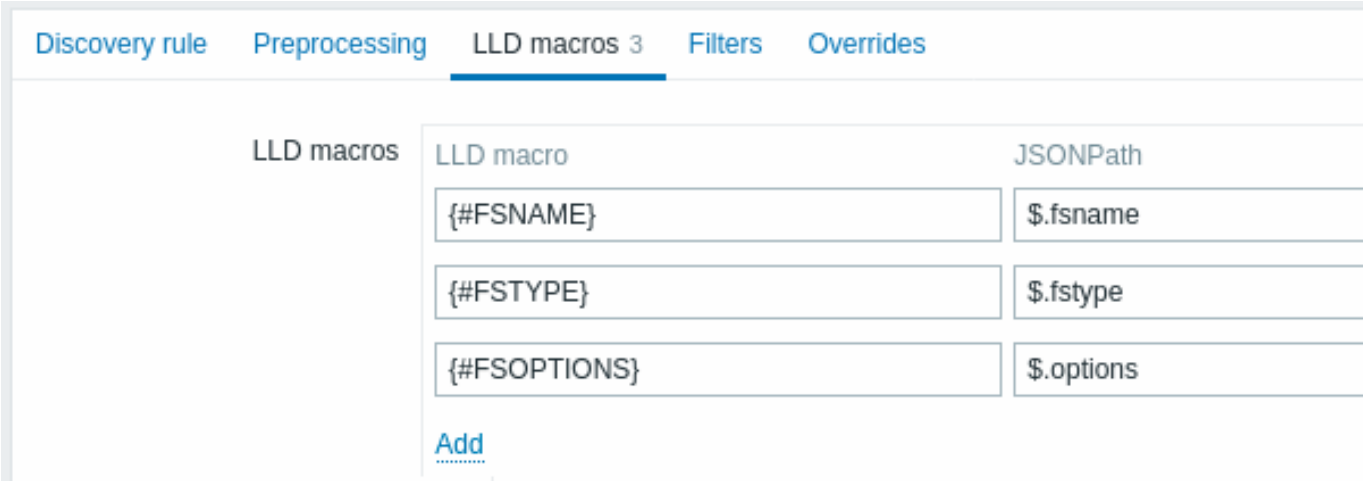

Зависимое правило LLD

Создайте правило низкоуровневого обнаружения с типом «Зависимый элемент данных»:

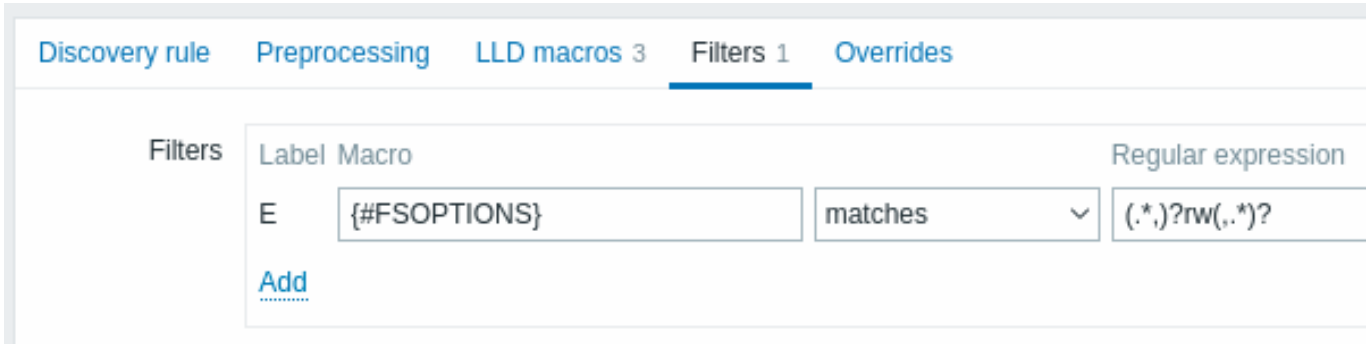


В качестве мастер-элемента данных выберите созданный нами элемент данных `vfs.fs.get`.

На вкладке «LLD макросы» определите пользовательские макросы с соответствующим JSONPath:

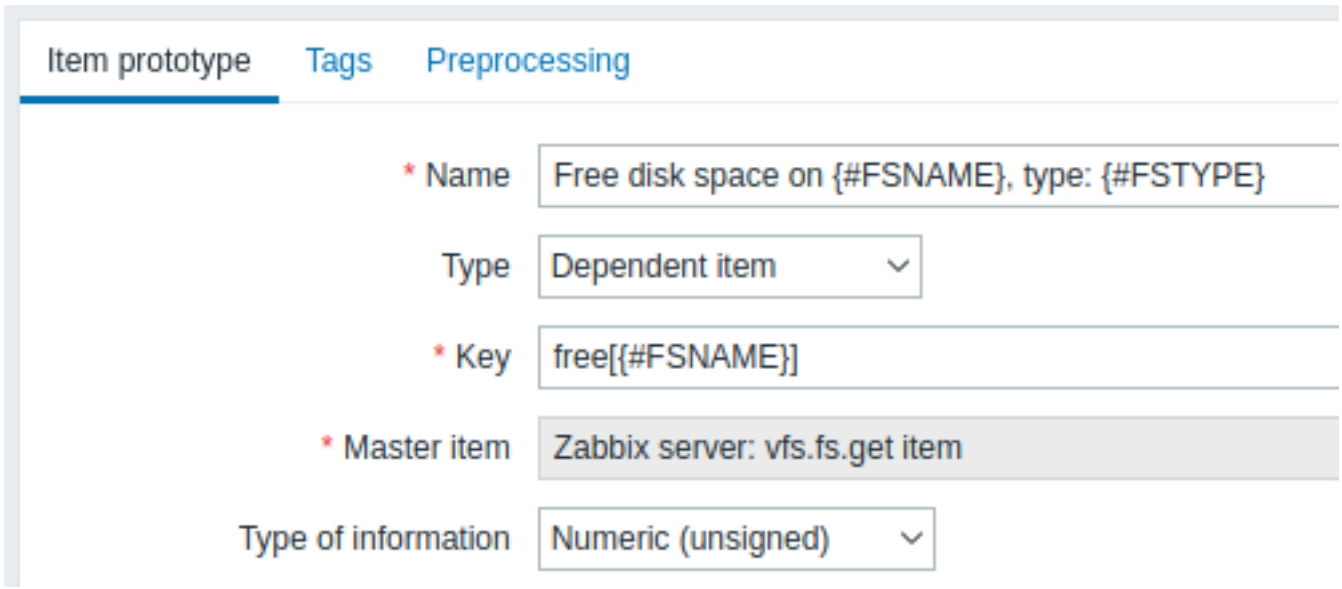


На вкладке «Фильтры» вы можете добавить регулярное выражение, которое отфильтрует только файловые системы **для чтения и записи**:



Прототип зависимого элемента данных

Создайте в этом правиле LLD прототип элемента данных с типом "Зависимый элемент данных". В качестве мастер-элемента данных для этого прототипа выберите созданный нами элемент данных `vfs.fs.get`.



Обратите внимание на использование пользовательских макросов в имени и ключе прототипа элемента данных:

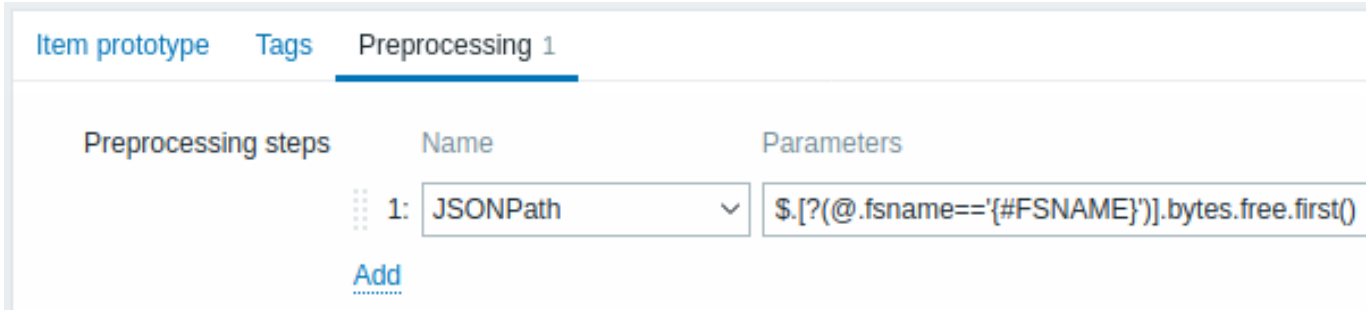
- *Имя*: Свободное место на диске {#FSNAME}, тип: {#FSTYPE}
- *Ключ*: Free[{#FSNAME}]

В качестве типа информации используйте:

- *Numeric (unsigned)* для таких метрик, как 'free', 'total', 'used'
- *Numeric (float)* для таких метрик, как 'pfree', 'pused' (в процентах)

На вкладке "Предобработка" прототипа элемента данных выберите JSONPath и используйте следующее выражение JSONPath в качестве параметра:

```
$.[?(@.fsname=='{#FSNAME}')].bytes.free.first()
```



Когда обнаружение начнется, будет создано по одному элементу данных для каждой точки монтирования. Этот элемент данных будет возвращать количество свободных байтов для заданной точки монтирования.

2 Обнаружение сетевых интерфейсов

Аналогично тому, как обнаруживаются файловые системы, можно также обнаруживать сетевые интерфейсы.

Ключ элемента данных

Ключ элемента данных, который следует использовать в правиле обнаружения, —

`net.if.discovery`

Поддерживаемые макросы

Вы можете использовать макрос {#IFNAME} в фильтре правила обнаружения и в прототипах элементов данных, триггеров и графиков.

Примеры прототипов элементов данных, которые можно создать на основе ”net.if.discovery”:

- ”net.if.in[{#IFNAME},bytes]”,
- ”net.if.out[{#IFNAME},bytes]”.

Обратите внимание, что в Windows также возвращается {#IFGUID}.

3 Обнаружение CPUs и ядер CPU

Аналогично file systems обнаружены, можно также обнаружить CPUs и ядра CPU.

Ключ элемента данных

Ключ элемента данных, который следует использовать в правиле обнаружения:

`system.cpu.discovery`

Поддерживаемые макросы

Этот ключ обнаружения возвращает два макроса — {#CPU.NUMBER} и {#CPU.STATUS}, которые соответственно определяют порядковый номер и состояние CPU. Он перечисляет процессоры от 0 до N - 1, где N — общее количество процессоров.

```
[
    {
        "{#CPU.NUMBER}": 0,
        "{#CPU.STATUS}": "online"
    },
    {
        "{#CPU.NUMBER}": 1,
        "{#CPU.STATUS}": "offline"
    },
    {
        "{#CPU.NUMBER}": 2,
        "{#CPU.STATUS}": "unknown" /* "unknown" is only on Windows */
    },
    {
        "{#CPU.NUMBER}": 3,
        "{#CPU.STATUS}": "online"
    }
]
```

Обратите внимание, что невозможно провести чёткое различие между фактическими физическими процессорами, ядрами и гиперпотоками. {#CPU.STATUS} в системах Linux, UNIX и BSD возвращает состояние процессора, которое может быть либо ”online”, либо ”offline”. В системах Windows этот же макрос может принимать третье значение — ”unknown”, что указывает на то, что процессор был обнаружен, но информация о нём ещё не была собрана.

Обнаружение CPU опирается на процесс collector агента, чтобы сохранять согласованность с данными, предоставляемыми collector, и экономить ресурсы при получении данных. В результате этот ключ элемента данных не работает с флагом командной строки тестирования (-t) бинарного файла агента, который возвращает статус NOT_SUPPORTED и сопутствующее сообщение о том, что процесс collector не был запущен.

Прототипы элементов данных, которые могут быть созданы на основе обнаружения CPU, включают, например:

- `system.cpu.util[{#CPU.NUMBER},<type>,<mode>]`
- `system.hw.cpu[{#CPU.NUMBER},<info>]`

Подробное описание ключей элементов данных см. в разделе ключи элементов данных агента Zabbix.

Примечания по обнаруженным элементам данных счётчиков производительности в NUMA-системах

Некоторые счётчики производительности Windows могут быть недоступны для некоторых логических процессоров в NUMA-системах.

Например, элементы данных, обнаруженные с помощью прототипа элемента данных с указанным ниже ключом, могут работать только для первого NUMA-узла. Элементы данных для других NUMA-узлов могут находиться в неподдерживаемом состоянии.

```
perf_counter[\Processor({#CPU.NUMBER})\% Processor Time,60]
```

Кроме того, правило обнаружения с ключом "system.cpu.discovery" не предоставляет никаких значений макросов низкоуровневого обнаружения для подстановки в такие счётчики производительности, как:

```
perf_counter["\Processor Information(<NUMA node index>,<CPU index in NUMA node>)\% Processor Time",60]
```

В то время как элементы данных system.cpu.util используют другие счётчики производительности для мониторинга процента использования CPU в зависимости от количества групп процессоров и количества процессоров. Также существует редкая ошибка, связанная с группами процессоров в Windows.

4 Обнаружение SNMP OID-ов

Обзор

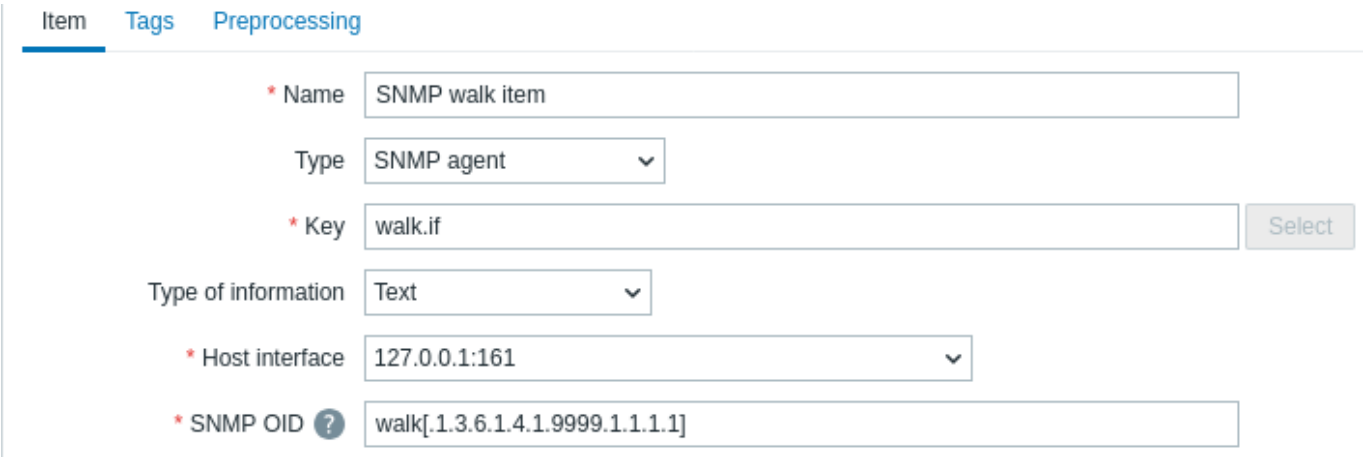
В этом разделе мы выполним обнаружение низкого уровня на устройстве SNMP.

Этот метод обнаружения SNMP OID поддерживается начиная с Zabbix сервер/прокси 6.4.

Пример конфигурации

1. Создайте SNMP-элемент данных агента с ключом, например:

```
walk[.1.3.6.1.4.1.9999.1.1.1.1]
```



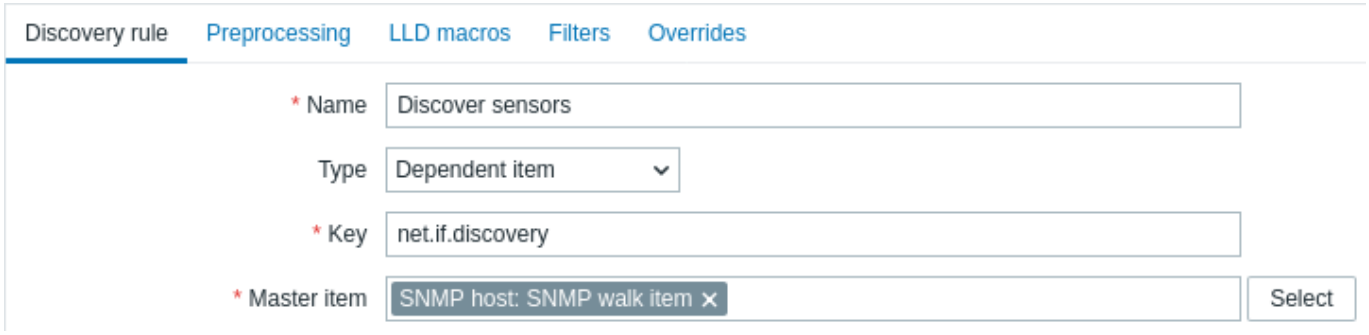
Этот элемент данных выполняет один обход SNMP-таблицы и возвращает все записи таблицы в одном запросе в формате, соответствующем выводу утилиты `snmpwalk` с параметрами форматирования `-Oe -Ot -On`.

Он вернет следующее многострочное текстовое значение:

```
.1.3.6.1.4.1.9999.1.1.1.1.1.1 = STRING: "Temperature Sensor"
.1.3.6.1.4.1.9999.1.1.1.1.2.1 = STRING: "temp"
.1.3.6.1.4.1.9999.1.1.1.1.3.1 = 100
.1.3.6.1.4.1.9999.1.1.1.1.1.2 = STRING: "Humidity Sensor"
.1.3.6.1.4.1.9999.1.1.1.1.2.2 = STRING: "humidity"
.1.3.6.1.4.1.9999.1.1.1.1.3.2 = 200
```

2. Создайте правило обнаружения:

- В поле *Name* введите понятное имя правила обнаружения (например, "Обнаружение датчиков").
- В поле *Type* выберите "Dependent item".
- В поле *Key* введите понятный ключ (например, "net.if.discovery").
- В поле *Master item* выберите "SNMP walk item".



3. На вкладке *Preprocessing* добавьте шаг предобработки с именем "SNMP walk to JSON" в выпадающем списке *Name* с 3 параметрами:

- *Field name*: "{#SENSORNAME}"; *OID prefix*: ".1.3.6.1.4.1.9999.1.1.1.1.1": *Format*: "Unchanged".
- *Field name*: "{#SENSORTYPE}"; *OID prefix*: ".1.3.6.1.4.1.9999.1.1.1.1.2": *Format*: "Unchanged".
- *Field name*: "{#SENSORVALUE}"; *OID prefix*: ".1.3.6.1.4.1.9999.1.1.1.1.3": *Format*: "Unchanged".

После предобработки правило обнаружения возвращает JSON-массив наборов макросов.

Например:

```
[
    {
        "{#SNMPINDEX}": "1",
        "{#SENSORNAME}": "Temperature Sensor",
        "{#SENSORTYPE}": "temp",
        "{#SENSORVALUE}": "100"
    },
    {
        "{#SNMPINDEX}": "2",
        "{#SENSORNAME}": "Humidity Sensor",
        "{#SENSORTYPE}": "humidity",
        "{#SENSORVALUE}": "200"
    }
]
```

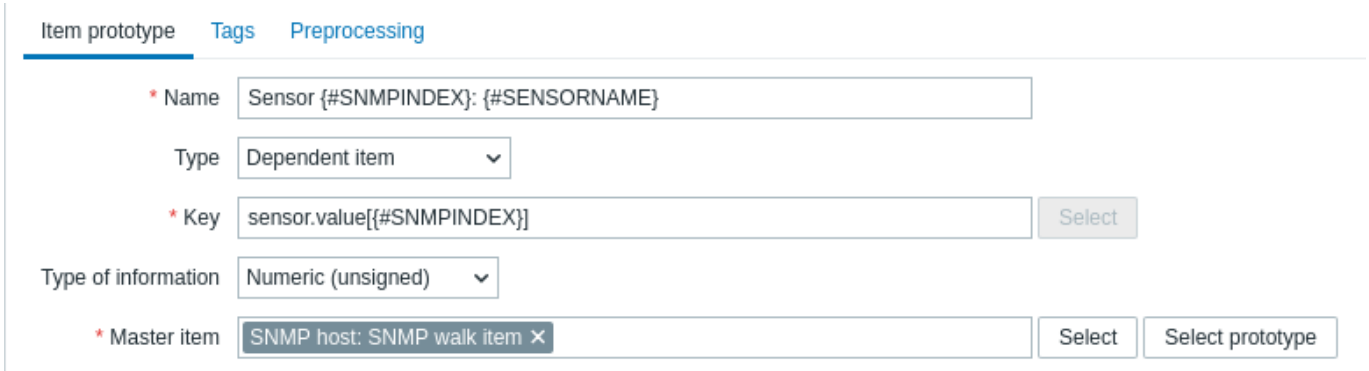
Каждый объект представляет один обнаруженный датчик и предоставляет такие макросы, как {#SNMPINDEX}, {#SENSORNAME}, {#SENSORTYPE} и {#SENSORVALUE}.

Они группируются по индексу SNMP, который является числовым суффиксом в конце каждого OID (например, .1, .2) — этот индекс уникально идентифицирует каждую строку в SNMP-таблице и автоматически извлекается как {#SNMPINDEX}.

4. Внутри правила обнаружения создайте один или несколько прототипов элементов данных (с правилом обнаружения в качестве master item).

Например, зависимый элемент данных значения датчика:

- В поле *Name* введите "Sensor {#SNMPINDEX}: {#SENSORNAME}".
- В поле *Type* выберите "Dependent item".
- В поле *Key* введите "sensor.value[{#SNMPINDEX}]".
- В поле *Master item* выберите "SNMP walk item".



На вкладке *Preprocessing* добавьте шаг предобработки с именем "SNMP walk value" и OID ".1.3.6.1.4.1.9999.1.1.1.1.3.{#SNMPINDEX}" в поле *Parameter*. *Format*: "Unchanged".

Будут обнаружены следующие элементы данных:

Name	Key	OID from which value is extracted	Item value
Sensor 1: Temperature Sensor	sensor.value[1]	1.3.6.1.4.1.9999.1.1.1.1.3.1	100
Sensor 2: Humidity Sensor	sensor.value[2]	1.3.6.1.4.1.9999.1.1.1.1.3.2	200

Когда правило обнаружения выполняется, создаются такие элементы данных, как `sensor.value[1]`, `sensor.value[2]`.

Каждый зависимый элемент данных извлекает свое значение из результата SNMP walk master-элемента данных с помощью предобработки, не выполняя собственных отдельных SNMP-запросов.

5. Ссылайтесь на прототипы зависимых элементов данных в прототипах триггеров, используя те же макросы из правила обнаружения. Пример:

```
{Template_Sensor:sensor.value[{#SNMPINDEX}].last()} > 75
```

Это создает триггер для каждого обнаруженного датчика (например, sensor.value[1], sensor.value[2]) и срабатывает, если последнее значение (температуры или влажности) превышает 75.

6. Включите зависимые элементы данных для каждой обнаруженной сущности. Пример ключа элемента графика:

```
sensor.value[{#SNMPINDEX}]
```

Для каждого `{#SNMPINDEX}` создается один график, отображающий температуру и влажность во времени.

Эта конфигурация выполняет только один запрос SNMP walk за цикл опроса, независимо от количества обнаруженных элементов данных. Все зависимые элементы данных извлекают свои значения из результата master SNMP walk с помощью предобработки, что значительно снижает SNMP-трафик и нагрузку.

Динамические индексы с walk[]

Динамические индексы (например, индексы интерфейсов) могут смещаться при изменении конфигурации оборудования. Чтобы учесть такое поведение, создается правило обнаружения master SNMP walk с ключом, например:

```
walk[1.3.6.1.2.1.2.2.1.10]
```

После предобработки SNMP walk to JSON результат может выглядеть так:

```
[
    {
        "{#SNMPINDEX}": "2",
        "{#VALUE}": "123456"
    },
    {
        "{#SNMPINDEX}": "3",
        "{#VALUE}": "654321"
    }
]
```

Прототип зависимого элемента данных использует макрос `{#SNMPINDEX}` для формирования ключа:

```
net.if.in[{#SNMPINDEX}]
```

Предобработка для этого прототипа включает имя ”SNMP walk value” с OID ”1.3.6.1.2.1.2.2.1.10.{#SNMPINDEX}” в поле *Parameter*. *Format*: ”Unchanged”.

Во время выполнения создаются фактические элементы данных, такие как `net.if.in[2]` и `net.if.in[3]`. Если индекс заданного интерфейса изменяется (например, если индекс 2 заменяется на 5 в таблице SNMP), то при следующем выполнении правила обнаружения:

- Старый зависимый элемент данных net.if.in[2] помечается как ”lost” или удаляется, и новые данные для этого элемента данных больше не собираются.
- Создается новый зависимый элемент данных net.if.in[5], начиная с пустой историей.
- Исторические данные из net.if.in[2] не переносятся автоматически в net.if.in[5].

Пример прототипа триггера:

```
{Template_Interface:net.if.in[{#SNMPINDEX}].last()} > 1000000000
```

Пример прототипа графика включает элементы данных:

```
net.if.in[{#SNMPINDEX}]
net.if.out[{#SNMPINDEX}]
```

Такая конфигурация обеспечивает надежный мониторинг таблиц с динамическими индексами при минимизации SNMP-трафика — требуется только один SNMP walk за цикл опроса, а прототипы зависимых элементов данных извлекают необходимые значения.

Обнаруженные сущности

Когда сервер запущен, он создаст реальные зависимые элементы данных, триггеры и графики на основе значений, которые возвращает правило обнаружения SNMP.

5 Обнаружение OID SNMP (устаревшее)

Обзор

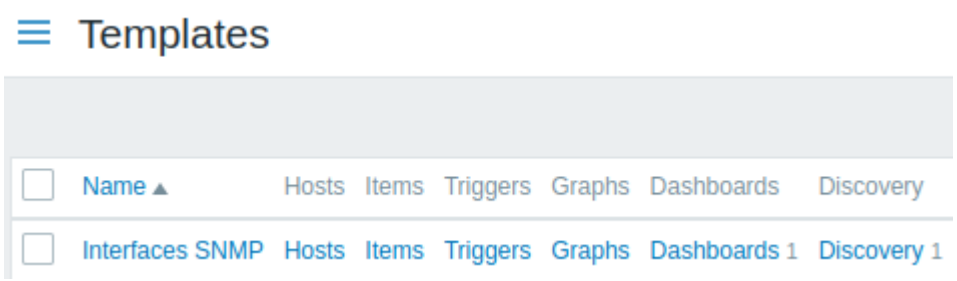
В этом разделе мы выполним SNMP-обнаружение на коммутаторе.

Ключ элемента данных

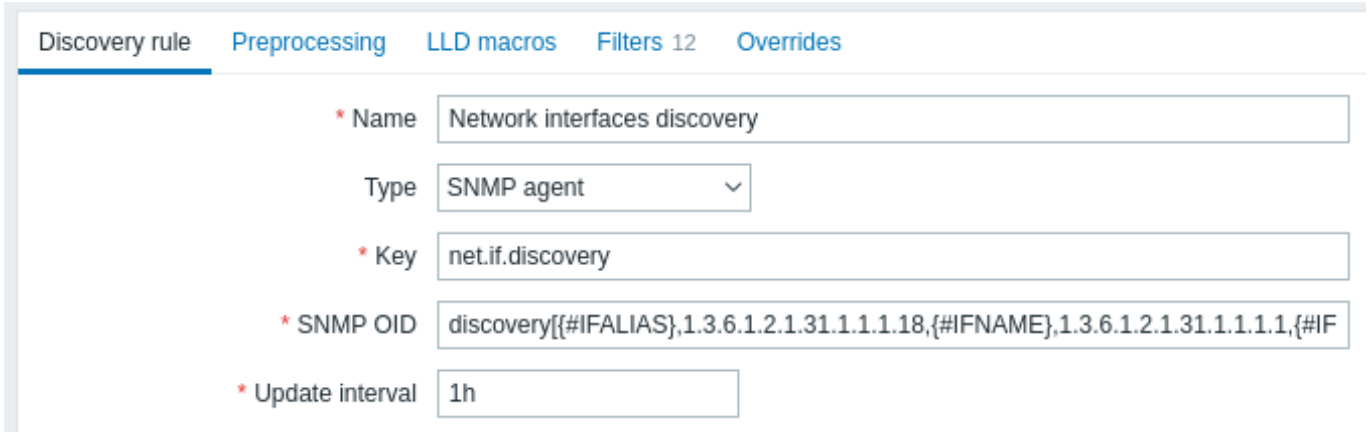
В отличие от обнаружения файловых систем и сетевых интерфейсов, элементу данных необязательно иметь ключ ”snmp.discovery” — достаточно, чтобы типом элемента данных был SNMP агент.

Чтобы настроить правило обнаружения, выполните следующие действия:

- Перейдите в: *Data collection* → *Templates*
- Нажмите *Discovery* в строке соответствующего шаблона



- Нажмите *Create discovery rule* в правом верхнем углу экрана
- Заполните форму правила обнаружения необходимыми данными, как показано на снимке экрана ниже



Все обязательные поля ввода отмечены красной звездочкой.

OID для обнаружения задаются в поле SNMP OID в следующем формате: `discovery[{#MACRO1}, oid1, {#MACRO2}, oid2, …,]`

где *{#MACRO1}*, *{#MACRO2}* ... — это допустимые имена lld-макросов, а *oid1*, *oid2*... — OID, способные возвращать осмысленные значения для этих макросов. Ко всем обнаруженным сущностям применяется встроенный макрос *{#SNMPINDEX}*, содержащий индекс обнаруженного OID. Обнаруженные сущности группируются по значению макроса *{#SNMPINDEX}*.

Ниже приведен пример, в котором на коммутаторе выполняется несколько snmpwalk:

```
$ snmpwalk -v 2c -c public 192.168.1.1 IF-MIB::ifDescr
IF-MIB::ifDescr.1 = STRING: WAN
IF-MIB::ifDescr.2 = STRING: LAN1
IF-MIB::ifDescr.3 = STRING: LAN2

$ snmpwalk -v 2c -c public 192.168.1.1 IF-MIB::ifPhysAddress
IF-MIB::ifPhysAddress.1 = STRING: 8:0:27:90:7a:75
IF-MIB::ifPhysAddress.2 = STRING: 8:0:27:90:7a:76
IF-MIB::ifPhysAddress.3 = STRING: 8:0:27:2b:af:9e
```

И SNMP OID задан как: `discovery[{#IFDESCR}, ifDescr, {#IFPHYSADDRESS}, ifPhysAddress]`

Теперь это правило обнаружит сущности, у которых макросы {#IFDESCR} установлены в **WAN**, **LAN1** и **LAN2**, макросы {#IFPHYSADDRESS} установлены в **8:0:27:90:7a:75**, **8:0:27:90:7a:76** и **8:0:27:2b:af:9e**, а макросы {#SNMPINDEX} установлены в индексы обнаруженных OID **1**, **2** и **3**:

```
[
    {
        "{#SNMPINDEX}": "1",
        "{#IFDESCR}": "WAN",
        "{#IFPHYSADDRESS}": "8:0:27:90:7a:75"
    },
    {
        "{#SNMPINDEX}": "2",
        "{#IFDESCR}": "LAN1",
        "{#IFPHYSADDRESS}": "8:0:27:90:7a:76"
    },
    {
        "{#SNMPINDEX}": "3",
        "{#IFDESCR}": "LAN2",
        "{#IFPHYSADDRESS}": "8:0:27:2b:af:9e"
    }
]
```

Если у сущности нет указанного OID, то соответствующий макрос для этой сущности будет пропущен. Например, если имеются следующие данные:

```
ifDescr.1 "Interface #1"
ifDescr.2 "Interface #2"
ifDescr.4 "Interface #4"

ifAlias.1 "eth0"
ifAlias.2 "eth1"
ifAlias.3 "eth2"
ifAlias.5 "eth4"
```

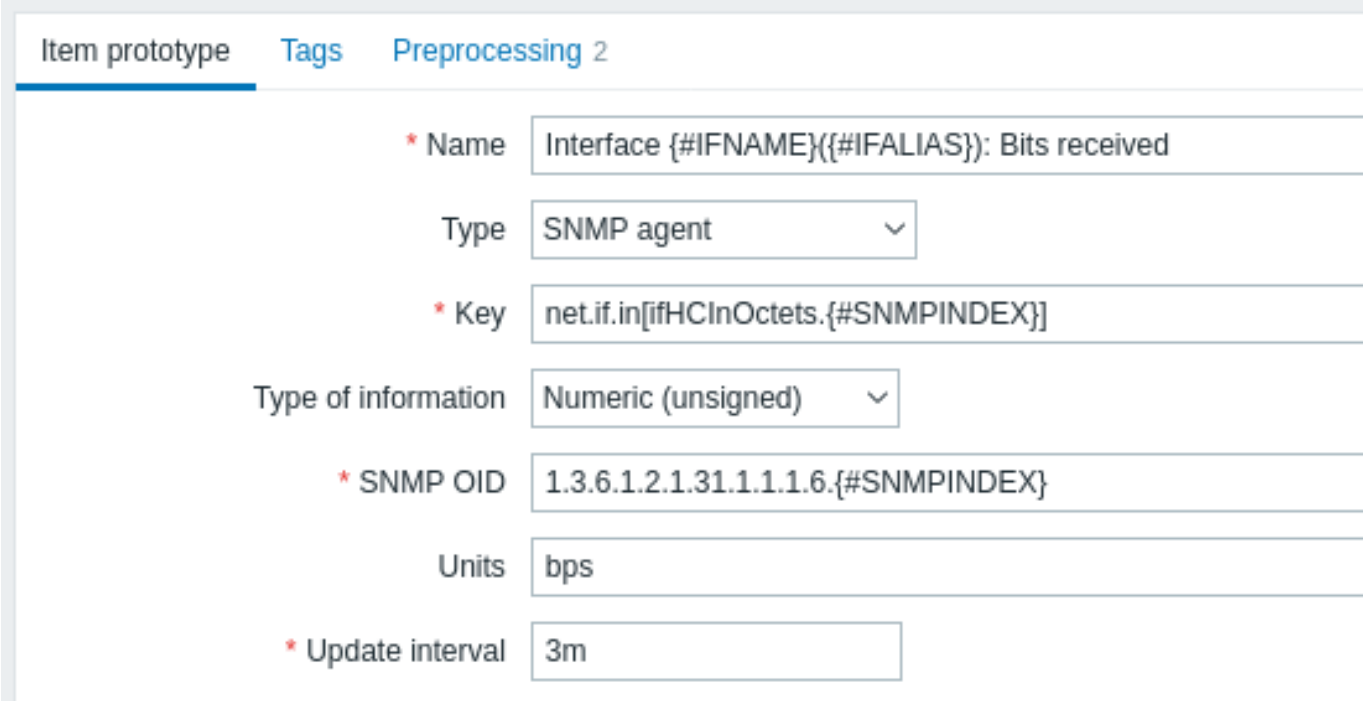
Тогда в этом случае SNMP-обнаружение `discovery[{#IFDESCR}, ifDescr, {#IFALIAS}, ifAlias]` вернет следующую структуру:

```
[
    {
        "{#SNMPINDEX}": 1,
        "{#IFDESCR}": "Interface #1",
        "{#IFALIAS}": "eth0"
    },
    {
        "{#SNMPINDEX}": 2,
        "{#IFDESCR}": "Interface #2",
        "{#IFALIAS}": "eth1"
    },
    {
        "{#SNMPINDEX}": 3,
        "{#IFALIAS}": "eth2"
    },
    {
```

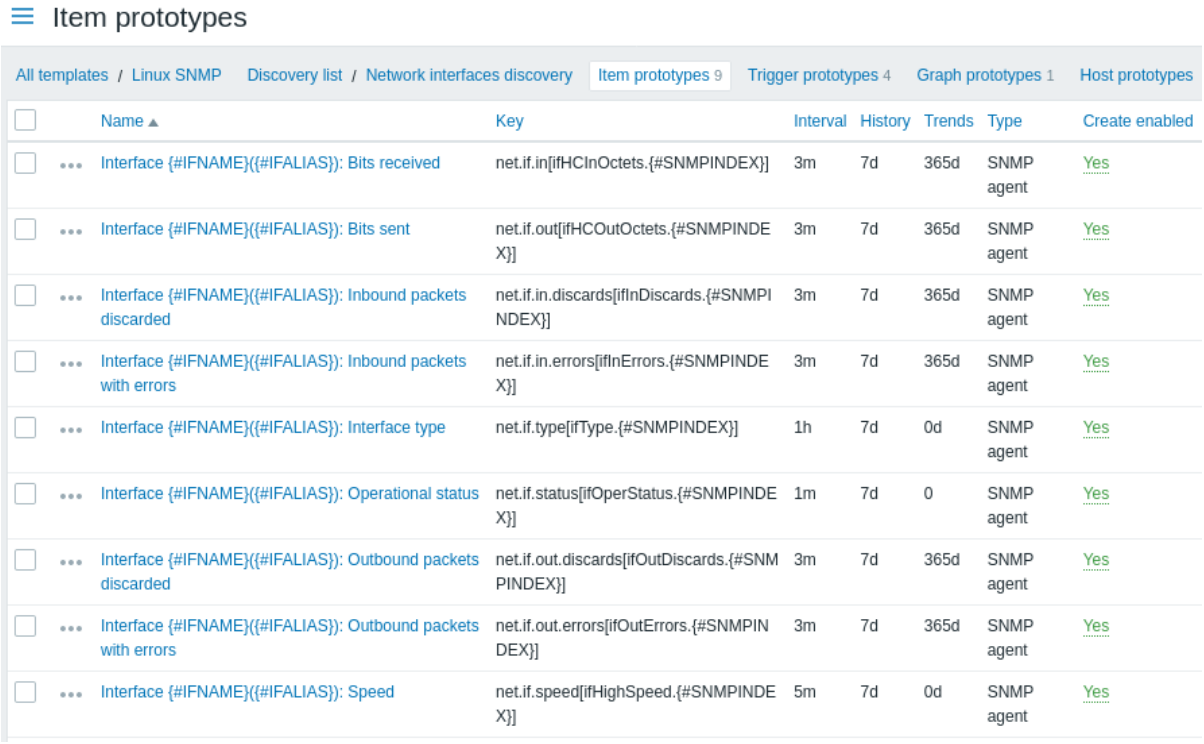
```
        "{#SNMPINDEX}": 4,
        "{#IFDESCR}": "Interface #4"
    },
    {
        "{#SNMPINDEX}": 5,
        "{#IFALIAS}": "eth4"
    }
]
```

Прототипы элементов данных

Следующий снимок экрана показывает, как можно использовать эти макросы в прототипах элементов данных:



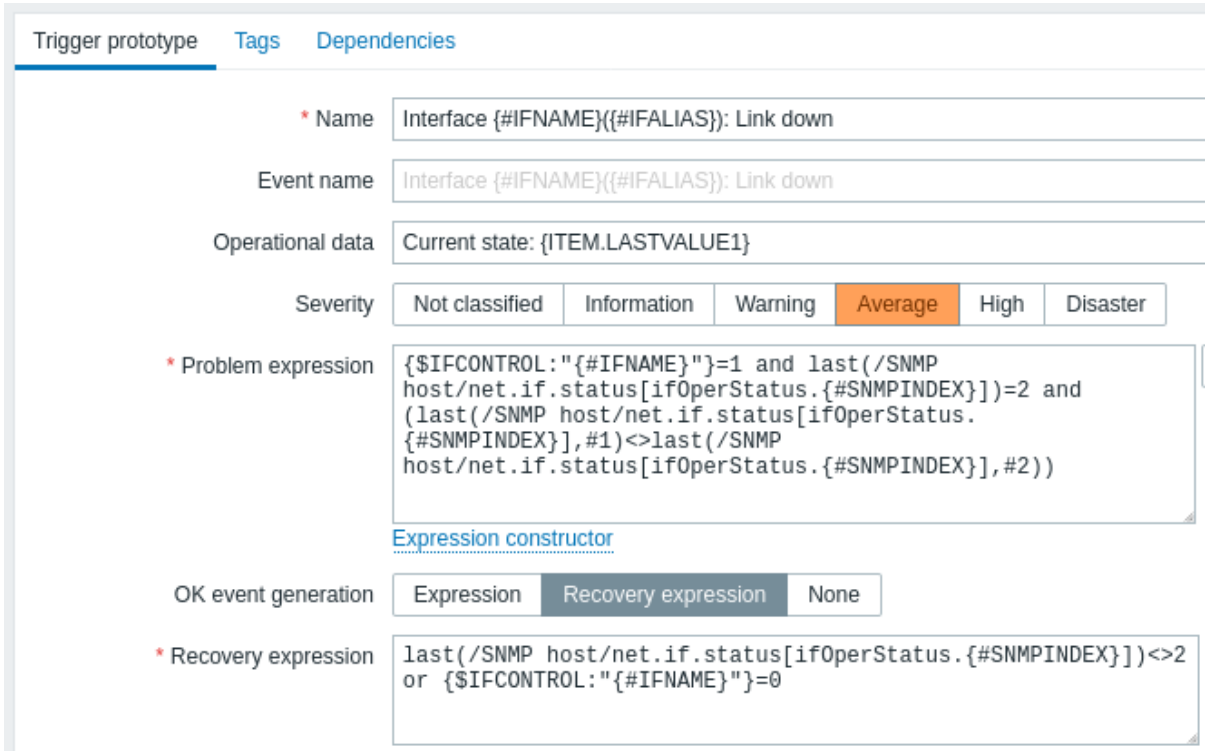
Вы можете создать столько прототипов элементов данных, сколько необходимо:

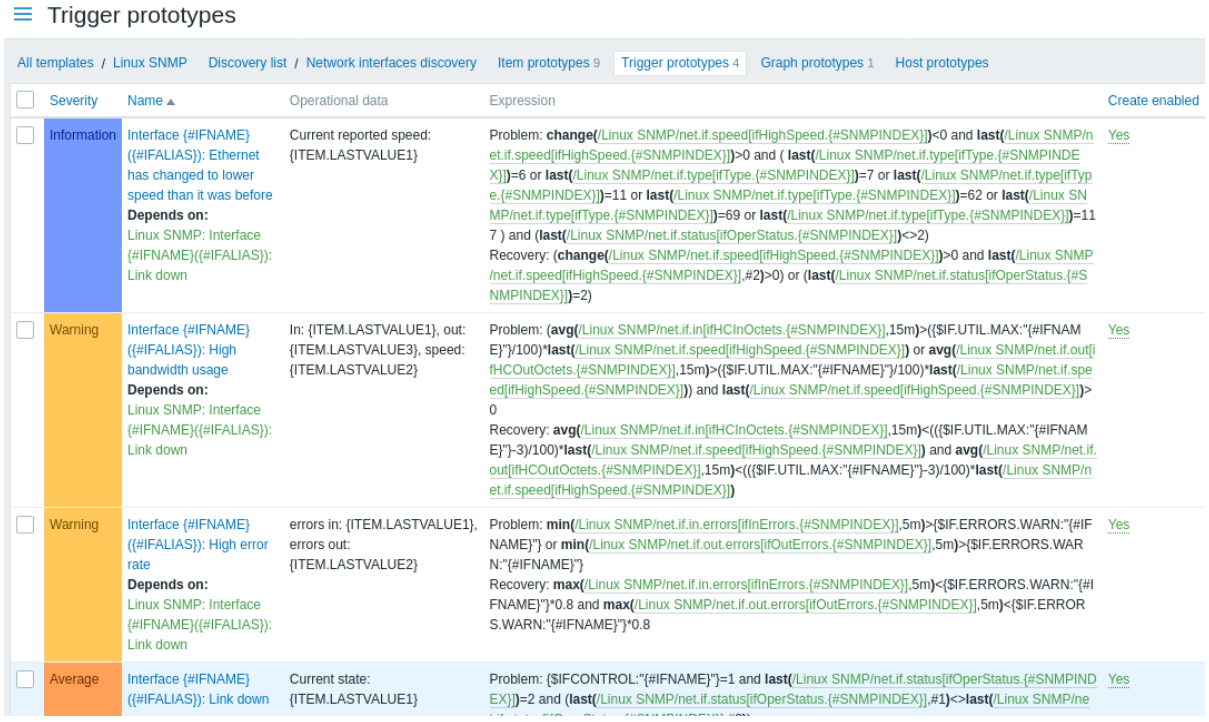


Name ▲	Key	Interval	History	Trends	Type	Create enabled
Interface {#IFNAME}({#IFALIAS}): Bits received	net.if.in[ifHCInOctets.{#SNMPINDEX}]	3m	7d	365d	SNMP agent	Yes
Interface {#IFNAME}({#IFALIAS}): Bits sent	net.if.out[ifHCOutOctets.{#SNMPINDEX}]	3m	7d	365d	SNMP agent	Yes
Interface {#IFNAME}({#IFALIAS}): Inbound packets discarded	net.if.in.discards[ifInDiscards.{#SNMPINDEX}]	3m	7d	365d	SNMP agent	Yes
Interface {#IFNAME}({#IFALIAS}): Inbound packets with errors	net.if.in.errors[ifInErrors.{#SNMPINDEX}]	3m	7d	365d	SNMP agent	Yes
Interface {#IFNAME}({#IFALIAS}): Interface type	net.if.type[ifType.{#SNMPINDEX}]	1h	7d	0d	SNMP agent	Yes
Interface {#IFNAME}({#IFALIAS}): Operational status	net.if.status[ifOperStatus.{#SNMPINDEX}]	1m	7d	0	SNMP agent	Yes
Interface {#IFNAME}({#IFALIAS}): Outbound packets discarded	net.if.out.discards[ifOutDiscards.{#SNMPINDEX}]	3m	7d	365d	SNMP agent	Yes
Interface {#IFNAME}({#IFALIAS}): Outbound packets with errors	net.if.out.errors[ifOutErrors.{#SNMPINDEX}]	3m	7d	365d	SNMP agent	Yes
Interface {#IFNAME}({#IFALIAS}): Speed	net.if.speed[ifHighSpeed.{#SNMPINDEX}]	5m	7d	0d	SNMP agent	Yes

Прототипы триггеров

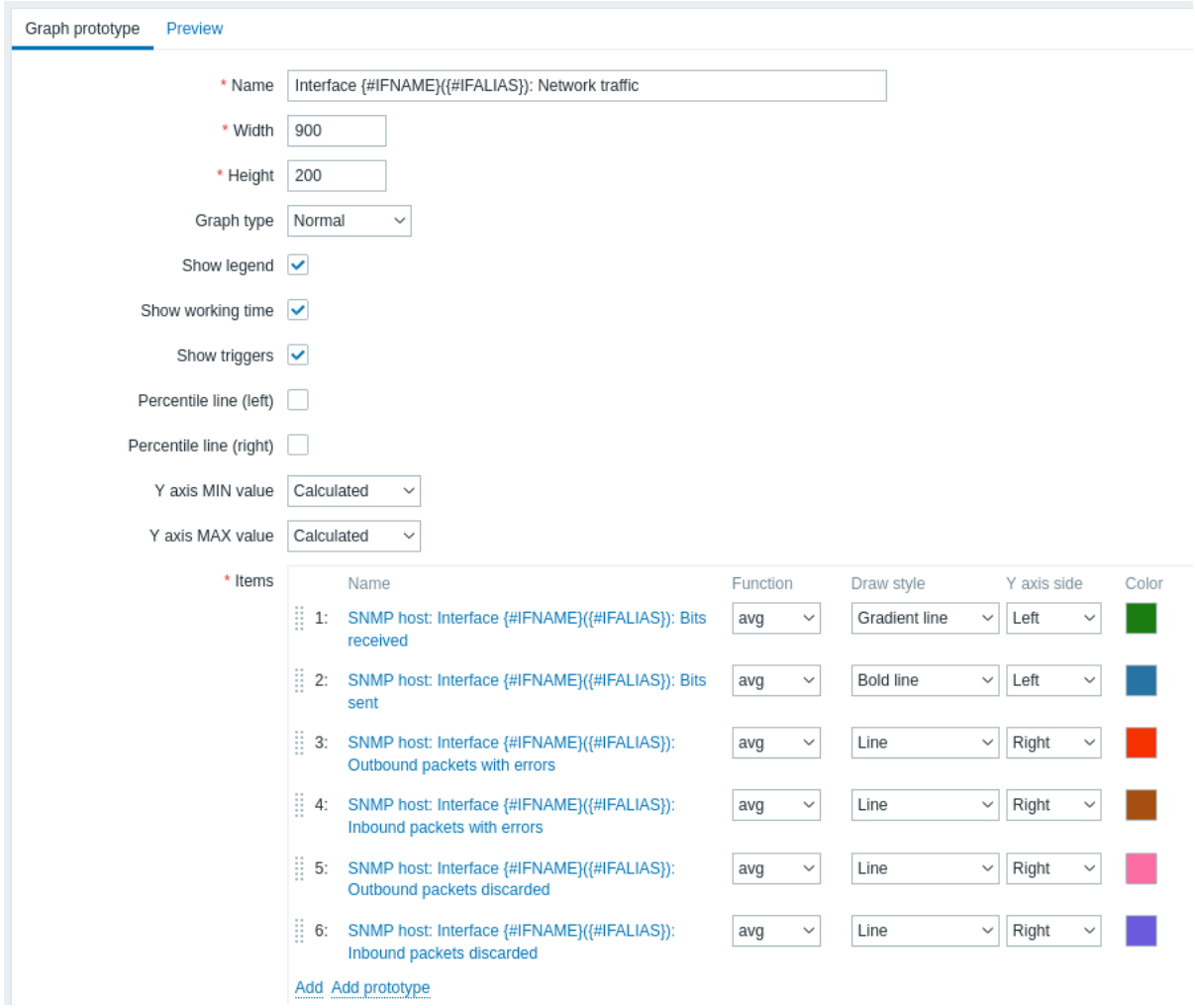
Следующий снимок экрана иллюстрирует, как можно использовать эти макросы в прототипах триггеров:

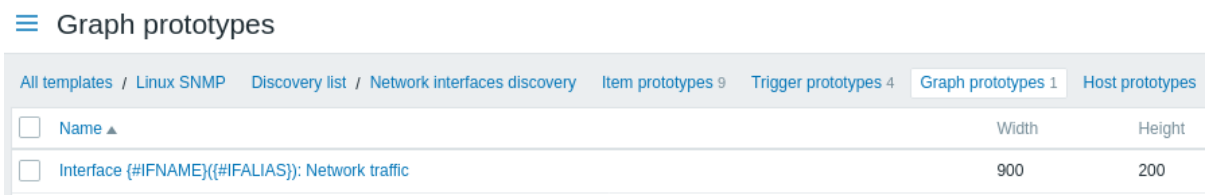




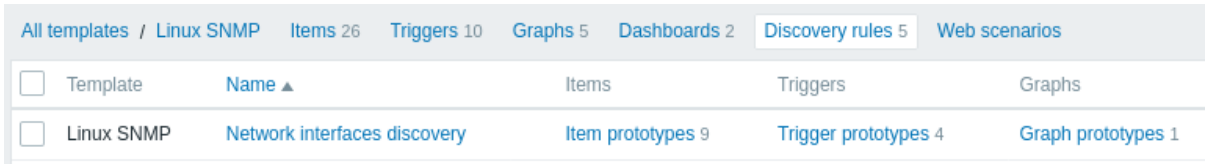
Прототипы графиков

Следующий снимок экрана иллюстрирует, как мы можем использовать эти макросы в прототипах графиков:





Сводка по нашему правилу обнаружения:



Обнаруженные сущности

Когда сервер запущен, он создаёт реальные элементы данных, триггеры и графики на основе значений, которые возвращает правило обнаружения SNMP. В конфигурации узла сети они помечены оранжевой ссылкой на правило обнаружения, из которого они были созданы.

Items

All hosts / SNMP host Enabled SNMP Items 81 Triggers 23 Graphs 14 Discovery rules 6 Web scenarios

Name ▲	Triggers	Key	Interval	History	Trends	Type	Status
Network interfaces discovery: Interface enp4s0(): Bits received	Triggers 1	net.if.in[ifHCInOctets.2]	3m	7d	365d	SNMP agent	Enabled
Network interfaces discovery: Interface enp4s0(): Bits sent	Triggers 1	net.if.out[ifHCOutOctets.2]	3m	7d	365d	SNMP agent	Enabled
Network interfaces discovery: Interface enp4s0(): Inbound packets discarded		net.if.in.discards[ifInDiscards.2]	3m	7d	365d	SNMP agent	Enabled
Network interfaces discovery: Interface enp4s0(): Inbound packets with errors	Triggers 1	net.if.in.errors[ifInErrors.2]	3m	7d	365d	SNMP agent	Enabled
Network interfaces discovery: Interface enp4s0(): Interface type	Triggers 1	net.if.type[ifType.2]	1h	7d	0d	SNMP agent	Enabled
Network interfaces discovery: Interface enp4s0(): Operational status	Triggers 2	net.if.status[ifOperStatus.2]	1m	7d	0	SNMP agent	Enabled
Network interfaces discovery: Interface enp4s0(): Outbound packets discarded		net.if.out.discards[ifOutDiscards.2]	3m	7d	365d	SNMP agent	Enabled
Network interfaces discovery: Interface enp4s0(): Outbound packets with errors	Triggers 1	net.if.out.errors[ifOutErrors.2]	3m	7d	365d	SNMP agent	Enabled
Network interfaces discovery: Interface enp4s0(): Speed	Triggers 2	net.if.speed[ifHighSpeed.2]	5m	7d	0d	SNMP agent	Enabled

Triggers

All hosts / SNMP host Enabled SNMP Items 81 Triggers 23 Graphs 14 Discovery rules 6 Web scenarios

Severity	Value	Name ▲	Operational data	Expression
Information	OK	Network interfaces discovery: Interface enp4s0(): Ethernet has changed to lower speed than it was before **Depends on:** SNMP host: Interface enp4s0(): Link down	Current reported speed: {ITEM.LASTVALUE1}	Problem: **change(**/SNMP host/net.if.speed[ifHighSpeed.2]**)**<0 and **last(**/SNMP host/net.if.speed[ifHighSpeed.2]**)**>0 and (**last(**/SNMP host/net.if.type[ifType.2]**)**=6 or **last(**/SNMP host/net.if.type[ifType.2]**)**=7 or **last(**/SNMP host/net.if.type[ifType.2]**)**=11 or **last(**/SNMP host/net.if.type[ifType.2]**)**=62 or **last(**/SNMP host/net.if.type[ifType.2]**)**=69 or **last(**/SNMP host/net.if.type[ifType.2]**)**=117) and (**last(**/SNMP host/net.if.status[ifOperStatus.2]**)**<>2) Recovery: (**change(**/SNMP host/net.if.speed[ifHighSpeed.2]**)**>0 and **last(**/SNMP host/net.if.speed[ifHighSpeed.2],#2**)**>0) or (**last(**/SNMP host/net.if.status[ifOperStatus.2]**)**=2)
Warning	OK	Network interfaces discovery: Interface enp4s0(): High bandwidth usage **Depends on:** SNMP host: Interface enp4s0(): Link down	In: {ITEM.LASTVALUE1}, out: {ITEM.LASTVALUE3}, speed: {ITEM.LASTVALUE2}	Problem: (**avg(**/SNMP host/net.if.in[ifHCInOctets.2],15m**)**>({$IF.UTIL.MAX:"enp4s0"}/100)***last(**/SNMP host/net.if.speed[ifHighSpeed.2]**)**) or **avg(**/SNMP host/net.if.out[ifHCOutOctets.2],15m**)**>({$IF.UTIL.MAX:"enp4s0"}/100)***last(**/SNMP host/net.if.speed[ifHighSpeed.2]**)**) and **last(**/SNMP host/net.if.speed[ifHighSpeed.2]**)**>0 Recovery: **avg(**/SNMP host/net.if.in[ifHCInOctets.2],15m**)**<(({$IF.UTIL.MAX:"enp4s0"}-3)/100)***last(**/SNMP host/net.if.speed[ifHighSpeed.2]**)** and **avg(**/SNMP host/net.if.out[ifHCOutOctets.2],15m**)**<(({$IF.UTIL.MAX:"enp4s0"}-3)/100)***last(**/SNMP host/net.if.speed[ifHighSpeed.2]**)**
Warning	OK	Network interfaces discovery: Interface enp4s0(): High error rate **Depends on:** SNMP host: Interface enp4s0(): Link down	errors in: {ITEM.LASTVALUE1}, errors out: {ITEM.LASTVALUE2}	Problem: **min(**/SNMP host/net.if.in.errors[ifInErrors.2],5m**)**>{$IF.ERRORS.WARN:"enp4s0"} or **min(**/SNMP host/net.if.out.errors[ifOutErrors.2],5m**)**>{$IF.ERRORS.WARN:"enp4s0"} Recovery: **max(**/SNMP host/net.if.in.errors[ifInErrors.2],5m**)**<{$IF.ERRORS.WARN:"enp4s0"}*0.8 and **max(**/SNMP host/net.if.out.errors[ifOutErrors.2],5m**)**<{$IF.ERRORS.WARN:"enp4s0"}*0.8
Average	OK	Network interfaces discovery: Interface enp4s0(): Link down	Current state: {ITEM.LASTVALUE1}	Problem: {$IFCONTROL:"enp4s0"}=1 and **last(**/SNMP host/net.if.status[ifOperStatus.2]**)**=2 and (**last(**/SNMP host/net.if.status[ifOperStatus.2],#1**)**<>**last(**/SNMP host/net.if.status[ifOperStatus.2],#2**)**) Recovery: **last(**/SNMP host/net.if.status[ifOperStatus.2]**)**<>2 or {$IFCONTROL:"enp4s0"}=0

Graphs

All hosts / SNMP host Enabled SNMP Items 81 Triggers 23 Graphs 14 Discovery rules 6 Web scenarios

Name ▲
Mounted filesystem discovery: /: Disk space usage
Linux SNMP: CPU jumps
CPU discovery: CPU usage
CPU discovery: CPU utilization
Network interfaces discovery: Interface enp4s0(): Network traffic

6 Обнаружение объектов JMX

Обзор

Можно обнаруживать все JMX MBeans или атрибуты MBean, либо указать шаблон для обнаружения этих объектов.

Для настройки правила обнаружения необходимо понимать разницу между MBean и атрибутами MBean. MBean — это объект, который может представлять устройство, приложение или любой ресурс, которым необходимо управлять.

Например, существует MBean, представляющий веб-сервер. Его атрибуты — это количество соединений, количество потоков, тайм-аут запросов, HTTP-кэш файлов, использование памяти и т. д. Если выразить эту мысль на понятном человеку языке, можно определить кофемашину как MBean, который имеет следующие атрибуты для мониторинга: количество воды на чашку, средний расход воды за определённый период времени, количество кофейных зёрен, необходимое на чашку, время пополнения кофейных зёрен и воды и т. д.

Ключ элемента данных

В настройках правила обнаружения выберите **JMX agent** в поле *Тип*.

Для обнаружения JMX-объектов поддерживаются два ключа элементов данных — jmx.discovery[] и jmx.get[]:

Ключ элемента данных	Возвращаемое значение	Параметры	Комментарий
jmx.discovery[<режим обнаружения>,<имя объекта>,<уникальное краткое описание>]	Этот элемент данных возвращает JSON-массив с LLD-макросами, описывающими объекты MBean или их атрибуты.	**режим обнаружения** — одно из следующих значений: *attributes* (получение атрибутов JMX MBean, по умолчанию) или *beans* (получение JMX MBeans) **имя объекта** — шаблон имени объекта (см. documentation), определяющий имена MBean, которые необходимо получить (по умолчанию пусто, что означает получение всех зарегистрированных beans) **уникальное краткое описание** — уникальное описание, позволяющее использовать несколько JMX-элементов данных с одинаковыми режимом обнаружения и именем объекта на узле сети (необязательно)	Примеры: → jmx.discovery — получить все атрибуты JMX MBean → jmx.discovery[beans] — получить все JMX MBeans → jmx.discovery[attributes,"*:type=GarbageCollector,name — получить все атрибуты сборщика мусора → jmx.discovery[beans,"*:type=GarbageCollector,name=* — получить все сборщики мусора Существуют некоторые ограничения на то, какие свойства MBean может возвращать этот элемент данных, из-за ограниченного набора символов, поддерживаемых при генерации имен макросов (поддерживаемые символы могут быть описаны следующим регулярным выражением: `A-Z0-9_\.`). Поэтому, например, для обнаружения свойств MBean, содержащих слова с дефисом или не-ASCII символы, необходимо использовать `jmx.get[]`.
jmx.get[<режим обнаружения>,<имя объекта>,<уникальное краткое описание>]			

Ключ элемента данных			
	Этот элемент данных возвращает JSON-массив с объектами MBean или их атрибутами. По сравнению с `jmx.discovery[]` он не определяет LLD-макросы.	**режим обнаружения** — одно из следующих значений: *attributes* (получение атрибутов JMX MBean, по умолчанию) или *beans* (получение JMX MBeans) **имя объекта** — шаблон имени объекта (см. documentation), определяющий имена MBean, которые необходимо получить (по умолчанию пусто, что означает получение всех зарегистрированных beans) **уникальное краткое описание** — уникальное описание, позволяющее использовать несколько JMX-элементов данных с одинаковыми режимом обнаружения и именем объекта на узле сети (необязательно)	При использовании этого элемента данных необходимо определить пользовательские макросы низкоуровневого обнаружения, указывающие на значения, извлеченные из возвращаемого JSON с помощью JSONPath.

Attention:
Если параметры не переданы, запрашиваются все атрибуты MBean из JMX. Отсутствие параметров для JMX-обнаружения или попытка получить все атрибуты для широкого диапазона, такого как `*:type=*,name=*`, может привести к потенциальным проблемам с производительностью.

Использование jmx.discovery

Этот элемент данных возвращает объект JSON с макросами низкоуровневого обнаружения, описывающими объекты или атрибуты MBean. Например, при обнаружении атрибутов MBean (переформатировано для наглядности):

```
[
    {
        "{#JMXVALUE}":"0",
        "{#JMXTYPE}":"java.lang.Long",
        "{#JMXOBJ}":"java.lang:type=GarbageCollector,name=PS Scavenge",
        "{#JMXDESC}":"java.lang:type=GarbageCollector,name=PS Scavenge,CollectionCount",
        "{#JMXATTR}":"CollectionCount"
    },
    {
        "{#JMXVALUE}":"0",
        "{#JMXTYPE}":"java.lang.Long",
        "{#JMXOBJ}":"java.lang:type=GarbageCollector,name=PS Scavenge",
        "{#JMXDESC}":"java.lang:type=GarbageCollector,name=PS Scavenge,CollectionTime",
        "{#JMXATTR}":"CollectionTime"
    },
    {
        "{#JMXVALUE}":"true",
        "{#JMXTYPE}":"java.lang.Boolean",
        "{#JMXOBJ}":"java.lang:type=GarbageCollector,name=PS Scavenge",
        "{#JMXDESC}":"java.lang:type=GarbageCollector,name=PS Scavenge,Valid",
        "{#JMXATTR}":"Valid"
    },
    {
        "{#JMXVALUE}":"PS Scavenge",
        "{#JMXTYPE}":"java.lang.String",
        "{#JMXOBJ}":"java.lang:type=GarbageCollector,name=PS Scavenge",
```

```
        "{#JMXDESC}":"java.lang:type=GarbageCollector,name=PS Scavenge,Name",
        "{#JMXATTR}":"Name"
    },
    {
        "{#JMXVALUE}":"java.lang:type=GarbageCollector,name=PS Scavenge",
        "{#JMXTYPE}":"javax.management.ObjectName",
        "{#JMXOBJ}":"java.lang:type=GarbageCollector,name=PS Scavenge",
        "{#JMXDESC}":"java.lang:type=GarbageCollector,name=PS Scavenge,ObjectName",
        "{#JMXATTR}":"ObjectName"
    }
]
```

При обнаружении MBean (переформатировано для наглядности):

```
[
    {
        "{#JMXDOMAIN}":"java.lang",
        "{#JMXTYPE}":"GarbageCollector",
        "{#JMXOBJ}":"java.lang:type=GarbageCollector,name=PS Scavenge",
        "{#JMXNAME}":"PS Scavenge"
    }
]
```

Поддерживаемые макросы

Следующие макросы поддерживаются для использования в фильтре правила обнаружения и прототипах элементов данных, триггеров и графиков:

Макрос	Описание
Обнаружение атрибутов MBean	
{#JMXVALUE}	Значение атрибута.
{#JMXTYPE}	Тип атрибута.
{#JMXOBJ}	Имя объекта.
{#JMXDESC}	Имя объекта, включая имя атрибута.
{#JMXATTR}	Имя атрибута.
Обнаружение MBean	
{#JMXDOMAIN}	Домен MBean. (*Зарезервированное имя Zabbix*)
{#JMXOBJ}	Имя объекта. (*Зарезервированное имя Zabbix*)
{#JMX<key property>}	Свойства MBean (например, {#JMXTYPE}, {#JMXNAME}) (см. Ограничения ниже).

Ограничения

Существуют некоторые ограничения, связанные с алгоритмом создания имен LLD-макросов из имен свойств MBean:

- имена атрибутов преобразуются в верхний регистр
- имена атрибутов игнорируются (LLD-макросы не создаются), если они состоят из символов, не поддерживаемых в именах LLD-макросов. Поддерживаемые символы могут быть описаны следующим регулярным выражением: `A-Z0-9_\.`
- если атрибут называется ”obj” или ”domain”, он будет проигнорирован из-за пересечения со значениями зарезервированных свойств Zabbix {#JMXOBJ} и {#JMXDOMAIN}

Рассмотрим следующий пример jmx.discovery (с режимом ”beans”). Для MBean определены следующие свойства (некоторые из них будут проигнорированы; см. ниже):

```
name=test
  =Type
attributes []=1,2,3
Name=NameOfTheTest
domAin=some
```

В результате JMX-обнаружения будут сгенерированы следующие LLD-макросы:

- {#JMXDOMAIN} - внутренний макрос Zabbix, описывающий домен MBean
- {#JMXOBJ} - внутренний макрос Zabbix, описывающий объект MBean
- {#JMXNAME} - создан из свойства ”name”

Игнорируемые свойства:

- тип : его имя содержит неподдерживаемые символы (не-ASCII)
- attributes[] : его имя содержит неподдерживаемые символы (квадратные скобки не поддерживаются)
- Name : оно уже определено (name=test)
- domAin : это зарезервированное имя Zabbix

Примеры

Давайте рассмотрим еще два практических примера создания правила LLD с использованием MBean. Чтобы лучше понять разницу между правилом LLD, собирающим MBean, и правилом LLD, собирающим атрибуты MBean, пожалуйста, взгляните на следующую таблицу:

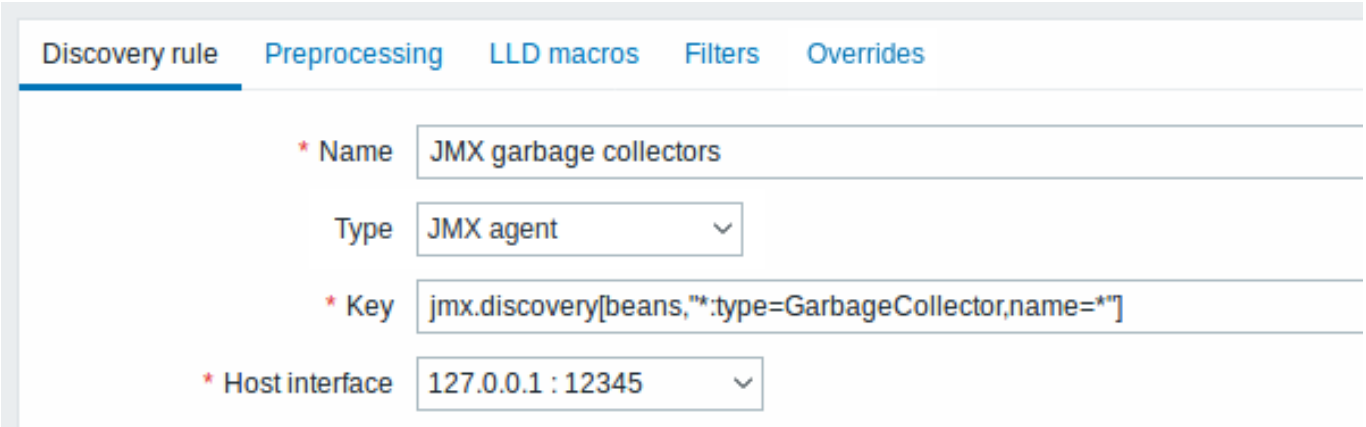
MBean1	**MBean2**	**MBean3**
MBean1Attribute1	MBean2Attribute1	MBean3Attribute1
MBean1Attribute2	MBean2Attribute2	MBean3Attribute2
MBean1Attribute3	MBean2Attribute3	MBean3Attribute3

Пример 1: Обнаружение MBeans

Это правило вернет 3 объекта: верхнюю строку столбца: MBean1, MBean2, MBean3.

Для получения дополнительной информации об объектах см. таблицу поддерживаемых макросов, раздел *Обнаружение MBeans*.

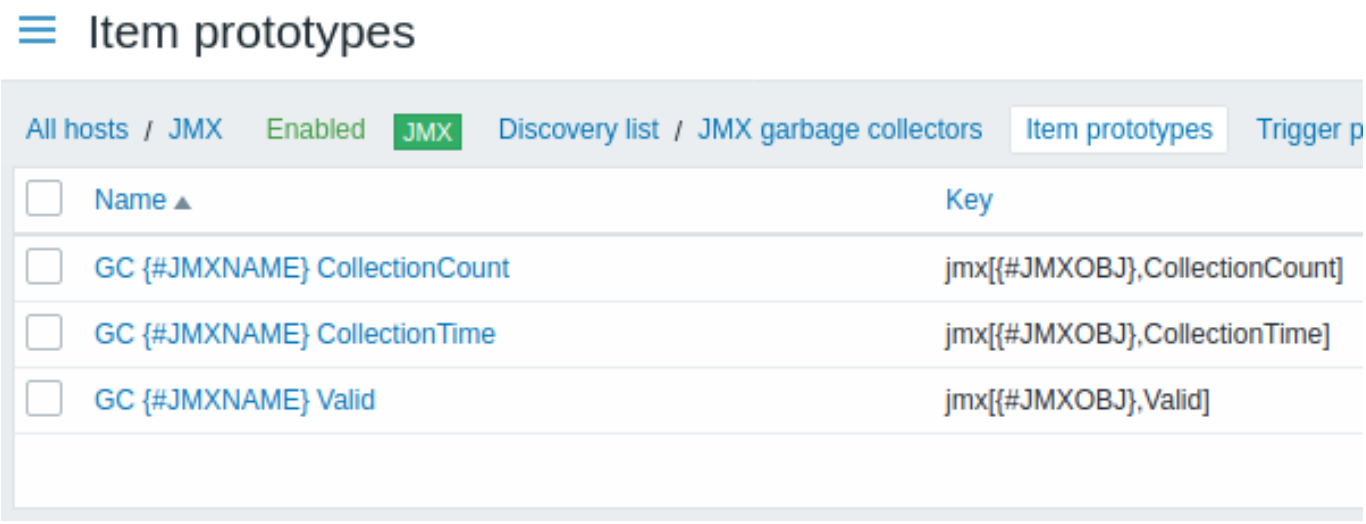
Настройка правила обнаружения, собирающего MBeans (без атрибутов), выглядит следующим образом:



Используемый здесь ключ:

```
jmx.discovery[beans,"*:type=GarbageCollector,name=*"]
```

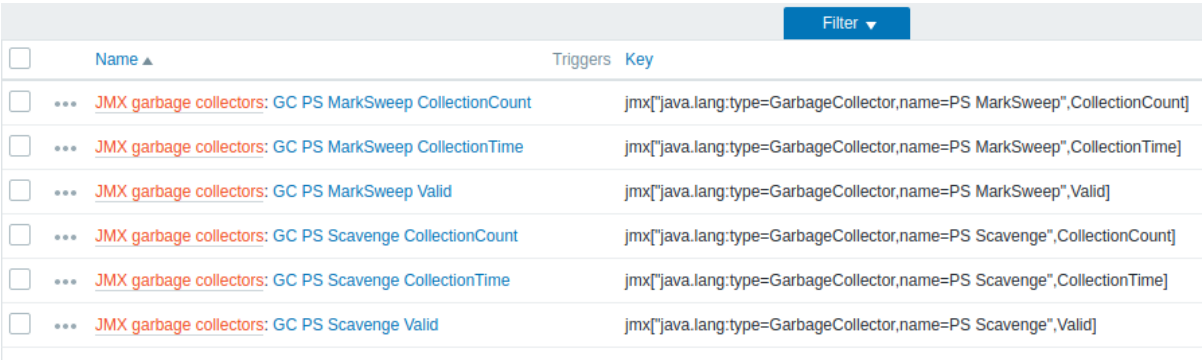
Будут обнаружены все сборщики мусора без атрибутов. Поскольку сборщики мусора имеют одинаковый набор атрибутов, мы можем использовать нужные атрибуты в прототипах элементов данных следующим образом:



Используемые здесь ключи:

```
jmx[{#JMXOBJ},CollectionCount]
jmx[{#JMXOBJ},CollectionTime]
jmx[{#JMXOBJ},Valid]
```

Правило обнаружения LLD в результате даст что-то близкое к следующему (элементы данных обнаружены для двух сборщиков мусора):

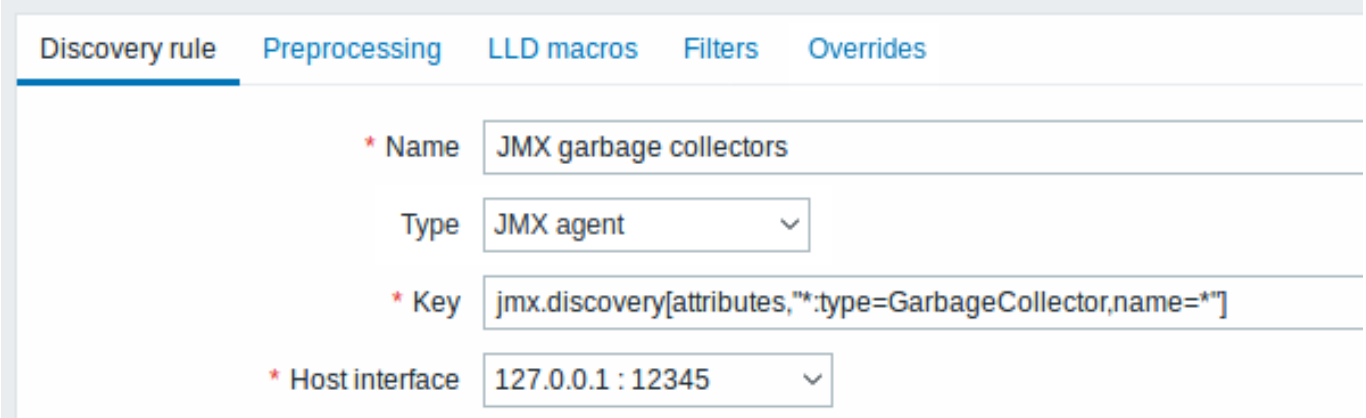


Пример 2: Обнаружение атрибутов MBean

Это правило вернёт 9 объектов со следующими полями: MBean1Attribute1, MBean2Attribute1, MBean3Attribute1,MBean1Attribute2,MBea MBean3Attribute2, MBean1Attribute3, MBean2Attribute3, MBean3Attribute3.

Для получения дополнительной информации об объектах см. таблицу поддерживаемых макросов, раздел *Обнаружение атрибутов MBean*.

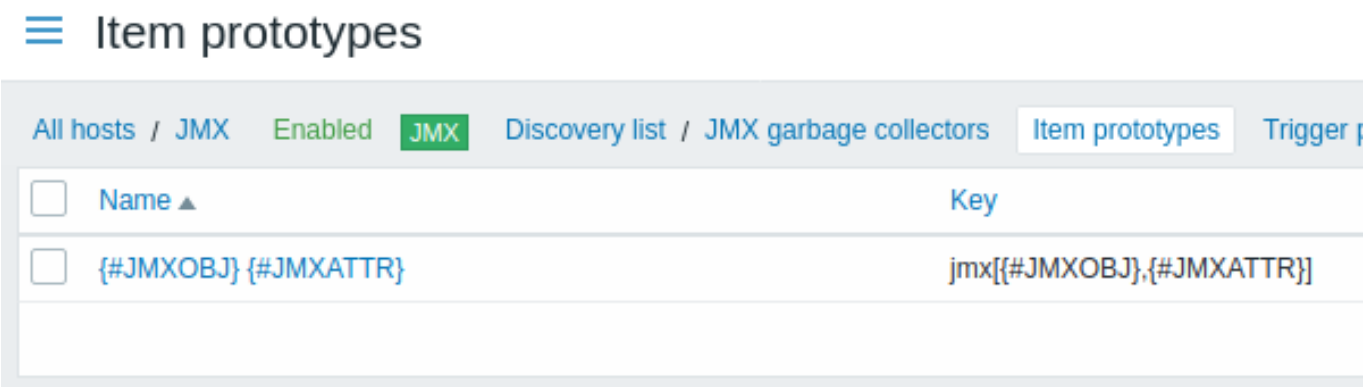
Конфигурация правила обнаружения, собирающего атрибуты MBean, выглядит следующим образом:



Используемый здесь ключ:

```
jmx.discovery[attributes,"*:type=GarbageCollector,name=*"]
```

Будут обнаружены все сборщики мусора с атрибутом, состоящим из одного элемента данных.



В данном случае элемент данных будет создан из прототипа для каждого атрибута MBean. Основной недостаток этой конфигурации заключается в том, что создание триггеров из прототипов триггеров невозможно, так как для всех атрибутов имеется только один прототип элемента данных. Поэтому такую настройку можно использовать для сбора данных, но она не рекомендуется для автоматического мониторинга.

Использование jmx.get

`jmx.get[]` похож на элемент данных `jmx.discovery[]`, но не преобразует свойства объектов Java в имена макросов низкоуровневого обнаружения и, следовательно, может возвращать значения без ограничений, связанных с генерацией имен макросов LLD, таких как дефисы или символы не ASCII.

При использовании `jmx.get[]` для обнаружения макросы низкоуровневого обнаружения можно определить отдельно на пользовательской вкладке LLD macro в конфигурации правила обнаружения, используя JSONPath для указания требуемых значений.

Обнаружение MBean

Элемент данных обнаружения: `jmx.get[beans,"com.example:type=*,*"]`

Ответ:

```
[
    {
        "object": "com.example:type=Hello,data-src=data-base,  =    ",
        "domain": "com.example",
        "properties": {
            "data-src": "data-base",
            "  ": "    ",
            "type": "Hello"
        }
    },
    {
        "object": "com.example:type=Atomic",
        "domain": "com.example",
        "properties": {
            "type": "Atomic"
        }
    }
]
```

Обнаружение атрибутов MBean

Элемент данных обнаружения: `jmx.get[attributes,"com.example:type=*,*"]`

Ответ:

```
[
    {
        "object": "com.example:type=*",
        "domain": "com.example",
        "properties": {
            "type": "Simple"
        }
    },
    {
        "object": "com.zabbix:type=yes,domain=zabbix.com,data-source=/dev/rand,  =    ,obj=true",
        "domain": "com.zabbix",
        "properties": {
            "type": "Hello",
            "domain": "com.example",
            "data-source": "/dev/rand",
            "  ": "    ",
            "obj": true
        }
    }
]
```

7 Обнаружение IPMI-сенсоров

Обзор

Можно автоматически обнаруживать IPMI-сенсоры.

Для этого можно использовать сочетание:

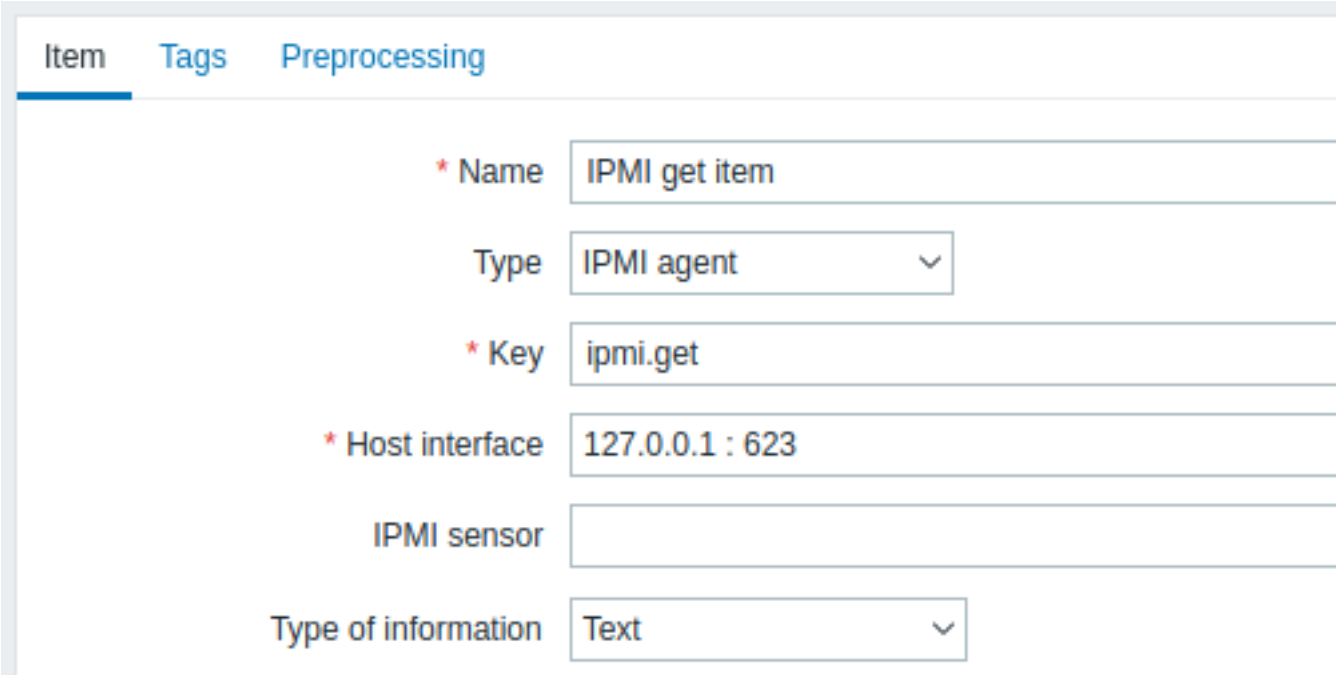
- IPMI-элемента данных `ipmi.get` в качестве мастер-элемента данных
- зависимого правила низкоуровневого обнаружения и прототипов элементов данных

Настройка

Мастер-элемент данных

Создайте IPMI элемент данных, используя следующий ключ:

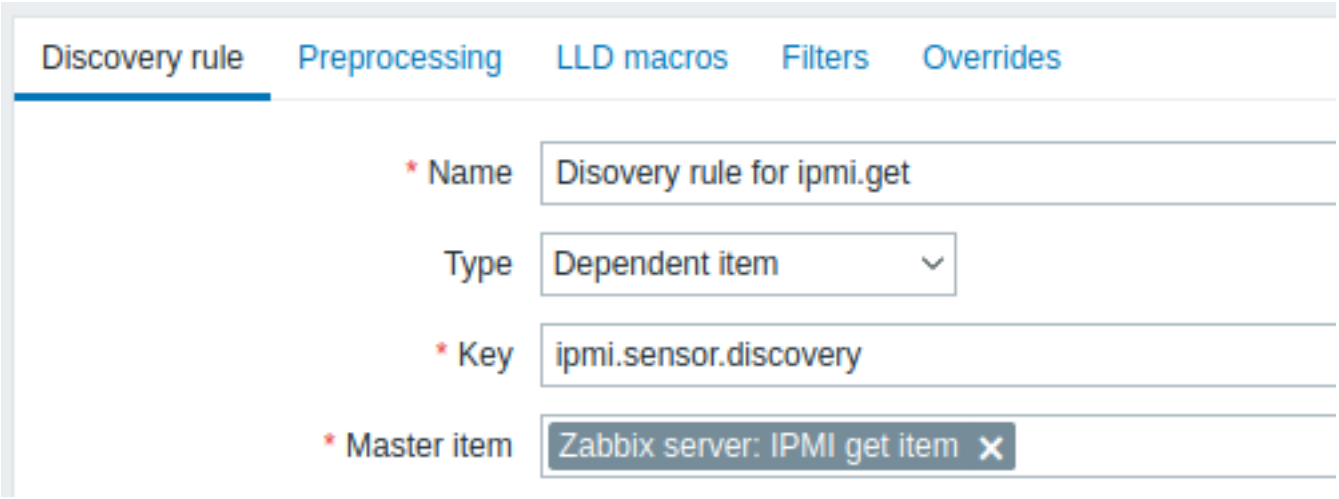
`ipmi.get`



Установите тип информации в значение ”Текст” для потенциально больших данных JSON.

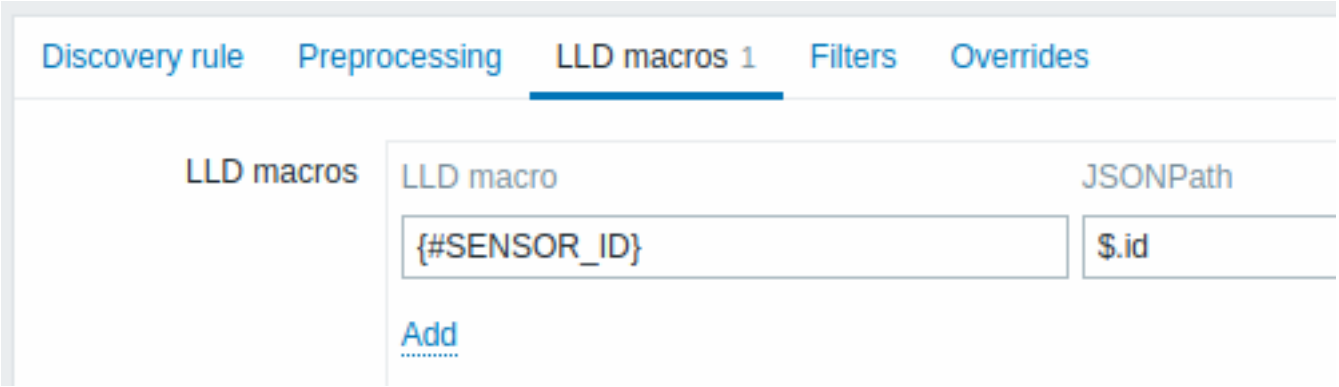
Зависимое правило LLD

Создайте правило низкоуровневого обнаружения типа «Зависимый элемент данных»:



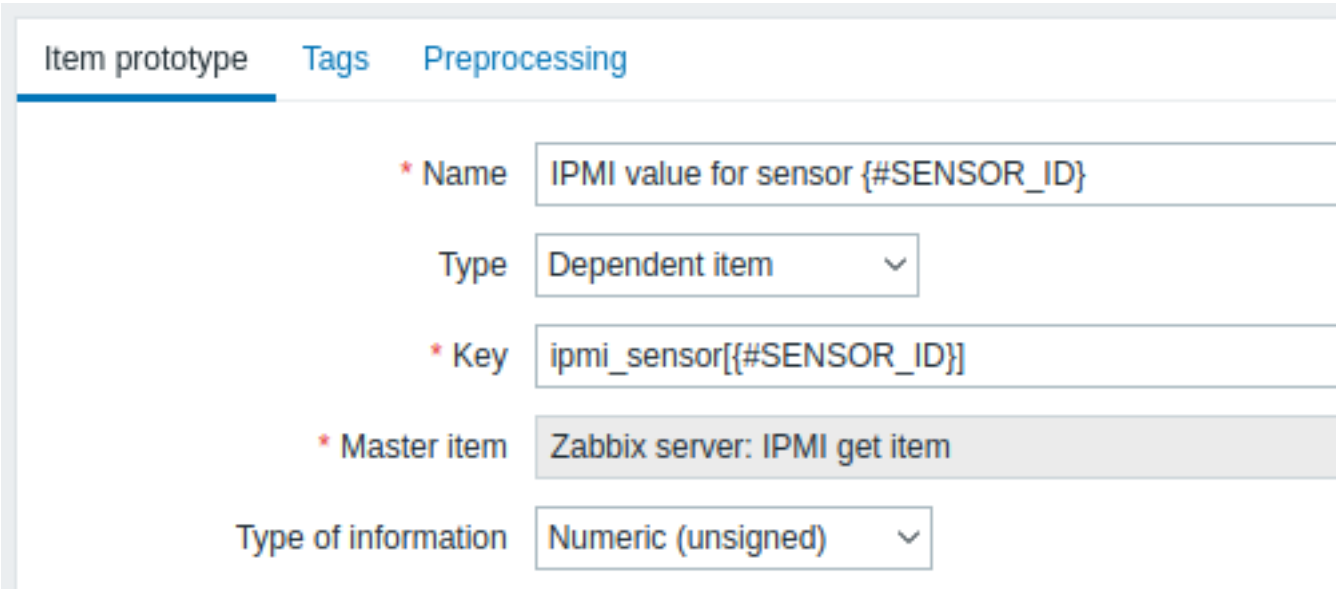
В качестве мастер-элемента данных выберите созданный нами элемент данных `ipmi.get`.

На вкладке «Макросы LLD» определите пользовательский макрос с соответствующим JSONPath:



Прототип зависимого элемента данных

Создайте в этом правиле LLD прототип элемента данных с типом "Зависимый элемент данных". В качестве мастер-элемента данных для этого прототипа выберите созданный нами элемент данных `ipmi.get`.



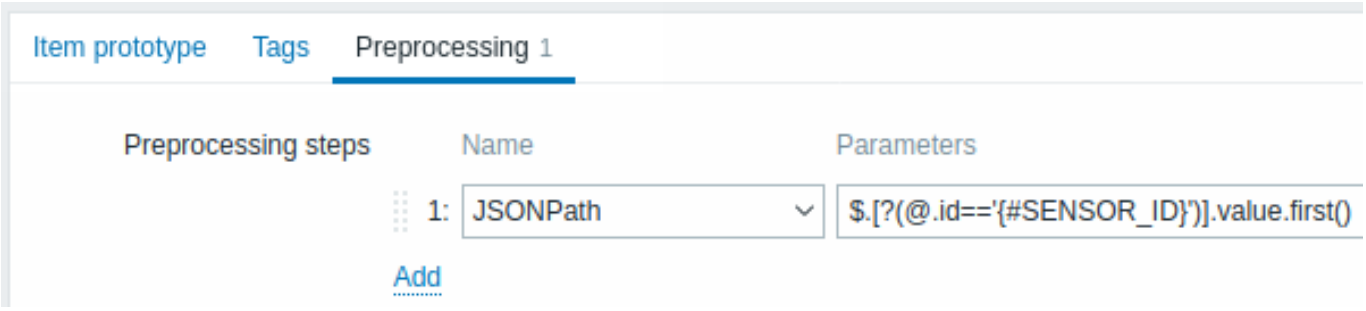
Обратите внимание на использование макроса {#SENSOR_ID} в имени и ключе прототипа элемента данных:

- *Имя*: Значение IPMI для датчика {#SENSOR_ID}
- *Ключ*: ipmi_sensor[{#SENSOR_ID}]

В качестве типа информации укажите *Числовой (без знака)*.

На вкладке "Предобработка" прототипа элемента данных выберите JSONPath и используйте следующее выражение JSONPath в качестве параметра:

```
$.[?(@.id=='{#SENSOR_ID}')].value.first()
```



Когда обнаружение начнется, будет создан один элемент данных для каждого датчика IPMI. Этот элемент данных будет возвращать целочисленное значение указанного датчика.

8 Обнаружение сервисов systemd

Обзор

Можно обнаружить systemd службы (службы, по умолчанию) используя Zabbix.

Ключ элемента данных

Элемент данных, который следует использовать в правиле обнаружения, это

`systemd.unit.discovery`

> **Attention:**
> Этот ключ элемента данных поддерживается только в Zabbix агент 2.

Этот элемент данных возвращает JSON с информацией о модулях systemd, например:

```
[{
    "{#UNIT.NAME}": "mysqld.service",
    "{#UNIT.DESCRIPTION}": "MySQL Server",
    "{#UNIT.LOADSTATE}": "loaded",
    "{#UNIT.ACTIVESTATE}": "active",
    "{#UNIT.SUBSTATE}": "running",
    "{#UNIT.FOLLOWED}": "",
    "{#UNIT.PATH}": "/org/freedesktop/systemd1/unit/mysqld_2eservice",
    "{#UNIT.JOBID}": 0,
    "{#UNIT.JOBTYPE}": "",
    "{#UNIT.JOBPATH}": "/",
    "{#UNIT.UNITFILESTATE}": "enabled"
    "{#UNIT.SERVICETYPE}": "simple"
}, {
    "{#UNIT.NAME}": "systemd-journald.socket",
    "{#UNIT.DESCRIPTION}": "Journal Socket",
    "{#UNIT.LOADSTATE}": "loaded",
    "{#UNIT.ACTIVESTATE}": "active",
    "{#UNIT.SUBSTATE}": "running",
    "{#UNIT.FOLLOWED}": "",
    "{#UNIT.PATH}": "/org/freedesktop/systemd1/unit/systemd_2djournald_2esocket",
    "{#UNIT.JOBID}": 0,
    "{#UNIT.JOBTYPE}": "",
    "{#UNIT.JOBPATH}": "/",
    "{#UNIT.UNITFILESTATE}": "enabled"
}]
```

Обнаружение отключённых systemd units

Также возможно обнаруживать **отключённые** systemd units. В этом случае в результирующем JSON возвращаются три макроса:

- {#UNIT.PATH}
- {#UNIT.ACTIVESTATE}
- {#UNIT.UNITFILESTATE}.

> **Attention:**
> Чтобы для отключённых systemd units были созданы элементы данных и триггеры из прототипов, убедитесь, что вы скорректировали (или удалили) запрещающие LLD-фильтры для {#UNIT.ACTIVESTATE} и {#UNIT.UNITFILESTATE}.

Поддерживаемые макросы

Следующие макросы поддерживаются для использования в фильтре правила обнаружения и прототипах элементов данных, триггеров и графиков:

Macro	Description
{#UNIT.NAME}	Основное имя юнита.
{#UNIT.DESCRIPTION}	Понятное человеку описание.
{#UNIT.LOADSTATE}	Состояние загрузки (то есть, был ли файл юнита успешно загружен).
{#UNIT.ACTIVESTATE}	Активное состояние (то есть, запущен ли юнит в данный момент).
{#UNIT.SUBSTATE}	Подсостояние (более детализированная версия активного состояния, специфичная для типа юнита, в отличие от самого активного состояния).

Macro	Description
{#UNIT.FOLLOWED}	Юнит, за состоянием которого следует данный юнит, если такой есть; в противном случае — пустая строка.
{#UNIT.PATH}	Путь к объекту юнита.
{#UNIT.JOBID}	Числовой ID задания, если для юнита задания поставлено задание в очередь; в противном случае 0.
{#UNIT.JOBTYPE}	Тип задания.
{#UNIT.JOBPATH}	Путь к объекту задания.
{#UNIT.UNITFILESTATE}	Состояние установки файла юнита.
{#UNIT.SERVICETYPE}	Тип сервисного юнита (например, `simple`, `forking`, `oneshot`, `idle` и т. д.). Этот макрос возвращается только в том случае, если юнит является сервисом.

Прототипы элементов данных

Прототипы элементов данных, которые можно создать на основе обнаружения сервисов systemd, включают, например:

- Имя элемента данных: `{#UNIT.DESCRIPTION} active state info`; ключ элемента данных: `systemd.unit.info["{#UNIT`
- Имя элемента данных: `{#UNIT.DESCRIPTION} load state info`; ключ элемента данных: `systemd.unit.info["{#UNIT.NA`

9 Обнаружение служб Windows

Обзор

Аналогично тому, как обнаруживаются файловые системы, также можно обнаруживать службы Windows.

Ключ элемента данных

Элемент данных, который следует использовать в правиле обнаружения:

`service.discovery`

Поддерживаемые макросы

Следующие макросы поддерживаются для использования в фильтре правила обнаружения и в прототипах элементов данных, триггеров и графиков:

Макрос	Описание
{#SERVICE.NAME}	Имя службы.
{#SERVICE.DISPLAYNAME}	Отображаемое имя службы.
{#SERVICE.DESCRIPTION}	Описание службы.
{#SERVICE.STATE}	Числовое значение состояния службы. Подробности см. в элементе данных service.info.
{#SERVICE.STATENAME}	Название состояния службы. Подробности см. в элементе данных service.info.
{#SERVICE.PATH}	Путь службы.
{#SERVICE.USER}	Пользователь службы.
{#SERVICE.STARTUP}	Числовое значение типа запуска службы. Подробности см. в элементе данных service.info.
{#SERVICE.STARTUPNAME}	Название типа запуска службы. Подробности см. в элементе данных service.info.
{#SERVICE.STARTUPTRIGGER}	Числовое значение, указывающее, имеет ли тип запуска службы: 0 - триггеры запуска отсутствуют 1 - триггеры запуска присутствуют Это полезно для обнаружения таких типов запуска службы, как *Automatic (trigger start)*, *Automatic delayed (trigger start)* и *Manual (trigger start).*

На основе обнаружения служб Windows вы можете создать прототип элемента данных вида

`service.info[{#SERVICE.NAME},<param>]`

где `param` принимает следующие значения: *state*, *displayname*, *path*, *user*, *startup* или *description*.

Например, чтобы получить отображаемое имя службы, вы можете использовать элемент данных "service.info[{#SERVICE.NAME},displ
Если значение `param` не указано ("service.info[{#SERVICE.NAME}]"), используется параметр *state* по умолчанию.

10 Обнаружение экземпляров счётчиков производительности Windows

Обзор

Можно обнаруживать экземпляры объектов счётчиков производительности Windows. Это полезно для счётчиков производительности с несколькими экземплярами.

Ключ элемента данных

Чтобы настроить правило обнаружения, используйте следующий элемент данных:

- `perf_instance.discovery[object]`

Обратите внимание, что имя объекта может быть локализовано. Например:

```
perf_instance.discovery[Processor] #                    .
perf_instance.discovery[Processador] #                      .
```

В качестве альтернативы, чтобы гарантировать, что имя объекта указывается на английском языке независимо от локализации ОС, используйте следующий элемент данных:

- `perf_instance_en.discovery[object]`

Например:

```
perf_instance_en.discovery[Processor]
perf_instance_en.discovery[Memory]
```

Поддерживаемые макросы

Процесс обнаружения вернет все экземпляры указанного объекта в макросе {#INSTANCE}:

```
[
    {"{#INSTANCE}":"0"},
    {"{#INSTANCE}":"1"},
    {"{#INSTANCE}":"_Total"}
]
```

Этот макрос можно использовать в прототипах элементов данных `perf_counter[]` и `perf_counter_en[]`.

Например, если ключ элемента данных, используемый в правиле обнаружения, — `perf_instance.discovery[Processor]`, вы можете создать следующий прототип элемента данных:

```
perf_counter["\Processor({#INSTANCE})\% Processor Time"]
```

Примечание:

- Если указанный объект не найден или не поддерживает переменные экземпляры, элемент данных обнаружения станет NOTSUPPORTED.
- Если указанный объект поддерживает переменные экземпляры, но в данный момент не имеет экземпляров, будет возвращен пустой массив JSON.
- Повторяющиеся экземпляры будут пропущены.

11 Обнаружение с использованием запросов WMI

Обзор

WMI — это мощный интерфейс в Windows, который можно использовать для получения различной информации о компонентах Windows, службах, состоянии и установленном программном обеспечении.

Его можно использовать для обнаружения физических дисков и сбора данных об их производительности, обнаружения сетевых интерфейсов, обнаружения гостевых систем Hyper-V, мониторинга служб Windows и многих других задач в ОС Windows.

Этот тип низкоуровневого обнаружения выполняется с помощью WQL-запросов, результаты которых автоматически преобразуются в JSON-объект, подходящий для низкоуровневого обнаружения.

Ключ элемента данных

Элемент данных, который следует использовать в правиле обнаружения:

`wmi.getall[<namespace>,<query>]`

Этот элемент данных преобразует результат запроса в JSON-массив. Например:

```
select * from Win32_DiskDrive where Name like '%PHYSICALDRIVE%'
```

может вернуть примерно следующее:

```
[
    {
        "DeviceID" : "\\.\PHYSICALDRIVE0",
        "BytesPerSector" : 512,
        "Capabilities" : [
            3,
            4
        ],
        "CapabilityDescriptions" : [
            "Random Access",
            "Supports Writing"
        ],
        "Caption" : "VBOX HARDDISK ATA Device",
        "ConfigManagerErrorCode" : 0,
        "ConfigManagerUserConfig" : "False",
        "CreationClassName" : "Win32_DiskDrive",
        "Description" : "Disk drive",
        "FirmwareRevision" : "1.0",
        "Index" : 0,
        "InterfaceType" : "IDE"
    },
    {
        "DeviceID" : "\\.\PHYSICALDRIVE1",
        "BytesPerSector" : 512,
        "Capabilities" : [
            3,
            4
        ],
        "CapabilityDescriptions" : [
            "Random Access",
            "Supports Writing"
        ],
        "Caption" : "VBOX HARDDISK ATA Device",
        "ConfigManagerErrorCode" : 0,
        "ConfigManagerUserConfig" : "False",
        "CreationClassName" : "Win32_DiskDrive",
        "Description" : "Disk drive",
        "FirmwareRevision" : "1.0",
        "Index" : 1,
        "InterfaceType" : "IDE"
    }
]
```

Макросы низкоуровневого обнаружения

Несмотря на то, что в возвращаемом JSON не создаются никакие макросы низкоуровневого обнаружения, эти макросы могут быть определены пользователем как дополнительный шаг, используя функциональность пользовательских макросов LLD с JSONPath, указывающим на обнаруженные значения в возвращаемом JSON.

Затем эти макросы могут использоваться для создания прототипов элементов данных, триггеров и т. д.

12 Обнаружение с использованием SQL-запросов ODBC

Обзор

Этот тип обнаружения низкого уровня выполняется с помощью SQL-запросов, результаты которых автоматически преобразуются в объект JSON, подходящий для обнаружения низкого уровня.

Ключ элемента данных

SQL-запросы выполняются с использованием типа элемента данных «Database monitor». Поэтому для получения рабочего правила обнаружения «Database monitor» применимо большинство инструкций со страницы Мониторинг ODBC.

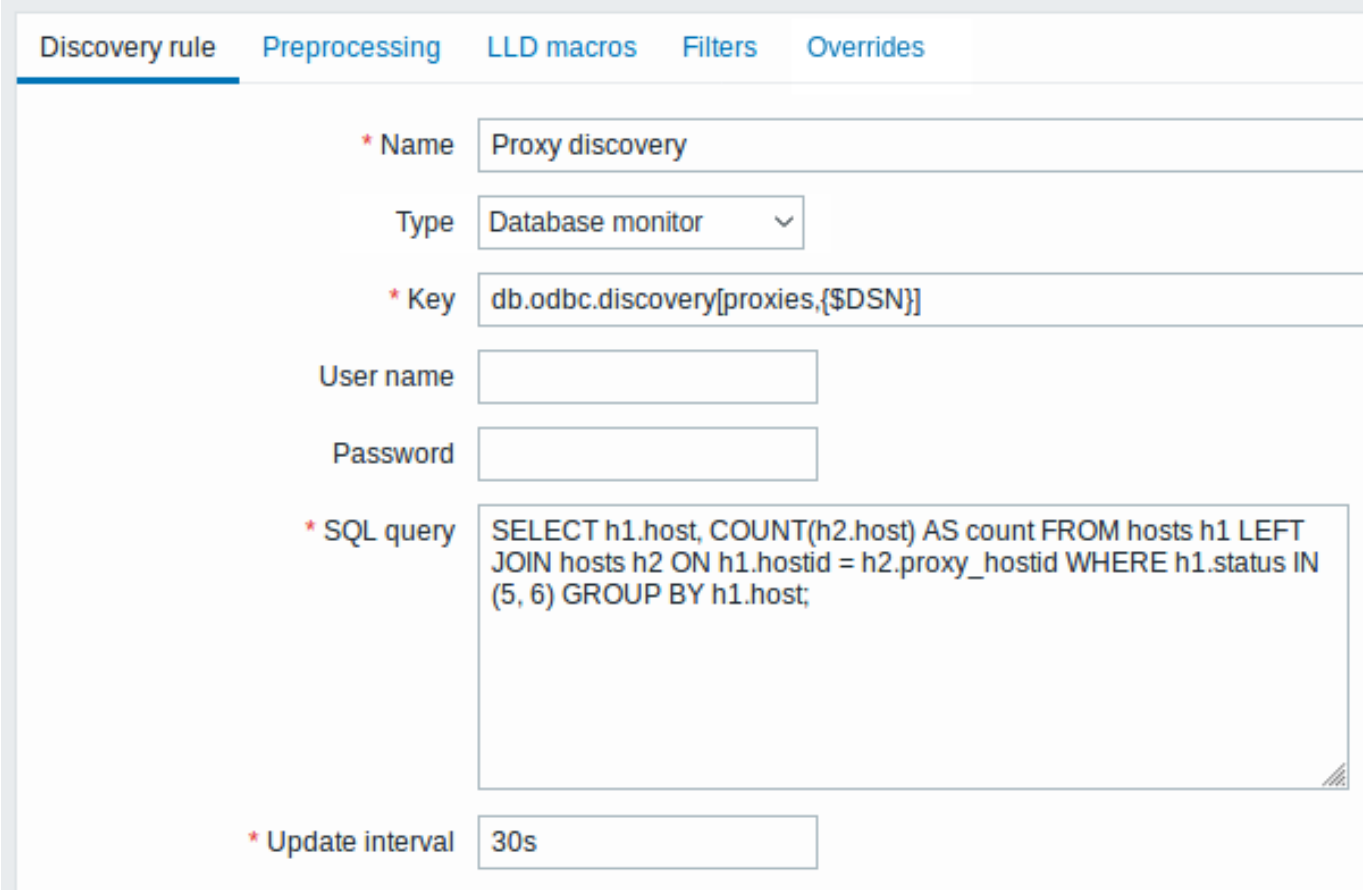
В правилах обнаружения «Database monitor» можно использовать два ключа элемента данных:

- **db.odbc.discovery**[<unique short description>,<dsn>,<connection string>] — этот элемент данных преобразует результат SQL-запроса в JSON-массив, превращая имена столбцов из результата запроса в имена макросов низкоуровневого обнаружения, сопоставленные с обнаруженными значениями полей. Эти макросы можно использовать при создании прототипов элементов данных, триггеров и т. д. См. также: Использование db.odbc.discovery.
- **db.odbc.get**[<unique short description>,<dsn>,<connection string>] — этот элемент данных преобразует результат SQL-запроса в JSON-массив, сохраняя исходные имена столбцов из результата запроса в качестве имен полей в JSON, сопоставленных с обнаруженными значениями. По сравнению с `db.odbc.discovery[]`, этот элемент данных не создает макросы низкоуровневого обнаружения в возвращаемом JSON, поэтому нет необходимости проверять, могут ли имена столбцов быть допустимыми именами макросов. Макросы низкоуровневого обнаружения при необходимости могут быть определены как дополнительный шаг с использованием функциональности пользовательских макросов LLD, где JSONPath указывает на обнаруженные значения в возвращаемом JSON. См. также: Использование db.odbc.get.

Использование db.odbc.discovery

Следующий пример демонстрирует, как SQL-запрос преобразуется в JSON с использованием низкоуровневого обнаружения прокси Zabbix на основе ODBC-запроса к базе данных Zabbix. Это полезно для автоматического создания "zabbix[proxy,<name>,lastaccess]" внутренних элементов данных для мониторинга того, какие прокси активны.

Начните с настройки правила обнаружения:



Все обязательные поля ввода отмечены красной звездочкой.

Здесь используется следующий прямой запрос к базе данных Zabbix для выбора всех прокси Zabbix вместе с количеством узлов сети, которые они мониторят. Количество узлов сети можно использовать, например, чтобы отфильтровать пустые прокси:

```
mysql> SELECT h1.host, COUNT(h2.host) AS count FROM hosts h1 LEFT JOIN hosts h2 ON h1.hostid = h2.proxyid
+---------+-------+
| host    | count |
+---------+-------+
| Japan 1 |     5 |
```

```
| Japan 2 |    12 |
| Latvia  |     3 |
+---------+-------+
3 rows in set (0.01 sec)
```

За счет внутренней работы элемента данных "db.odbc.discovery[,{$DSN}]" результат этого запроса автоматически преобразуется в следующий JSON:

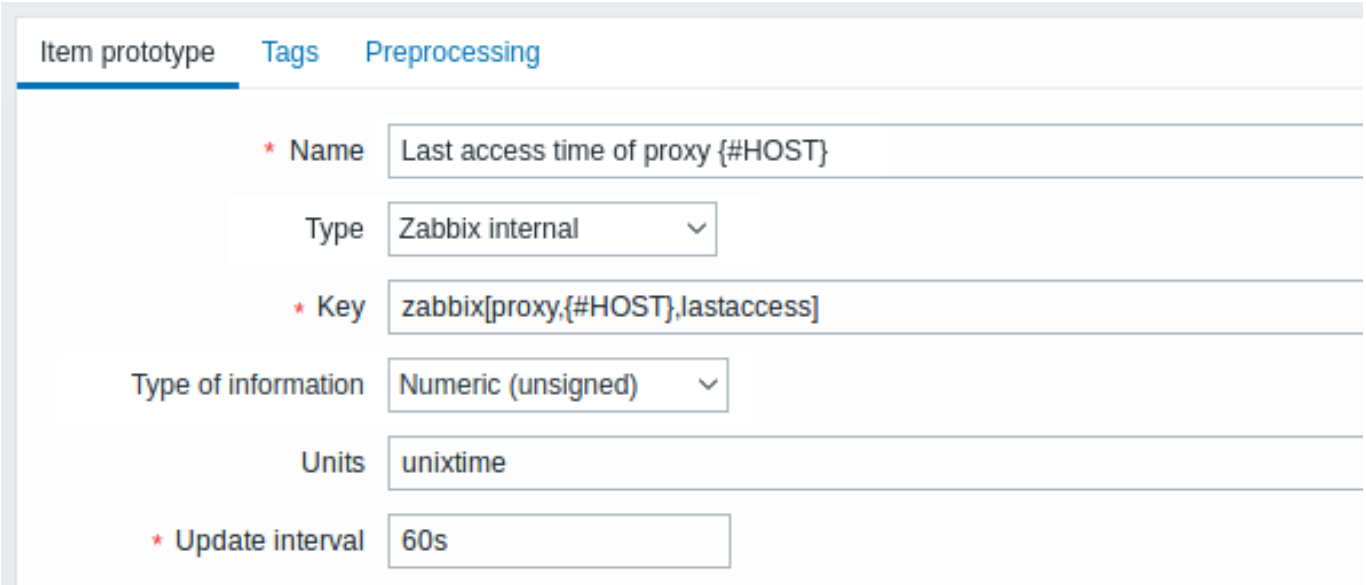
```
[
    {
        "{#HOST}": "Japan 1",
        "{#COUNT}": "5"
    },
    {
        "{#HOST}": "Japan 2",
        "{#COUNT}": "12"
    },
    {
        "{#HOST}": "Latvia",
        "{#COUNT}": "3"
    }
]
```

Как видно, имена столбцов становятся именами макросов, а выбранные строки становятся значениями этих макросов.

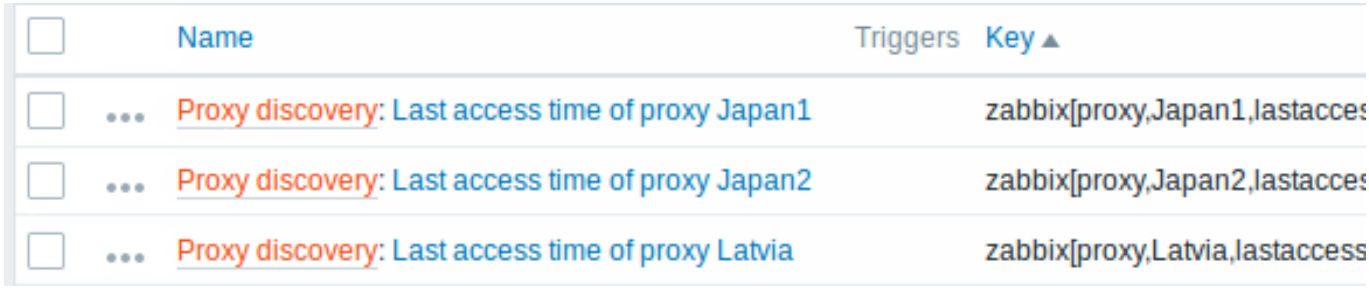
Note:
Если не очевидно, как имя столбца будет преобразовано в имя макроса, рекомендуется использовать псевдонимы столбцов, как "COUNT(h2.host) AS count" в приведенном выше примере.
Если имя столбца не может быть преобразовано в допустимое имя макроса, правило обнаружения становится неподдерживаемым, а сообщение об ошибке содержит номер проблемного столбца. Если требуется дополнительная помощь, полученные имена столбцов выводятся при DebugLevel=4 в файле журнала сервера Zabbix:

```
$ grep db.odbc.discovery /tmp/zabbix_server.log
 ...
 23876:20150114:153410.856 In db_odbc_discovery() query:'SELECT h1.host, COUNT(h2.host) FROM hosts h1 I
 23876:20150114:153410.860 db_odbc_discovery() column[1]:'host'
 23876:20150114:153410.860 db_odbc_discovery() column[2]:'COUNT(h2.host)'
 23876:20150114:153410.860 End of db_odbc_discovery():NOTSUPPORTED
 23876:20150114:153410.860 Item [Zabbix server:db.odbc.discovery[proxies,{$DSN}]] error: Cannot convert
```

Теперь, когда мы понимаем, как SQL-запрос преобразуется в объект JSON, мы можем использовать макрос {#HOST} в прототипах элементов данных:



После выполнения обнаружения для каждого прокси будет создан элемент данных:



Использование db.odbc.get

При использовании `db.odbc.get[,{$DSN}]` и следующего примера SQL:

```
mysql> SELECT h1.host, COUNT(h2.host) AS count FROM hosts h1 LEFT JOIN hosts h2 ON h1.hostid = h2.proxyid
+---------+-------+
| host    | count |
+---------+-------+
| Japan 1 |     5 |
| Japan 2 |    12 |
| Latvia  |     3 |
+---------+-------+
3 rows in set (0.01 sec)
```

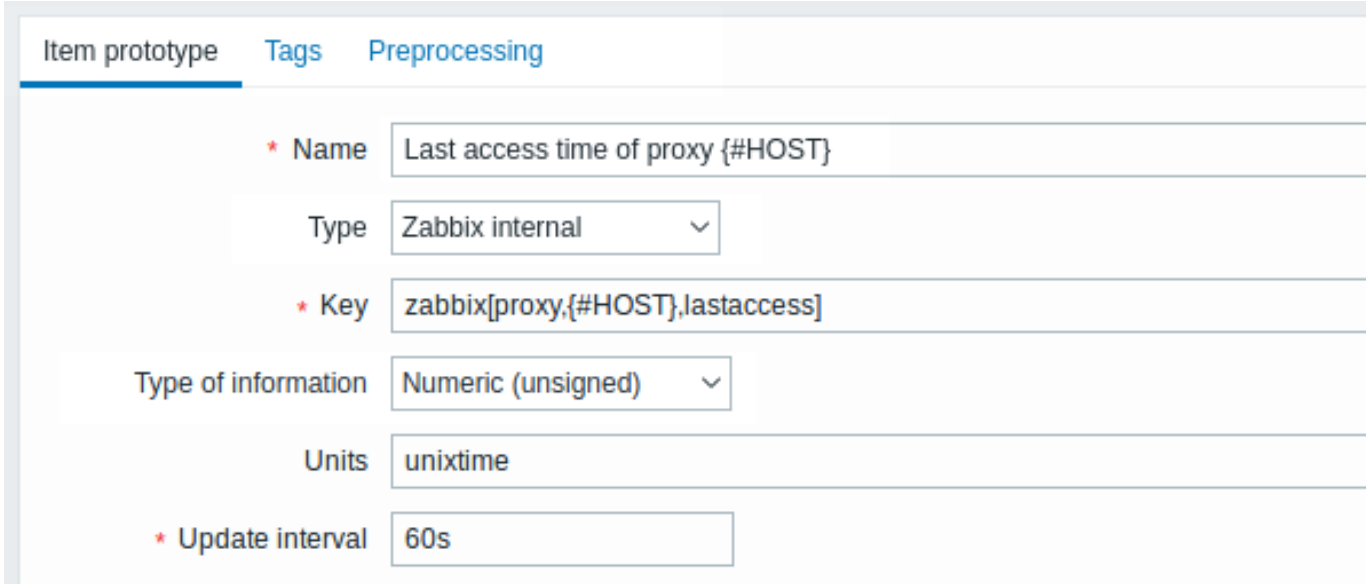
будет возвращён следующий JSON:

```
[
    {
        "host": "Japan 1",
        "count": "5"
    },
    {
        "host": "Japan 2",
        "count": "12"
    },
    {
        "host": "Latvia",
        "count": "3"
    }
]
```

Как видите, здесь нет макросов низкоуровневого обнаружения. Однако пользовательские макросы низкоуровневого обнаружения можно создать на вкладке LLD macros правила обнаружения с использованием JSONPath, например:

```
{#HOST} → $.host
```

Теперь этот макрос {#HOST} можно использовать в прототипах элементов данных:



13 Обнаружение с использованием данных Prometheus

Обзор

Данные, предоставленные в формате строк Prometheus, могут использоваться для обнаружения низкого уровня.

Подробную информацию о том, как в Zabbix реализован запрос данных Prometheus, см. в разделе Проверки Prometheus.

Конфигурация

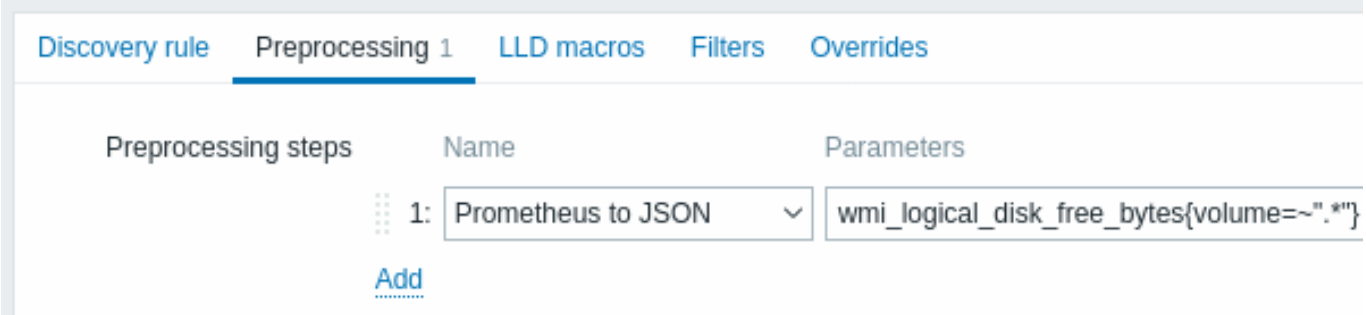
Правило низкоуровневого обнаружения следует создать как зависимый элемент данных для основного HTTP элемента данных, который собирает данные Prometheus.

Prometheus в JSON

В правиле обнаружения перейдите на вкладку Preprocessing и выберите вариант предобработки *Prometheus в JSON*. Данные в формате JSON необходимы для обнаружения, и вариант предобработки *Prometheus в JSON* вернёт именно их, со следующими атрибутами:

- имя метрики
- значение метрики
- help (если присутствует)
- type (если присутствует)
- labels (если присутствуют)
- исходная строка

Например, при запросе `wmi_logical_disk_free_bytes`:



из следующих строк Prometheus:

```
# HELP wmi_logical_disk_free_bytes Free space in bytes (LogicalDisk.PercentFreeSpace)
# TYPE wmi_logical_disk_free_bytes gauge
wmi_logical_disk_free_bytes{volume="C:"} 3.5180249088e+11
wmi_logical_disk_free_bytes{volume="D:"} 2.627731456e+09
wmi_logical_disk_free_bytes{volume="HarddiskVolume4"} 4.59276288e+08
```

будет возвращено:

```
[
    {
        "name": "wmi_logical_disk_free_bytes",
        "help": "Free space in bytes (LogicalDisk.PercentFreeSpace)",
        "type": "gauge",
        "labels": {
            "volume": "C:"
         },
        "value": "3.5180249088e+11",
        "line_raw": "wmi_logical_disk_free_bytes{volume=\"C:\"} 3.5180249088e+11"
    },
    {
        "name": "wmi_logical_disk_free_bytes",
        "help": "Free space in bytes (LogicalDisk.PercentFreeSpace)",
        "type": "gauge",
        "labels": {
            "volume": "D:"
         },
        "value": "2.627731456e+09",
        "line_raw": "wmi_logical_disk_free_bytes{volume=\"D:\"} 2.627731456e+09"
```

```
    },
    {
        "name": "wmi_logical_disk_free_bytes",
        "help": "Free space in bytes (LogicalDisk.PercentFreeSpace)",
        "type": "gauge",
        "labels": {
            "volume": "HarddiskVolume4"
         },
        "value": "4.59276288e+08",
        "line_raw": "wmi_logical_disk_free_bytes{volume=\"HarddiskVolume4\"} 4.59276288e+08"
    }
]
```

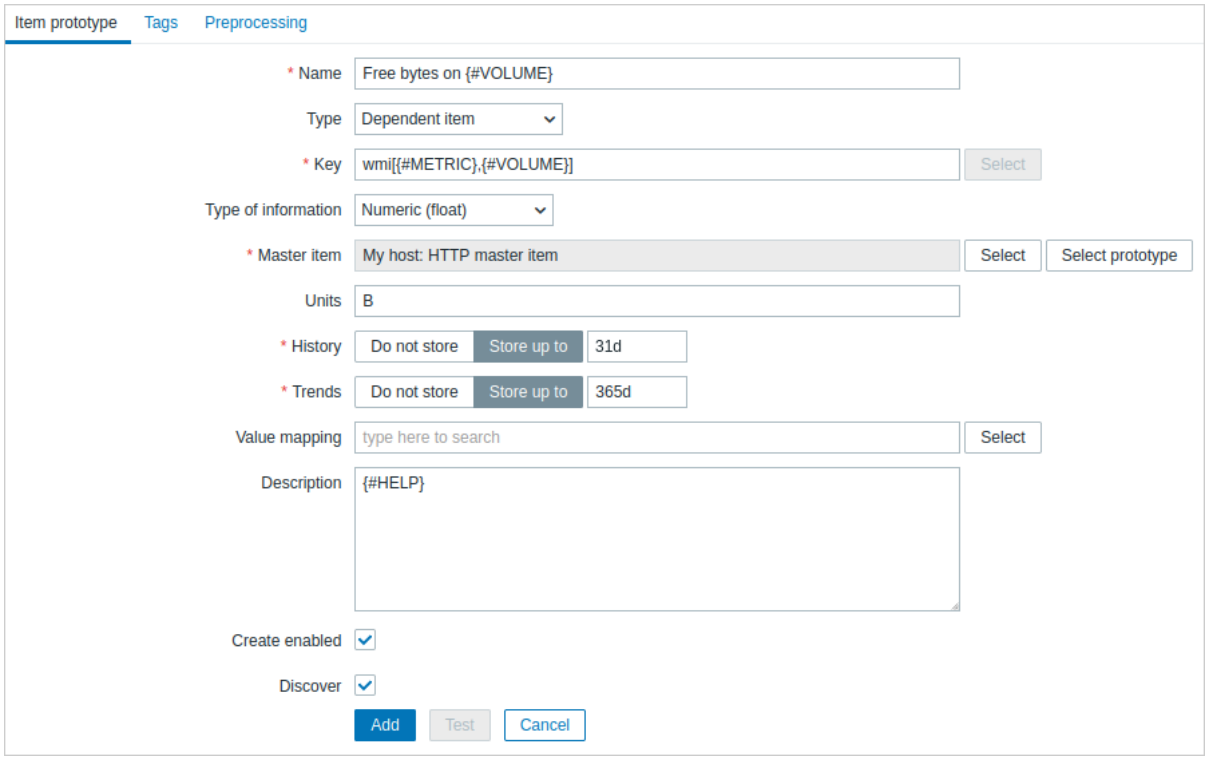
Сопоставление макросов LLD

Затем вам нужно перейти на вкладку макросов LLD и выполнить следующие сопоставления:

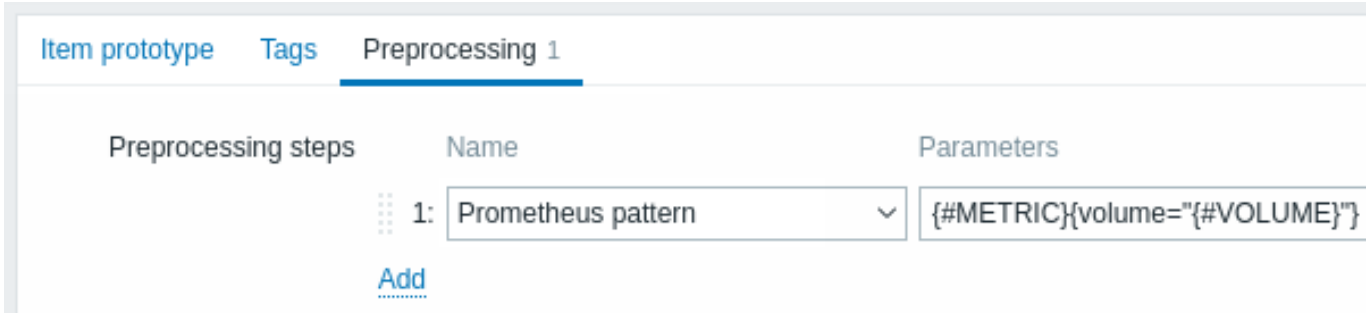
```
{#VOLUME}=$.labels['volume']
{#METRIC}=$['name']
{#HELP}=$['help']
```

Прототип элемента данных

Вы можете создать прототип элемента данных следующим образом:



с параметрами предобработки:



14 Обнаружение блочных устройств

Аналогично тому, как обнаруживаются файловые системы, можно также обнаруживать блочные устройства и их тип.

Ключ элемента данных

Ключ элемента данных, который следует использовать в правиле обнаружения:

`vfs.dev.discovery`

Этот элемент данных поддерживается только на платформах Linux.

Вы можете создавать правила обнаружения, используя этот элемент обнаружения, и:

- фильтр: **{#DEVNAME} соответствует** `sd[\D]$` — для обнаружения устройств с именами "sd0", "sd1", "sd2", ...
- фильтр: **{#DEVTYPE} соответствует** `disk` **И {#DEVNAME} не соответствует** `^loop.*` — для обнаружения устройств типа disk, имя которых не начинается с "loop"

Поддерживаемые макросы

Этот ключ обнаружения возвращает два макроса — {#DEVNAME} и {#DEVTYPE}, которые определяют имя и тип блочного устройства соответственно, например:

```
[
   {
      "{#DEVNAME}":"loop1",
      "{#DEVTYPE}":"disk"
   },
   {
      "{#DEVNAME}":"dm-0",
      "{#DEVTYPE}":"disk"
   },
   {
      "{#DEVNAME}":"sda",
      "{#DEVTYPE}":"disk"
   },
   {
      "{#DEVNAME}":"sda1",
      "{#DEVTYPE}":"partition"
   }
]
```

Обнаружение блочных устройств позволяет использовать элементы данных `vfs.dev.read[]` и `vfs.dev.write[]` для создания прототипов элементов данных с использованием макроса {#DEVNAME}, например:

- "vfs.dev.read[{#DEVNAME},sps]"
- "vfs.dev.write[{#DEVNAME},sps]"

{#DEVTYPE} предназначен для фильтрации устройств.

15 Обнаружение интерфейсов узла сети в Zabbix

Обзор

Существует возможность обнаружения всех интерфейсов, настроенных для узла сети в веб-интерфейсе Zabbix.

Ключ элемента данных

Элемент данных, который следует использовать в правиле обнаружения, — это внутренний элемент данных Zabbix:

`zabbix[host,discovery,interfaces]`

Этот элемент данных возвращает JSON с описанием интерфейсов, включая:

- IP-адрес/DNS-имя узла сети (в зависимости от настройки узла сети «Connect to»)
- Номер порта
- Тип интерфейса (агент Zabbix, SNMP, JMX, IPMI)
- Является ли он интерфейсом по умолчанию
- Включена ли функция bulk request — только для интерфейсов SNMP.

Например:

`[{"{#IF.CONN}":"192.168.3.1","{#IF.IP}":"192.168.3.1","{#IF.DNS}":"","{#IF.PORT}":"10050","{#IF.TYPE}":"AG`

При наличии нескольких интерфейсов их записи в JSON упорядочиваются по:

- Типу интерфейса,

- Признаку default — интерфейс по умолчанию располагается перед интерфейсами, не являющимися интерфейсами по умолчанию,
- ID интерфейса (по возрастанию).

Поддерживаемые макросы

Следующие макросы поддерживаются для использования в фильтре правила обнаружения и прототипах элементов данных, триггеров и графиков:

Макрос	Описание
{#IF.CONN}	IP адрес или DNS имя хоста интерфейса.
{#IF.IP}	IP адрес интерфейса.
{#IF.DNS}	DNS имя хоста интерфейса.
{#IF.PORT}	Номер порта интерфейса.
{#IF.TYPE}	Тип интерфейса («AGENT», «SNMP», «JMX» или «IPMI»).
{#IF.DEFAULT}	Состояние умолчания у интерфейса: 0 - не является интерфейсом по умолчанию 1 - интерфейс по умолчанию
{#IF.SNMP.BULK}	Состояние массовой обработки SNMP у интерфейса: 0 - деактивировано 1 - активировано Этот макрос возвращается, только если типом интерфейса является «SNMP».

8 Пользовательские правила LLD

Обзор

Также можно создать полностью настраиваемое правило LLD, обнаруживающее любые типы сущностей — например, базы данных на сервере баз данных.

Для этого следует создать пользовательский элемент данных, который возвращает строку JSON, указывающую обнаруженные объекты и, при необходимости, некоторые их свойства. Количество макросов на сущность не ограничено — в то время как встроенные правила обнаружения возвращают один или два макроса (например, два для обнаружения файловых систем), можно возвращать и больше.

Пример

Требуемый формат JSON-строки лучше всего показать на примере. Предположим, что вы используете старый агент Zabbix 1.8 (который не поддерживает ключ vfs.fs.discovery), но вам по-прежнему необходимо обнаруживать файловые системы. Ниже приведён простой Perl-скрипт для Linux, который обнаруживает смонтированные файловые системы и выводит результат в формате JSON, включая как имя файловой системы, так и её тип. Один из способов его использования — в качестве 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 имён макросов LLD: **0-9** , **A-Z** , **_** , **.** Строчные буквы в именах не поддерживаются.

Ниже показан пример его вывода (переформатированный 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 позволяет использовать 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 (переформатированный 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}" в качестве макроса и "rootfs|ext3" в качестве регулярного выражения.

Note:
Для пользовательских правил LLD необязательно 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 другим массивом или значением.

Обратите внимание, что при использовании пользовательского параметра возвращаемое значение ограничено 16 МБ. Подробнее см. в разделе ограничения данных для возвращаемых значений LLD.

14 Распределенный мониторинг

Обзор Zabbix обеспечивает эффективные и надежные пути мониторинга распределенной IT инфраструктуры, используя прокси.

Прокси можно использовать для сбора данных в локально от имени централизованного Zabbix сервера, а затем перенаправлять данные на сервер.

Возможности прокси

При выборе использования/неиспользования прокси необходимо учитывать несколько факторов.

	Прокси
Легковесность	**Да**
GUI	Нет
Работает независимо	**Да**
Простота обслуживания	**Да**
Автоматическое создание БД	**Да**[1]
Локальное администрирование	Нет
Готовность для встроенного оборудования	**Да**
Односторонние TCP-соединения	**Да**
Централизованная конфигурация	**Да**
Генерирует уведомления	Нет

[1] Функция автоматического создания БД работает только с SQLite. Для других поддерживаемых баз данных требуется настройка вручную.

Attention:
Прокси Zabbix не учитывает периоды обслуживания; подробности см. в разделе Расчет очередей во время обслуживания.

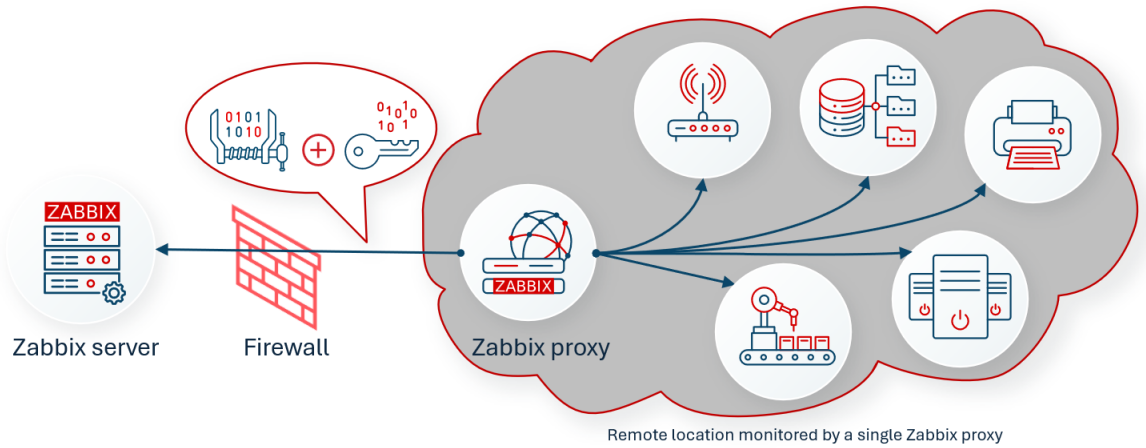
1 Прокси

Обзор Прокси Zabbix может собирать данные о производительности и доступности от имени сервера Zabbix. Таким образом, прокси может взять на себя часть нагрузки по сбору данных и разгрузить сервер Zabbix.

Кроме того, использование прокси — это самый простой способ реализации централизованного и распределенного мониторинга, когда все агенты и прокси отчитываются одному серверу Zabbix, а все данные собираются централизованно.

Прокси Zabbix можно использовать для:

- Мониторинга удаленных площадок
- Мониторинга площадок с ненадежной связью
- Разгрузки сервера Zabbix при мониторинге тысяч устройств
- Упрощения обслуживания распределенного мониторинга



Для прокси требуется только одно TCP-соединение с сервером Zabbix. Таким образом, проще обойти ограничения межсетевого экрана, так как нужно настроить только одно правило межсетевого экрана.

Attention:
Прокси Zabbix должен использовать отдельную базу данных. Указание базы данных сервера Zabbix нарушит конфигурацию.

Все данные, собранные прокси, хранятся локально перед передачей на сервер. Таким образом, никакие данные не будут потеряны из-за временных проблем со связью с сервером. Параметры *ProxyLocalBuffer* и *ProxyOfflineBuffer* в файле конфигурации прокси определяют, как долго данные хранятся локально.

Attention:
Может случиться так, что прокси, который получает последние изменения конфигурации напрямую из базы данных сервера Zabbix, имеет более актуальную конфигурацию, чем сервер Zabbix, чья конфигурация может обновляться не так быстро из-за значения CacheUpdateFrequency. В результате прокси может начать собирать данные и отправлять их на сервер Zabbix, который будет игнорировать эти данные.

Прокси Zabbix является сборщиком данных. Он не вычисляет триггеры, не обрабатывает события и не отправляет оповещения. Для обзора функциональности прокси см. следующую таблицу:

Функция		Поддерживается прокси
Элементы данных		
	Проверки агента Zabbix	**Да**
	Проверки агента Zabbix (активные)	**Да** [1]
	Простые проверки	**Да**
	Элементы данных trapper	**Да**
	Проверки SNMP	**Да**
	Ловушки SNMP	**Да**
	Проверки IPMI	**Да**
	Проверки JMX	**Да**
	Мониторинг лог-файлов	**Да**
	Внутренние проверки	**Да**
	Проверки SSH	**Да**
	Проверки Telnet	**Да**
	Внешние проверки	**Да**
	Зависимые элементы данных	**Да**
	Элементы данных script	**Да**
	Элементы данных browser	**Да**
Встроенный веб-мониторинг		**Да**
Предобработка значений элементов данных		**Да**
Обнаружение сети		**Да**
Авторегистрация активного агента		**Да**
Низкоуровневое обнаружение		**Да** [2]
Удаленные команды		**Да**
Вычисление триггеров		*Нет*
Обработка событий		*Нет*

Функция	Поддерживается прокси
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] Для LLD прокси Zabbix только собирает и предварительно обрабатывает данные, а затем отправляет их на сервер Zabbix для дальнейшей обработки.

Защита от перегрузки

Если сервер Zabbix был недоступен некоторое время, а прокси собрали много данных, то после запуска сервера он может оказаться перегружен (использование кэша истории некоторое время держится на уровне 95-100%). Такая перегрузка может привести к снижению производительности, когда проверки обрабатываются медленнее, чем должны. Для предотвращения проблем, возникающих из-за перегрузки кэша истории, была реализована защита от такого сценария.

Когда кэш истории сервера Zabbix заполнен, скорость записи в кэш истории ограничивается, что приостанавливает процессы сбора данных на сервере. Наиболее распространенный случай перегрузки кэша истории возникает после простоя сервера, когда прокси выгружают собранные данные. Чтобы этого избежать, было добавлено ограничение прокси (в настоящее время его нельзя отключить).

Когда использование кэша истории достигает 80%, сервер Zabbix переходит в режим ограничения. В режиме ограничения сервер принимает данные от прокси только тогда, когда использование кэша истории ниже 60%, поочередно выбирая прокси, от которых принимаются данные. Как только использование кэша истории опускается ниже 20%, сервер возвращается в обычный режим.

Кроме того, в обычном режиме сервер Zabbix ограничивает отдельные прокси, отправляющие очень большие пакеты (10 000+ записей), если использование кэша истории превышает 60%. Это решение принимается в момент, когда сервер оценивает выгрузку данных от прокси, поэтому оно не всегда может сразу отражаться на графиках использования кэша истории (внутренний элемент данных `zabbix[wcache,history,pused]` и интервал его обновления могут не фиксировать кратковременные всплески).

Этот режим ограничения будет продолжаться до тех пор, пока использование кэша снова не достигнет 80%, не опустится до 20% или пока список ограничений не станет пустым. В первом случае сервер снова перестанет принимать данные от прокси. В двух других случаях сервер начнет работать в обычном режиме, принимая данные от всех прокси.

Вышеуказанную информацию можно проиллюстрировать следующей таблицей:

Использование кэша записи истории	Режим сервера Zabbix	Действие сервера Zabbix
Достигает 80%	Ожидание	Прекращает принимать данные от прокси, но поддерживает *список ограничений* (приоритетный список прокси, с которыми нужно связаться позже).
Достигает 60%	Обычный, но готовый к ограничению	Может отклонять очень большие выгрузки от прокси (более 10 тыс. записей) при принятии решения о приеме данных; продолжает принимать остальные данные от прокси.
Опускается до 20%	Обычный	Удаляет список ограничений и начинает принимать данные от прокси в обычном режиме.

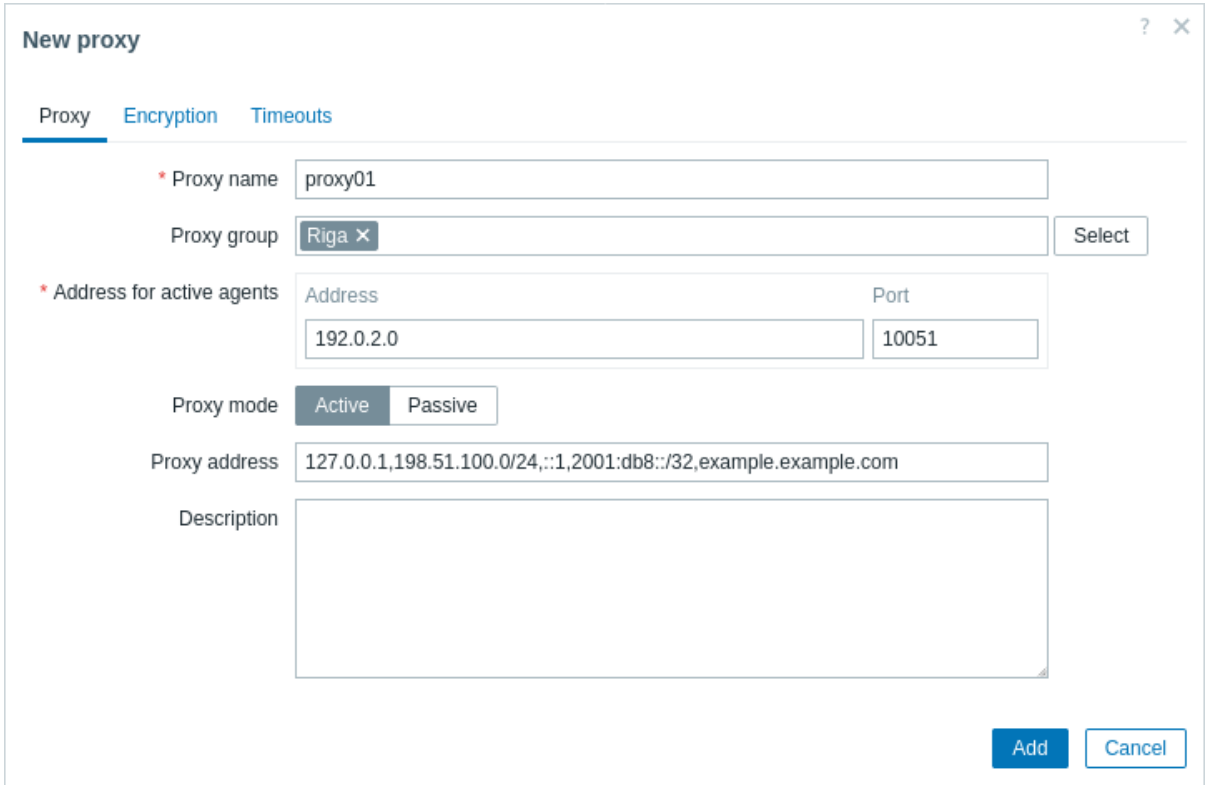
Вы можете использовать внутренний элемент данных zabbix[wcache,history,pused], чтобы сопоставить это поведение сервера Zabbix с метрикой.

Настройка После того как вы установили и настроили прокси, пришло время настроить его в веб-интерфейсе Zabbix.

Добавление прокси

Чтобы настроить прокси в веб-интерфейсе Zabbix:

- Перейдите в: *Administration* > *Proxies*
- Нажмите *Create proxy*

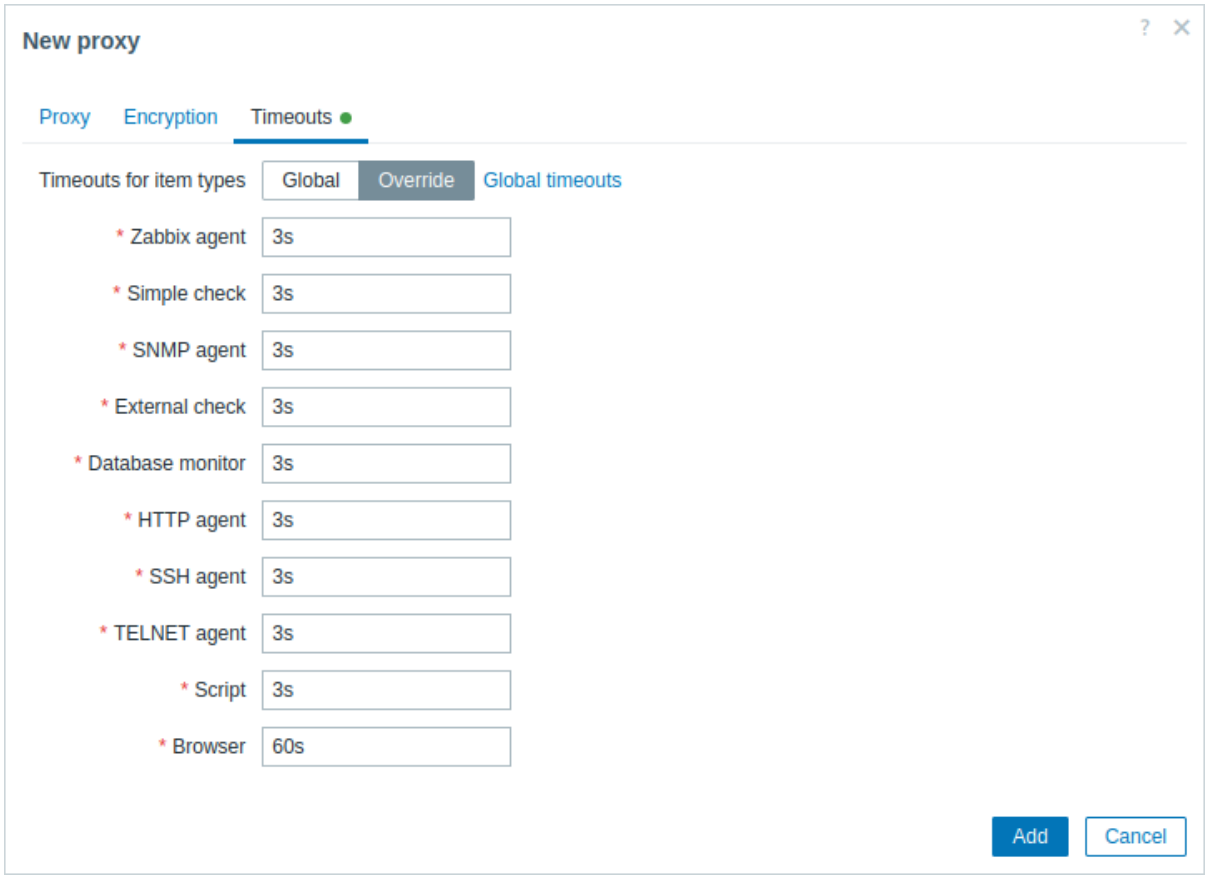


Параметр	Описание
Proxy name	Введите имя прокси. Оно должно совпадать со значением параметра *Hostname* в файле конфигурации прокси.
Proxy group	Выберите одну группу прокси для балансировки нагрузки/высокой доступности прокси.
Address for active agents	Введите адрес, к которому должны подключаться отслеживаемые активные агенты или отправители. Поддерживается **только** для агентов Zabbix 7.0 и новее. Этот адрес используется для подключения как к активным, так и к пассивным прокси. Это поле доступно, только если в поле *Proxy group* выбрана группа прокси.
Address	IP-адрес/DNS-имя для подключения.
Port	Номер TCP-порта для подключения (по умолчанию 10051). Пользовательские макросы поддерживаются.
Proxy mode	Выберите режим прокси. **Active** — прокси будет подключаться к серверу Zabbix и запрашивать данные конфигурации **Passive** — сервер Zabbix подключается к прокси Обратите внимание, что без шифрования соединений данные конфигурации прокси (чувствительные данные) могут стать доступными сторонам, имеющим доступ к порту trapper сервера Zabbix при использовании активного прокси. Это возможно, потому что любой может выдать себя за активный прокси и запросить данные конфигурации, если аутентификация не выполняется или адреса прокси не ограничены в поле *Proxy address*.
Proxy address	Если указано, то запросы активного прокси принимаются только из этого списка IP-адресов, разделённых запятыми, при необходимости в нотации CIDR, или DNS-имён активного прокси Zabbix. Это поле доступно, только если в поле *Proxy mode* выбран активный прокси. Макросы не поддерживаются.
Interface	Введите сведения об интерфейсе для пассивного прокси. Это поле доступно, только если в поле Proxy mode выбран пассивный прокси.
Address	IP-адрес/DNS-имя пассивного прокси.
Port	Номер TCP-порта пассивного прокси (по умолчанию 10051). Пользовательские макросы поддерживаются.
Description	Введите описание прокси.

Вкладка **Encryption** позволяет требовать зашифрованные соединения с прокси.

Параметр	Описание
Connections to proxy	Как сервер подключается к пассивному прокси: без шифрования (по умолчанию), с использованием PSK (предварительно разделённого ключа) или сертификата.
Connections from proxy	Выберите, какие типы соединений разрешены от активного прокси. Можно выбрать несколько типов соединений одновременно (это полезно для тестирования и переключения на другой тип соединения). По умолчанию используется ”No encryption”.
Issuer	Разрешённый издатель сертификата. Сначала сертификат проверяется с помощью CA (центра сертификации). Если он действителен и подписан этим CA, то поле *Issuer* можно использовать для дополнительного ограничения допустимого CA. Это необязательное поле, предназначенное для использования, если в вашей установке Zabbix применяются сертификаты от нескольких CA.
Subject	Разрешённый субъект сертификата. Сначала сертификат проверяется с помощью CA. Если он действителен и подписан этим CA, то 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 разделённого ключа. Не указывайте конфиденциальную информацию в идентификаторе PSK, так как он передаётся по сети без шифрования, чтобы сообщить получателю, какой PSK использовать.
PSK	Предварительно разделённый ключ (hex-строка). Максимальная длина: 512 шестнадцатеричных цифр (256-байтный PSK), если Zabbix использует библиотеку GnuTLS или OpenSSL, 64 шестнадцатеричных цифры (32-байтный PSK), если Zabbix использует библиотеку mbed TLS (PolarSSL). Пример: 1f87b595725ac58dd977beef14b97461a7c1045b9a1c963065002c5473194952

Вкладка **Timeouts** позволяет переопределить глобальные тайм-ауты для типов элементов данных, которые это поддерживают.



Параметр	Описание
Timeouts for item types	Задайте тайм-аут элемента данных (в зависимости от его типа): **Global** — использовать глобальный тайм-аут (отображается в затенённом поле *Timeout* для каждого типа элемента данных); **Override** — задать пользовательский тайм-аут (в поле *Timeout* для каждого типа элемента данных). Допустимый диапазон: 1 - 600s (по умолчанию: наследуется из глобальных тайм-аутов). Поддерживаются суффиксы времени, например 30s, 1m, и пользовательские макросы. Нажатие на ссылку *Global timeouts* позволяет настроить глобальные тайм-ауты. Обратите внимание, что ссылка *Global timeouts* видна только пользователям типа *Super admin* с правами на раздел веб-интерфейса *Administration* > *General*. Обратите внимание, что хотя тайм-ауты на уровне прокси переопределяют глобальные, они, в свою очередь, будут переопределены индивидуальными тайм-аутами элементов данных, если они настроены.

Note:
Если основная версия прокси не совпадает с основной версией сервера, рядом с *Timeouts for item types* будет отображаться значок с сообщением при наведении "Timeouts disabled because the proxy and server versions do not match". В таких случаях прокси будет использовать 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

Настройка узла сети

Вы можете указать, что отдельный узел сети должен отслеживаться прокси или группой прокси, в форме настройки узла сети, используя поле *Monitored by*.



Массовое обновление узлов сети — это еще один способ указать, что узлы сети должны отслеживаться прокси или группой прокси.

1 Синхронизация конфигурации мониторинга

Обзор

На этой странице приведены сведения об обновлении конфигурации мониторинга для прокси, то есть о том, как изменения, внесённые в конфигурацию мониторинга на сервере, синхронизируются с прокси.

Инкрементальное обновление

Обновление конфигурации прокси выполняется инкрементально. Во время синхронизации конфигурации обновляются только измененные сущности (то есть, если ни одна сущность не была изменена, ничего отправлено не будет). Такой подход позволяет экономить ресурсы и задавать меньший интервал (почти мгновенный) для обновления конфигурации прокси.

Изменения конфигурации прокси отслеживаются с помощью номеров ревизий. В данные конфигурации, отправляемые на прокси, включаются только сущности, ревизии которых больше ревизии конфигурации прокси.

Сущности для синхронизации конфигурации следующие:

Сущность	Подробности
данные TLS авторегистрации	Все данные TLS авторегистрации.

Сущность	Подробности
выражения	Все выражения (регулярные выражения, таблицы выражений).
глобальная конфигурация	Глобальная конфигурация, определенная в таблице 'config'
узел сети	Все свойства, интерфейсы, инвентарные данные, элементы данных, предобработка элементов данных, параметры элементов данных, веб-сценарии узла сети.
макросы узла сети	Все макросы, определенные на узле сети, и все связанные с ним ID шаблонов.
правило обнаружения	Правила обнаружения и проверки, назначенные прокси.
прокси	

Это означает:

- Если элемент данных изменен на **узле сети**, будет синхронизирована вся конфигурация этого узла сети.
- Если изменено **регулярное выражение**, будут синхронизированы все регулярные выражения.

Исключением являются макросы узла сети, которые также отправляются, если на узле сети было изменено что-либо.

Команда `-R config_cache_reload` на прокси также инициирует инкрементальное обновление.

Обратите внимание, что полная синхронизация конфигурации выполняется при запуске/перезапуске прокси, переключении HA, изменении токена сессии или если обновление конфигурации на прокси завершилось неудачно, например, если соединение было разорвано во время получения данных конфигурации.

Параметры конфигурации

Параметр **ProxyConfigFrequency** определяет, как часто конфигурация прокси-сервера синхронизируется с сервером (по умолчанию 10 секунд).

Обратите внимание, что ProxyConfigFrequency:

- параметр сервера для пассивных прокси
- параметр proxy для активных прокси

На 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-сервер зарегистрирует ошибку и завершит работу.

Расчет очередей во время обслуживания

> **Attention:**
> Прокси Zabbix не знает о периодах обслуживания; подробности смотрите в разделе Расчет очередей во время обслуживания.

2 Балансировка нагрузки прокси и высокая доступность

Обзор

Прокси Zabbix можно объединять в группы прокси, чтобы обеспечить балансировку нагрузки между прокси и высокую доступность.

Балансировка нагрузки между прокси и высокая доступность — это автоматическое перераспределение узлов сети между прокси в группе прокси:

- Если прокси отключается, его узлы сети будут перемещены на другие прокси, что позволяет сохранить высокую доступность прокси.
- Если у прокси значительно больше или меньше узлов сети, чем у других прокси, его узлы сети будут перемещены на другие прокси для балансировки нагрузки.

Перераспределение узлов сети работает только между прокси в группе, которые соответствуют следующим условиям:

- Прокси работают на Zabbix 7.0 или новее.
- Версия прокси соответствует версии сервера Zabbix. При использовании Zabbix агент (пассивный) версия прокси должна соответствовать версии агента. Для активных агентов требуется только Zabbix 7.0 или новее.
- Группа прокси имеет состояние «в сети».

- Узлы сети настроены на 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, чтобы отслеживать состояние 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 групп прокси, который непрерывно отслеживает состояние всех прокси в каждой группе прокси и распределение их узлов сети.

Высокая доступность прокси внутри группы обеспечивается с помощью переключения при отказе прокси: когда прокси отключается, его узлы сети немедленно перераспределяются на другие прокси. Также выполняется балансировка нагрузки прокси, поскольку узлы сети переназначаются 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 или дефицит узлов сети). Если дисбаланс сохраняется после льготного периода (10 x задержка переключения при отказе), группа прокси ставится в очередь на перераспределение узлов сети.

Менеджер групп прокси перераспределяет узлы сети, используя следующую логику:

1. Вычислить среднее количество узлов сети на прокси.
2. Для прокси с избытком узлов сети — переместить избыточные узлы сети в пул неназначенных прокси.
3. Для прокси с дефицитом узлов сети — вычислить, сколько узлов сети необходимо для достижения баланса.
4. Удалить требуемое количество узлов сети с прокси с наибольшим количеством узлов сети.
5. Переместить неназначенные узлы сети на прокси с наименьшим количеством узлов сети.

Примеры перераспределения узлов сети:

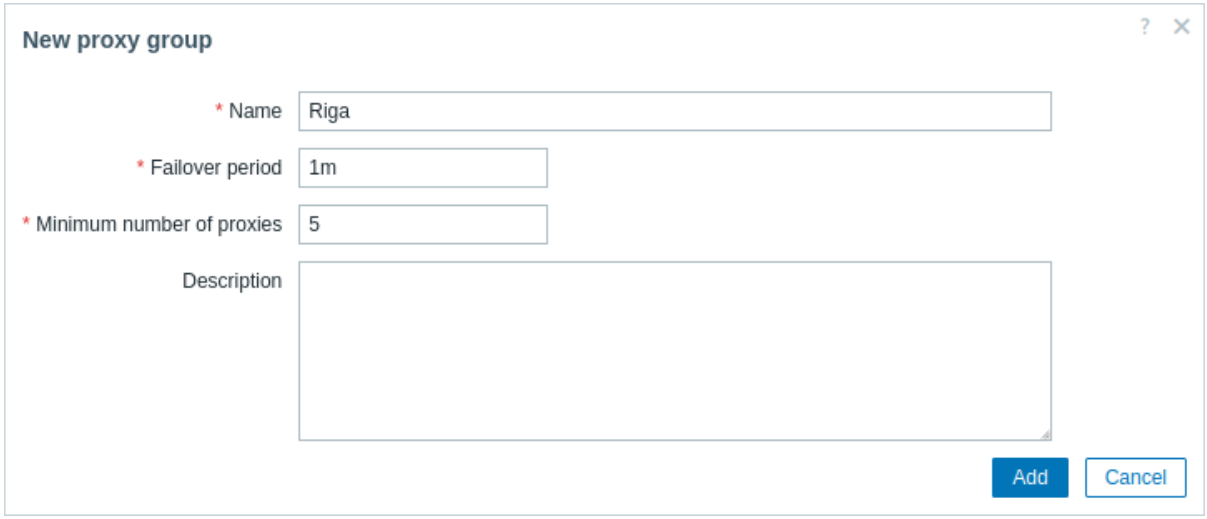
Узлы сети на прокси	Среднее по группе	Переназначение узлов сети
100	50	Да
60	50	Нет
40	50	Нет
25	50	Да
15	5	Да
10	5	Нет

Attention:
Если группа прокси отслеживает менее 10 узлов сети, это 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 *Administration* > *Proxy groups*
2. Нажмите *Create proxy group*



Параметр	Описание
Name	Имя группы прокси.
Failover period	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, состояние прокси меняется на *Offline*, а его узлы сети немедленно перераспределяются на другие прокси. Балансировка нагрузки прокси начинается через 10 x этот период. Поддерживаются суффиксы времени (например, 30s, 1m) и пользовательские макросы.
Minimum number of proxies	Минимальное количество доступных прокси, необходимое для того, чтобы группа прокси оставалась доступной (по умолчанию: 1; диапазон: 1–1000). Поддерживаются пользовательские макросы. Это значение должно быть меньше общего количества прокси в группе. Например, в группе из 10 прокси установка минимального значения 10 приведет к тому, что группа станет недоступной при отказе любого прокси. Обратите внимание, что доступные прокси в недоступной группе продолжают работать нормально, но балансировка нагрузки/высокая доступность выполняться не будут.
Description	Описание группы прокси.
Proxies	Отображает список не более чем из пяти прокси (в виде ссылок или обычного текста, в зависимости от прав пользователя на прокси) при редактировании группы, содержащей 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 отслеживаются группой прокси, а не отдельными прокси (для переноса узлов сети с прокси в группу прокси можно использовать массовое обновление).

Если используется Zabbix агент, 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 Zabbix агент 7.0 (или более поздней версии).

Если параметр `ServerActive` включает только один прокси из группы прокси (или сервер Zabbix), агент всё равно сможет подключиться к правильному прокси. Когда служба агента запускается и подключается к указанному прокси, агент получает и кэширует полный список IP-адресов прокси и их текущую нагрузку в группе. Затем активные проверки будут перенаправляться на правильный доступный прокси для узла сети на основе текущего назначения прокси-узел сети в группе прокси.

Warning:
Указание только одного прокси в параметре `ServerActive` Zabbix агента может привести к потере данных мониторинга, если агент запускается/перезагружается в то время, когда указанный прокси недоступен.

Note:
При использовании Zabbix sender запросы данных также перенаправляются на правильный доступный прокси для узла сети на основе текущего назначения прокси-узел сети в группе прокси. Однако, если вы отправляете значения для нескольких узлов сети из входного файла, используйте опцию `-g`, чтобы предотвратить отправку данных не тому прокси.

Zabbix агент 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 узел. Например:

- Во время активных проверок прокси может перенаправить агента на другой прокси. Если этот прокси заблокирован межсетевым экраном, обмен данными зависнет в ожидании ответа.
- В стабильных конфигурациях высокой доступности без недавней перебалансировки агенты могут никогда не обращаться к резервным прокси. Если правила межсетевого экрана изменились и не были проверены, переключение на резервный узел может завершиться неудачей.

Тестирование балансировки нагрузки прокси

Чтобы протестировать балансировку нагрузки прокси:

1. Настройте группу прокси.
2. Убедитесь, что группа прокси находится в состоянии online.
3. Убедитесь, что узлы сети отслеживаются группой прокси, а не отдельными прокси (вы можете использовать массовое обновление узлов сети, чтобы переместить узлы сети с прокси в группу прокси).
4. Подождите несколько секунд, пока обновится конфигурация и узлы сети распределятся между прокси в группе прокси. Наблюдайте изменения, обновляя список узлов сети в *Администрирование* > *Прокси*.

Важные примечания

- SNMP traps 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.
- Проверки базы данных требуют расширенных разрешений на объекте базы данных/сервере.
- Узлы VMware, отслеживаемые группой прокси, будут случайным образом распределяться между прокси в группе. Это приводит к тому, что каждый прокси кэширует все данные VMware, создавая дополнительную нагрузку на vCenter.
- Узлы сети, созданные на основе данных авторегистрации от прокси в группе прокси, настраиваются как отслеживаемые этой группой прокси. Однако узлы сети, созданные на основе данных сетевого обнаружения от прокси в группе прокси, настраиваются как отслеживаемые этим прокси.

15 Шифрование

Обзор Zabbix поддерживает шифрованный обмен данными между компонентами Zabbix с использованием протокола Transport Layer Security (TLS) версий 1.2 и 1.3 (в зависимости от криптографической библиотеки). Поддерживается шифрование на основе сертификатов и на основе предварительно разделяемых ключей.

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 сервера/прокси
- 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 при взаимодействии с сервером, в то время как другие могут использовать шифрование на основе предварительно разделяемых ключей, а третьи могут продолжать использовать нешифрованный обмен данными (как и раньше).
- Сервер (прокси) может использовать разные настройки шифрования для разных узлов сети.

Программы-демоны Zabbix используют один порт прослушивания для шифрованных и нешифрованных входящих соединений. Добавление шифрования не требует открытия новых портов в межсетевых экранах.

Ограничения

- Закрытые ключи хранятся в открытом виде в файлах, доступных для чтения компонентами Zabbix во время запуска.

- Предварительно разделяемые ключи вводятся в веб-интерфейсе Zabbix и хранятся в базе данных Zabbix в открытом виде.
- 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 устанавливается с полным TLS-рукопожатием; кэширование сессий и tickets не реализованы.
- Добавление шифрования увеличивает время проверок 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 использовании шифрования для установления TLS-соединения добавляется около 1000 мс.
  - Может потребоваться увеличить тайм-ауты, иначе некоторые элементы данных и действия, запускающие удалённые скрипты на агентах, могут работать с незашифрованными соединениями, но завершаться по тайм-ауту при использовании шифрования.
- Шифрование не поддерживается обнаружением сети. Проверки агента Zabbix, выполняемые в процессе обнаружения сети, 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 должен быть скомпилирован и слинкован 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
- 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 будет использовать только сертификаты.

Note:
Подробнее о настройке SSL для веб-интерфейса Zabbix можно узнать в этих рекомендациях.

Библиотека выбирается указанием соответствующей опции для скрипта "configure":

- `--with-gnutls[=DIR]`
- `--with-openssl[=DIR]` (также используется для LibreSSL)

Например, чтобы настроить исходные коды для сервера и агента с *OpenSSL*, можно использовать что-то вроде:

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

Разные компоненты Zabbix могут быть скомпилированы с разными криптографическими библиотеками (например, сервер с *OpenSSL*, агент с *GnuTLS*).

Attention:
Если вы планируете использовать предварительно разделяемые ключи (PSK), рекомендуется использовать библиотеки *GnuTLS* или *OpenSSL 1.1.0* (или новее) в компонентах Zabbix, использующих PSK. Библиотеки *GnuTLS* и *OpenSSL 1.1.0* поддерживают наборы шифров PSK с Perfect Forward Secrecy. Более старые версии библиотеки *OpenSSL* (1.0.1, 1.0.2c) также поддерживают PSK, однако доступные наборы шифров PSK не обеспечивают Perfect Forward Secrecy.

Управление шифрованием соединений Соединения в Zabbix могут использовать:

- без шифрования (по умолчанию)
- шифрование на основе RSA-сертификатов
- 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 Zabbix прокси (в активном режиме указывает только соединения с сервером) и Zabbix агент (для активных проверок). В веб-интерфейсе Zabbix эквивалентом TLSConnect является поле

Connections to host на вкладке *Data collection → Hosts → <some host> → Encryption* и поле *Connections to proxy* на вкладке *Administration → Proxies → <some proxy> → Encryption*. Если настроенный тип шифрования для соединения не сработает, другие типы шифрования пробоваться не будут.

`TLSAccept` используется в конфигурационных файлах Zabbix прокси (в пассивном режиме указывает только соединения от сервера) и Zabbix агент (для пассивных проверок). В веб-интерфейсе Zabbix эквивалентом TLSAccept является поле *Connections from host* на вкладке *Data collection → Hosts → <some host> → Encryption* и поле *Connections from proxy* на вкладке *Administration → Proxies → <some proxy> → Encryption*.

Обычно для входящих соединений настраивается только один тип шифрования. Однако может потребоваться сменить тип шифрования, например, с нешифрованного на основанный на сертификатах, с минимальным временем простоя и возможностью отката. Для этого:

- Установите `TLSAccept=unencrypted,cert` в конфигурационном файле агента и перезапустите Zabbix агент
- Проверьте соединение с агентом с помощью zabbix_get, используя сертификат. Если всё работает, можно перенастроить шифрование для этого агента в веб-интерфейсе Zabbix на вкладке *Data collection → Hosts → <some host> → Encryption*, установив *Connections to host* в значение "Certificate".
- Когда кэш конфигурации сервера обновится (и конфигурация прокси также обновится, если узел сети мониторится через прокси), соединения с этим агентом будут зашифрованы
- Если всё работает как ожидается, можно установить `TLSAccept=cert` в конфигурационном файле агента и перезапустить Zabbix агент. Теперь агент будет принимать только зашифрованные соединения на основе сертификатов. Нешифрованные соединения и соединения на основе PSK будут отклоняться.

Аналогичным образом это работает на сервере и прокси. Если в веб-интерфейсе Zabbix в конфигурации узла сети поле *Connections from host* установлено в значение "Certificate", то от агента (активные проверки) и zabbix_sender (элементы данных trapper) будут приниматься только зашифрованные соединения на основе сертификатов.

Скорее всего, вы настроите входящие и исходящие соединения на использование одного и того же типа шифрования или вообще без шифрования. Однако технически возможно настроить их асимметрично, например, использовать шифрование на основе сертификатов для входящих соединений и шифрование на основе PSK для исходящих.

Конфигурация шифрования для каждого узла сети отображается в веб-интерфейсе Zabbix в разделе *Data collection → Hosts* в столбце *Agent encryption*. Например:

Example	Connections to host	Allowed connections from host	Rejected connections from host
NONE	Без шифрования	Без шифрования	Зашифрованные, на основе сертификатов и на основе PSK
CERT NONE PSK CERT	Зашифрованные, на основе сертификатов	Зашифрованные, на основе сертификатов	Без шифрования и зашифрованные на основе PSK
PSK NONE PSK CERT	Зашифрованные, на основе PSK	Зашифрованные, на основе PSK	Без шифрования и зашифрованные на основе сертификатов
PSK NONE PSK CERT	Зашифрованные, на основе PSK	Без шифрования и зашифрованные на основе PSK	Зашифрованные на основе сертификатов
CERT NONE PSK CERT	Зашифрованные, на основе сертификатов	Без шифрования, на основе PSK или на основе сертификатов	-

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 по их использованию с шифрованием.

Наборы шифров По умолчанию наборы шифров настраиваются внутренним образом во время запуска Zabbix.

Также поддерживаются пользовательские наборы шифров для GnuTLS и OpenSSL. Пользователи могут настроить наборы шифров в соответствии со своими политиками безопасности. Использование этой возможности необязательно (встроенные наборы шифров по умолчанию по-прежнему работают).

Для криптографических библиотек, скомпилированных с настройками по умолчанию, встроенные в Zabbix правила обычно приводят к следующим наборам шифров (в порядке от более высокого к более низкому приоритету):

Библиотека	Наборы шифров сертификатов	Наборы шифров PSK
GnuTLS 3.1.18	TLS_ECDHE_RSA_AES_128_GCM_SHA256 TLS_ECDHE_RSA_AES_128_CBC_SHA256 TLS_ECDHE_RSA_AES_128_CBC_SHA1 TLS_RSA_AES_128_GCM_SHA256 TLS_RSA_AES_128_CBC_SHA256 TLS_RSA_AES_128_CBC_SHA1	TLS_ECDHE_PSK_AES_128_CBC_SHA256 TLS_ECDHE_PSK_AES_128_CBC_SHA1 TLS_PSK_AES_128_GCM_SHA256 TLS_PSK_AES_128_CBC_SHA256 TLS_PSK_AES_128_CBC_SHA1
OpenSSL 1.0.2c	ECDHE-RSA-AES128-GCM-SHA256 ECDHE-RSA-AES128-SHA256 ECDHE-RSA-AES128-SHA AES128-GCM-SHA256 AES128-SHA256 AES128-SHA	PSK-AES128-CBC-SHA
OpenSSL 1.1.0	ECDHE-RSA-AES128-GCM-SHA256 ECDHE-RSA-AES128-SHA256 ECDHE-RSA-AES128-SHA AES128-GCM-SHA256 AES128-CCM8 AES128-CCM AES128-SHA256 AES128-SHA	ECDHE-PSK-AES128-CBC-SHA256 ECDHE-PSK-AES128-CBC-SHA PSK-AES128-GCM-SHA256 PSK-AES128-CCM8 PSK-AES128-CCM PSK-AES128-CBC-SHA256 PSK-AES128-CBC-SHA
OpenSSL 1.1.1d	TLS_AES_256_GCM_SHA384 TLS_CHACHA20_POLY1305_SHA256 TLS_AES_128_GCM_SHA256 ECDHE-RSA-AES128-GCM-SHA256 ECDHE-RSA-AES128-SHA256 ECDHE-RSA-AES128-SHA AES128-GCM-SHA256 AES128-CCM8 AES128-CCM AES128-SHA256 AES128-SHA	TLS_CHACHA20_POLY1305_SHA256 TLS_AES_128_GCM_SHA256 ECDHE-PSK-AES128-CBC-SHA256 ECDHE-PSK-AES128-CBC-SHA PSK-AES128-GCM-SHA256 PSK-AES128-CCM8 PSK-AES128-CCM PSK-AES128-CBC-SHA256 PSK-AES128-CBC-SHA

Настраиваемые пользователем наборы шифров Встроенные критерии выбора наборов шифров могут быть переопределены с помощью настроенных пользователем наборов шифров.

Attention:
Настраиваемые пользователем наборы шифров — это функция, предназначенная для опытных пользователей, которые понимают наборы шифров TLS, их безопасность и последствия ошибок, а также уверенно выполняют диагностику проблем TLS.

Встроенные критерии выбора наборов шифров могут быть переопределены с помощью следующих параметров:

Область переопределения	Параметр	Значение	Описание
Выбор набора шифров для сертификатов	TLSCipherCert13	Допустимые OpenSSL 1.1.1 строки шифров для протокола TLS 1.3 (их значения передаются в функцию 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 OpenSSL строки шифров для TLS 1.2 или допустимые GnuTLS строки приоритетов. Их значения передаются соответственно в функции `SSL_CTX_set_cipher_list()` или `gnutls_priority_init()`.	Критерии выбора набора шифров на основе сертификатов для TLS 1.2/1.3 (GnuTLS), TLS 1.2 (OpenSSL)

Область переопределения	Параметр	Значение	Описание
Выбор набора шифров для PSK	TLSCipherPSK13	Допустимые OpenSSL 1.1.1 строки шифров для протокола TLS 1.3 (их значения передаются в функцию 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 OpenSSL строки шифров для TLS 1.2 или допустимые GnuTLS строки приоритетов. Их значения передаются соответственно в функции `SSL_CTX_set_cipher_list()` или `gnutls_priority_init()`.	Критерии выбора набора шифров на основе PSK для TLS 1.2/1.3 (GnuTLS), TLS 1.2 (OpenSSL)
Объединенный список наборов шифров для сертификата и PSK	TLSCipherAll13	Допустимые OpenSSL 1.1.1 строки шифров для протокола TLS 1.3 (их значения передаются в функцию OpenSSL `SSL_CTX_set_ciphersuites()`).	Критерии выбора набора шифров для TLS 1.3 Только OpenSSL 1.1.1 или новее.
	TLSCipherAll	Допустимые OpenSSL строки шифров для TLS 1.2 или допустимые GnuTLS строки приоритетов. Их значения передаются соответственно в функции `SSL_CTX_set_cipher_list()` или `gnutls_priority_init()`.	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 установка значения TLSCipher* в криптографической библиотеке завершается ошибкой, то сервер, прокси или агент не запустится, а ошибка будет записана в журнал.

Важно понимать, в каких случаях 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 задаётся параметром `--tls-connect`)

Входящие соединения

С входящими соединениями всё немного сложнее, поскольку правила зависят от конкретных компонентов и конфигурации.

Для Zabbix **агент**:

Настройка соединения 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 соединения	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 определённую закономерность:

- TLSCipherAll и TLSCipherAll13 можно указывать только в том случае, если используется объединённый список наборов шифров на основе сертификатов **и** PSK. Это происходит в двух случаях: сервер (прокси) с настроенным сертификатом (наборы шифров PSK всегда настраиваются на сервере, прокси, если криптографическая библиотека поддерживает PSK), агент, настроенный на приём 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 для `TLSCipher*`. Они могут быть хорошей отправной точкой для задания 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, а также если OpenSSL скомпилирован без поддержки PSK.

Примеры пользовательской настройки наборов шифров

Ниже приведены примеры пользовательской настройки наборов шифров:

- Тестирование строк шифров и разрешение только наборов шифров с PFS
- Переход с AES128 на AES256

Тестирование строк шифров и разрешение только наборов шифров PFS

Чтобы увидеть, какие наборы шифров были выбраны, необходимо установить `DebugLevel=4` в конфигурационном файле или использовать параметр `-vv` для zabbix_sender.

Прежде чем вы получите нужные наборы шифров, может потребоваться поэкспериментировать с параметрами `TLSCipher*`. Неудобно перезапускать Zabbix сервер, прокси или агент много раз только для того, чтобы подстроить параметры `TLSCipher*`. Более удобные варианты — использовать zabbix_sender или команду `openssl`. Покажем оба.

1. Использование zabbix_sender.

Создадим тестовый конфигурационный файл, например `/home/zabbix/test.conf`, с 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 вам понадобятся действительные сертификаты CA и агента, а также PSK. Скорректируйте 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 вы можете усилить безопасность, разрешив только несколько наборов шифров, например только наборы шифров с PFS (Perfect Forward Secrecy). Попробуем разрешить только наборы шифров с PFS, используя параметры `TLSCipher*`.

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 конфигурационный файл `test.conf`:

```
TLSCipherCert=EECDH+aRSA+AES128
TLSCipherPSK=kECDHEPSK+AES128
```

и выполните проверку снова:

```
$ 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, как и ожидалось, и содержат только наборы шифров TLS 1.3 и наборы шифров ECDHE-* для TLS 1.2.

2. TLSCipherAll и TLSCipherAll13 нельзя протестировать с помощью zabbix_sender; они не влияют на значение "certificate and PSK ciphersuites", показанное в примере выше. Чтобы настроить TLSCipherAll и TLSCipherAll13, необходимо экспериментировать с агентом, прокси или сервером.

Итак, чтобы разрешить только наборы шифров PFS, вам может понадобиться добавить до трёх 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 в каждом из них настроен сертификат, а у агента также настроен PSK.

Если в вашей среде Zabbix используется только шифрование на основе PSK и не используются сертификаты, тогда нужен только один параметр:

```
TLSCipherPSK=kECDHEPSK+AES128
```

Теперь, когда вы понимаете, как это работает, вы можете протестировать выбор набора шифров даже вне Zabbix с помощью команды `openssl`. Проверим все три значения 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
```

Возможно, вы предпочтёте `openssl ciphers` с параметром `-V` для более подробного вывода:

```
$ 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
```

Аналогично 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 в качестве встроенного значения по умолчанию для данных. Предположим, что вы используете сертификаты и хотите перейти на AES256 в OpenSSL 1.1.1.

Этого можно добиться, 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, относящиеся к сертификатам, параметры `TLSCipherPSK*` также определены, чтобы избежать их значений по умолчанию, которые включают менее безопасные шифры для более широкой совместимости. Наборы шифров PSK нельзя полностью отключить на сервере/прокси.

А 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
```

1 Использование сертификатов

Обзор

Zabbix может использовать сертификаты RSA 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.

Дополнительную информацию о настройке и эксплуатации внутреннего CA, создании и подписывании запросов на сертификаты, а также отзыве сертификатов см. в руководствах, таких как OpenSSL PKI Tutorial v2.0.

Тщательно продумайте и протестируйте расширения сертификатов. Подробнее см. в разделе Ограничения при использовании расширений сертификатов X.509 v3.

Параметры настройки сертификатов

Для настройки сертификатов в компонентах Zabbix поддерживаются следующие параметры конфигурации.

Параметр	Обязательный	Описание
TLSCAFile	да	Полный путь к файлу, содержащему сертификаты корневого(ых) центра(ов) сертификации (CA) для проверки сертификата другой стороны. При использовании цепочки сертификатов с несколькими элементами сначала указывайте сертификаты CA более низкого уровня, затем сертификаты CA более высокого уровня. Сертификаты от нескольких CA могут быть включены в один файл.
TLSCRLFile	нет	Полный путь к файлу, содержащему списки отзыва сертификатов (CRL).
TLSCertFile	да	Полный путь к файлу, содержащему сертификат. При использовании цепочки сертификатов с несколькими элементами сначала указывайте сертификат сервера, прокси или агента, затем сертификаты CA более низкого уровня и в конце сертификаты CA более высокого уровня.
TLSKeyFile	да	Полный путь к файлу, содержащему закрытый ключ. Убедитесь, что этот файл доступен для чтения только пользователю Zabbix, установив соответствующие права доступа.
TLSServerCertIssuer	нет	Разрешенный издатель сертификата сервера.
TLSServerCertSubject	нет	Разрешенный субъект сертификата сервера.

Примеры настройки

После настройки необходимых сертификатов настройте компоненты Zabbix на использование шифрования на основе сертификатов.

Ниже приведены подробные шаги по настройке:

- Сервер Zabbix
- Прокси Zabbix
- Агент Zabbix

Сервер Zabbix

1. Подготовьте файл сертификата CA.

Чтобы проверять сертификаты другой стороны, сервер Zabbix должен иметь доступ к файлу, содержащему самоподписанные корневые сертификаты CA верхнего уровня. Например, если требуются сертификаты от 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/цепочку сертификатов сервера Zabbix в файл, например `/home/zabbix/zabbix_server.crt`. Первым должен идти сертификат сервера Zabbix, затем — 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/wUtcVIvWXdMIFJMOHw==
-----END CERTIFICATE-----
```

Note:
Используйте только указанные выше атрибуты как для клиентских, так и для серверных сертификатов, 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
```

Zabbix прокси

1. Подготовьте файлы с сертификатами корневого CA верхнего уровня, сертификатом/цепочкой сертификатов Zabbix прокси и закрытым ключом, как описано в разделе Zabbix сервер. Затем соответствующим образом отредактируйте параметры `TLSCAFile`, `TLSCertFile` и `TLSKeyFile` в файле конфигурации Zabbix прокси.

2. Отредактируйте дополнительные параметры TLS в файле конфигурации Zabbix прокси:

- Для активного прокси: `TLSConnect=cert`
- Для пассивного прокси: `TLSAccept=cert`

Note:
Для повышения безопасности прокси вы также можете задать параметры `TLSServerCertIssuer` и `TLSServerCertSubject`. Дополнительную информацию см. в разделе Ограничение допустимого издателя и субъекта сертификата.

Параметры TLS в итоговом файле конфигурации прокси могут выглядеть следующим образом:

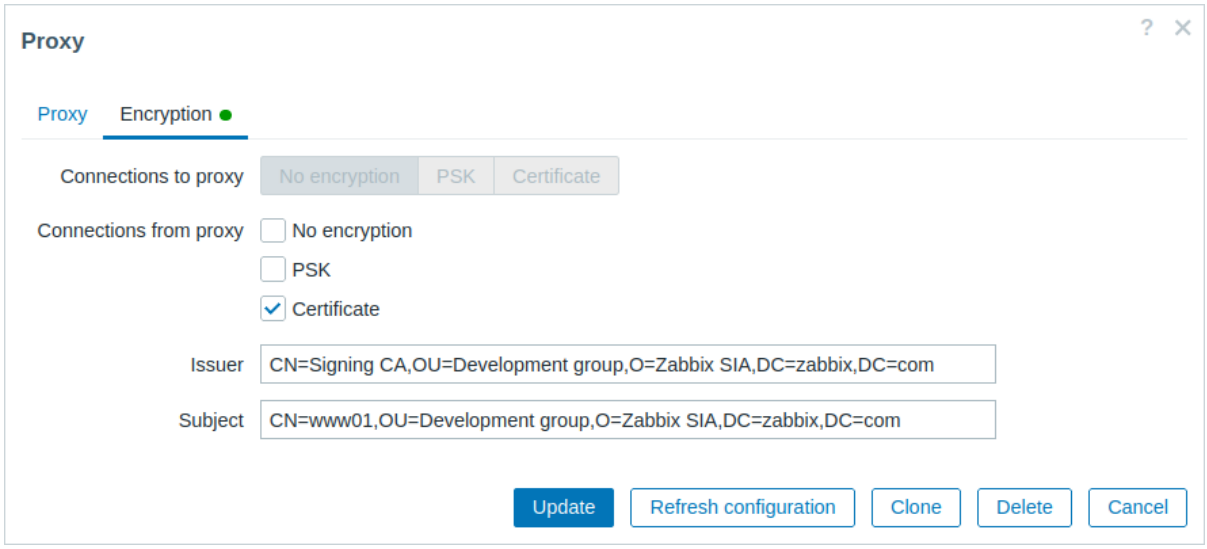
```
TLSConnect=cert
TLSAccept=cert
TLSCAFile=/home/zabbix/zabbix_ca_file.crt
TLSServerCertIssuer=CN=Signing CA,OU=Development group,O=Zabbix SIA,DC=zabbix,DC=com
TLSServerCertSubject=CN=Zabbix server,OU=Development group,O=Zabbix SIA,DC=zabbix,DC=com
TLSCertFile=/home/zabbix/zabbix_proxy.crt
TLSKeyFile=/home/zabbix/zabbix_proxy.key
```

3. Настройте шифрование для этого прокси в веб-интерфейсе Zabbix:

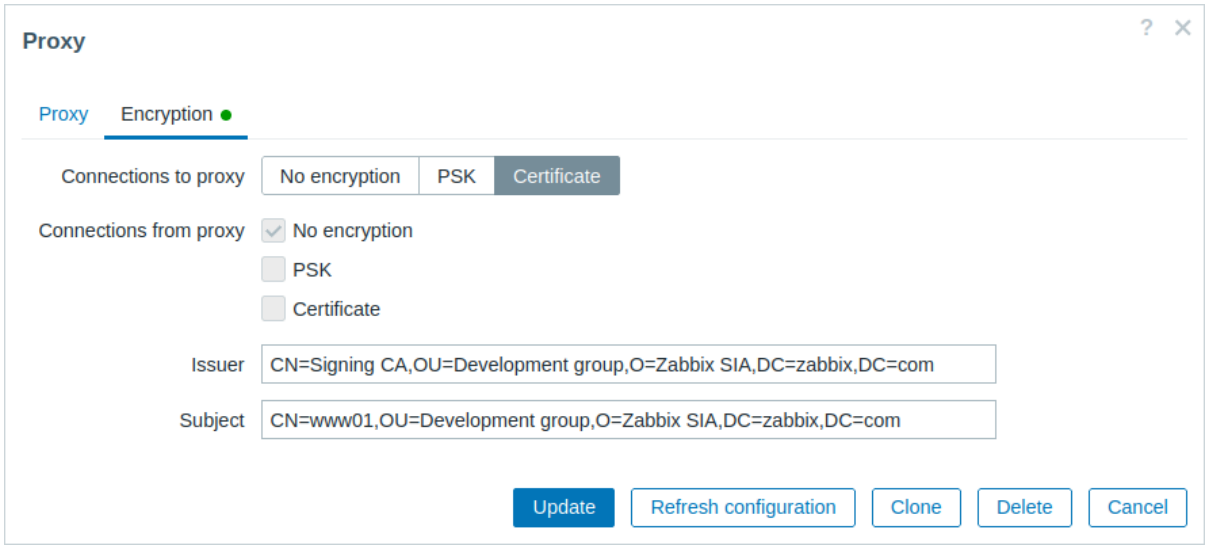
- Перейдите в: *Администрирование* → *Прокси*.
- Выберите прокси и откройте вкладку *Шифрование*.

В приведённых ниже примерах поля *Issuer* и *Subject* заполнены. Дополнительную информацию о том, зачем и как использовать эти поля, см. в разделе Ограничение допустимого издателя и субъекта сертификата.

Для активного прокси:



Для пассивного прокси:



Zabbix агент

1. Подготовьте файлы с корневыми сертификатами CA, сертификатом/цепочкой сертификатов Zabbix агента и закрытым ключом, как описано в разделе Zabbix сервер. Затем соответствующим образом отредактируйте параметры `TLSCAFile`, `TLSCertFile` и `TLSKeyFile` в файле конфигурации Zabbix агента.

2. Отредактируйте дополнительные параметры TLS в файле конфигурации Zabbix агента:

- Для активного агента: `TLSConnect=cert`
- Для пассивного агента: `TLSAccept=cert`

Note:
Для повышения безопасности агента вы можете задать параметры TLSServerCertIssuer и TLSServerCertSubject. Подробнее см. в разделе Ограничение допустимого издателя и субъекта сертификата.

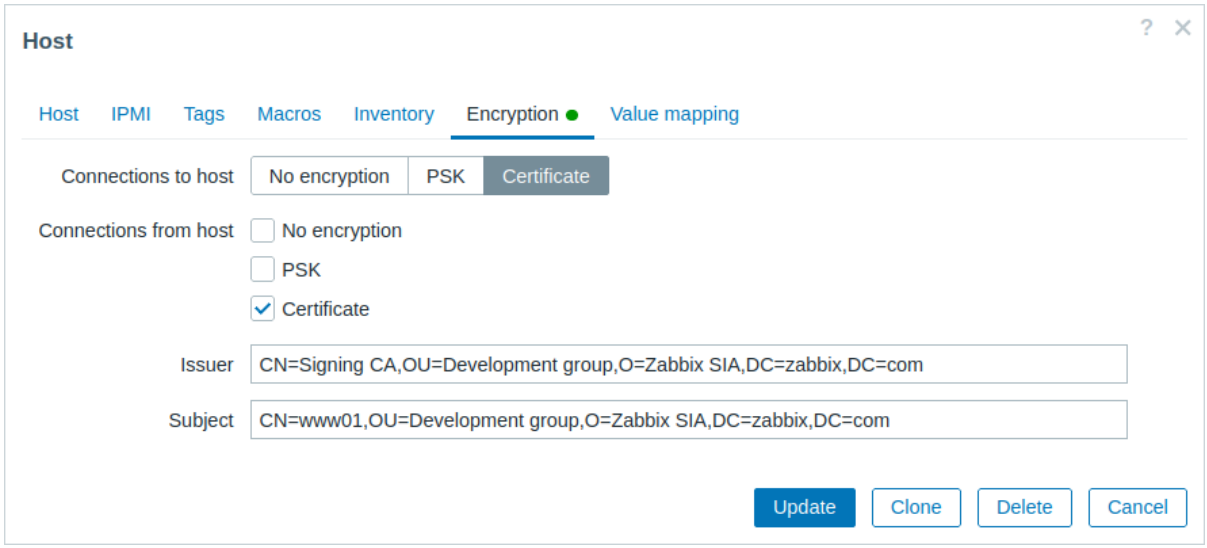
Параметры TLS в итоговом файле конфигурации агента могут выглядеть следующим образом. Обратите внимание, что в примере предполагается, что узел сети контролируется прокси, поэтому он указан как Subject сертификата:

```
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, контролируемого этим агентом.

- Перейдите в: *Сбор данных → Узлы сети*.
- Выберите узел сети и откройте вкладку *Шифрование*.

В приведенном ниже примере поля *Issuer* и *Subject* заполнены. Подробнее о том, зачем и как использовать эти поля, см. в разделе Ограничение допустимого издателя и субъекта сертификата.



Веб-служба Zabbix

1. Подготовьте файлы с сертификатами корневого CA, сертификатом/цепочкой сертификатов веб-службы Zabbix и закрытым ключом, как описано в разделе сервер Zabbix. Затем соответствующим образом измените параметры TLSCAFile, TLSCertFile и TLSKeyFile в файле конфигурации веб-службы Zabbix.

2. Измените дополнительный параметр TLS в файле конфигурации веб-службы Zabbix: TLSAccept=cert

Параметры TLS в итоговом файле конфигурации веб-службы 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 на подключение к веб-службе Zabbix, настроенной для использования TLS, изменив параметр WebServiceURL в файле конфигурации сервера Zabbix:

```
WebServiceURL=https://example.com:443/report
```

Ограничение допустимого издателя и субъекта сертификата

Когда два компонента Zabbix (например, сервер и агент) устанавливают TLS-соединение, они проверяют сертификаты друг друга. Если сертификат другой стороны подписан доверенным центром сертификации (с предварительно настроенным корневым сертификатом в TLSCAFile), является действительным, не истек и проходит другие проверки, то 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 выдать себя за кого угодно (например, сертификат узла сети может быть использован для выдачи себя за сервер). Хотя это может быть приемлемо в небольших средах, где сертификаты подписываются выделенным внутренним центром сертификации и риск подмены невелик, в более крупных средах или средах с повышенными требованиями к безопасности этого может быть недостаточно.

Если ваш корневой центр сертификации выдает сертификаты, которые не должны приниматься Zabbix, или если вы хотите снизить риск подмены, вы можете ограничить допустимые сертификаты, указав их издателя и субъекта.

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 взаимодействовать с сервером Zabbix, если сертификат этого сервера 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 являются необязательными.
- Неуказанная строка означает, что принимается любая строка.
- Строки сравниваются *как есть* и должны совпадать точно.
- Символы UTF-8 поддерживаются. Однако подстановочные знаки (*) или 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).
- Нулевые символы (U+0000) не поддерживаются. Если встречается нулевой символ, сопоставление завершится неудачей.
- Стандарты RFC 4517 и RFC 4518 не поддерживаются.

Например, если строки организации (O) в `Issuer` и `Subject` содержат завершающие пробелы, а строка подразделения организации (OU) в `Subject` содержит двойные кавычки, эти символы должны быть экранированы:

```
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 полей и форматирование

Zabbix следует рекомендациям RFC 4514, который задаёт "обратный" порядок для этих полей: начиная с полей самого нижнего уровня (CN), затем переходя к полям среднего уровня (OU, O) и завершая полями самого верхнего 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 различается в зависимости от используемых параметров, и поэтому 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
```

Тогда вывод будет в обратном порядке, с разделением запятыми, и его можно будет использовать в конфигурационных файлах Zabbix и веб-интерфейсе:

```
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 при использовании расширений сертификатов X.509 v3

При использовании 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 допустимого издателя и субъекта 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 принимать, так и отклонять сертификаты в зависимости от этих сочетаний.

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 клиента) и *serverAuth* (для аутентификации TLS WWW сервера). Например:

- При пассивных проверках, когда агент Zabbix работает как TLS-сервер, атрибут *serverAuth* должен быть включен в сертификат агента.
- При активных проверках, когда агент работает как TLS-клиент, атрибут *clientAuth* должен быть включен в сертификат агента.

Хотя GnuTLS может выдавать предупреждение о нарушении использования ключа, обычно он позволяет продолжить обмен данными, несмотря на эти предупреждения.

Расширение Name Constraints

Поддержка расширения *Name Constraints* различается в разных криптографических библиотеках. Убедитесь, что выбранная вами библиотека поддерживает это расширение. Это расширение может ограничить возможность Zabbix загружать сертификаты CA, если данный 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 конфигурационные файлы и могут быть указаны с помощью параметра `TLSCRLFile` в конфигурационных файлах 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 следующим образом:

```
-----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 в `TLSCRLFile` включён соответствующий CRL (даже если он пустой).

2 Использование общих ключей (pre-shared keys)

Обзор

Каждый общий ключ (pre-shared key, PSK) в Zabbix является парой из:

- несекретной строки идентификации PSK,
- секретного значения строки PSK.

Строкой идентификации PSK является непустая строка UTF-8. Например, «PSK ID 001 Zabbix agentd». Это уникальное имя, по которому компоненты Zabbix ссылаются на этот конкретный общий ключ. Не помещайте чувствительную информацию в строку идентификации PSK - она передаётся по сети незашифрованной.

Значением PSK является сложно угадываемая строка из шестнадцатеричных цифр, например: «e560cb0d918d26d31b4f642181f5f570a

Ограничения размера

В Zabbix существуют ограничения размера для идентификатора и значения PSK; в некоторых случаях криптографическая библиотека может иметь более низкий предел:

Компонент	Макс. размер идентификатора PSK	Мин. размер значения PSK	Макс. размер значения PSK
Zabbix	128 символов UTF-8	128 бит (16-байтный PSK, вводится как 32 шестнадцатеричные цифры)	2048 бит (256-байтный PSK, вводится как 512 шестнадцатеричных цифр)
GnuTLS	128 байт (может включать символы UTF-8)	-	2048 бит (256-байтный PSK, вводится как 512 шестнадцатеричных цифр)
OpenSSL 1.0.x, 1.1.0	127 байт (может включать символы UTF-8)	-	2048 бит (256-байтный PSK, вводится как 512 шестнадцатеричных цифр)
OpenSSL 1.1.1	127 байт (может включать символы UTF-8)	-	512 бит (64-байтный PSK, вводится как 128 шестнадцатеричных цифр)
OpenSSL 1.1.1a and later	127 байт (может включать символы UTF-8)	-	2048 бит (256-байтный PSK, вводится как 512 шестнадцатеричных цифр)

Attention:
Веб-интерфейс Zabbix позволяет настраивать 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 пределы, пользователь сам несет ответственность за настройку идентификатора и значения PSK с допустимой длиной для этих компонентов.
Превышение ограничений длины приводит к сбоям связи между компонентами Zabbix.

Перед тем как сервер Zabbix подключается к агент с использованием PSK, сервер ищет идентификатор и значение PSK, настроенные для этого агент, в базе данных (фактически в кэше конфигурации). При получении соединения агент использует идентификатор и значение PSK из своего файла конфигурации. Если обе стороны имеют одинаковую строку идентификатора PSK и одинаковое значение PSK, соединение может быть успешно установлено.

Attention:
Каждый идентификатор PSK должен быть связан только с одним значением. Пользователь сам несет ответственность за то, чтобы не было двух PSK с одной и той же 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, 256-битный (32 байта) PSK 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 приведённая выше команда "psktool" создаёт файл базы данных с идентификатором PSK и связанным с ним PSK. Zabbix ожидает, что в файле PSK будет только PSK, поэтому из файла следует удалить строку идентификатора и двоеточие (':').

Настройка PSK для взаимодействия сервер-агент (пример)

На узле сети агента запишите значение PSK в файл, например, `/home/zabbix/zabbix_agentd.psk`. Файл должен содержать PSK в первой текстовой строке, например:

```
1f87b595725ac58dd977beef14b97461a7c1045b9a1c963065002c5473194952
```

Установите права доступа к файлу PSK — он должен быть доступен для чтения только пользователю Zabbix.

Отредактируйте параметры TLS в конфигурационном файле агента `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 соединения с использованием PSK. Идентификатор PSK будет "PSK 001".

Перезапустите агент. Теперь вы можете проверить соединение с помощью `zabbix_get`, например:

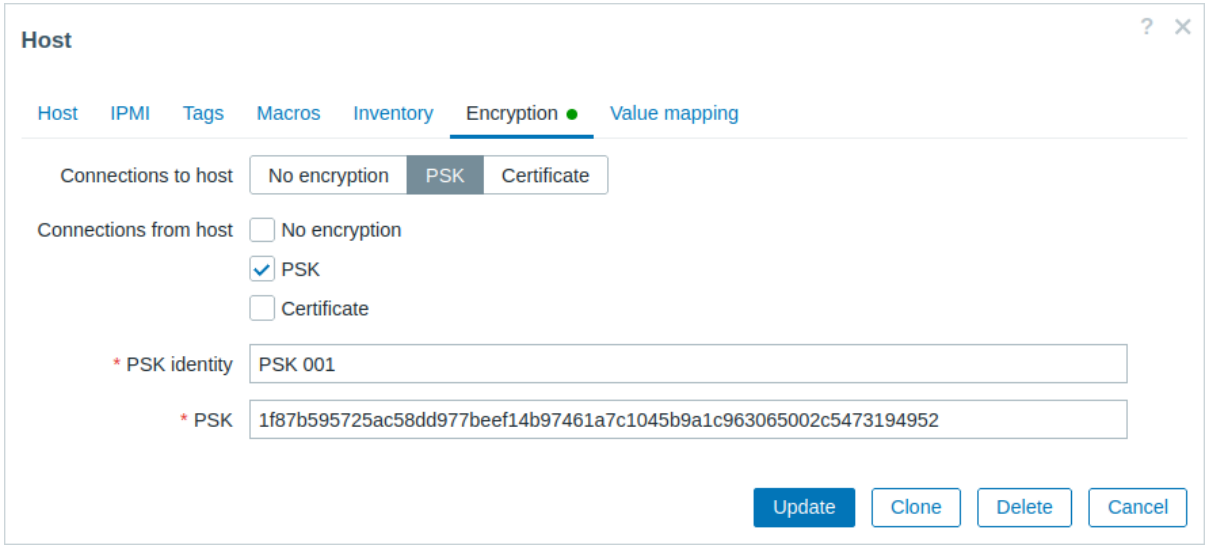
```
zabbix_get -s 127.0.0.1 -k "system.cpu.load[all,avg1]" --tls-connect=psk --tls-psk-identity="PSK 001" --tl
```

(Чтобы минимизировать время простоя, смотрите, как изменить тип соединения в разделе Управление шифрованием соединений).

Настройте шифрование PSK для этого агента в веб-интерфейсе Zabbix:

- Перейдите в: *Сбор данных → Узлы сети*
- Выберите узел сети и откройте вкладку **Шифрование**

Пример:



Все обязательные поля ввода отмечены красной звездочкой.

Когда кэш конфигурации будет синхронизирован с базой данных, новые соединения будут использовать PSK. Проверьте журналы сервера и агента на наличие сообщений об ошибках.

Настройка PSK для связи сервер - активный прокси (пример)

На прокси запишите значение PSK в файл, например, `/home/zabbix/zabbix_proxy.psk`. Файл должен содержать PSK в первой текстовой строке, например:

```
e560cb0d918d26d31b4f642181f5f570ad89a390931102e5391d08327ba434e9
```

Установите права доступа к файлу PSK — он должен быть доступен для чтения только пользователю Zabbix.

Измените параметры TLS в файле конфигурации прокси `zabbix_proxy.conf`, например, задайте:

```
TLSConnect=psk
TLSPSKFile=/home/zabbix/zabbix_proxy.psk
TLSPSKIdentity=PSK 002
```

Прокси будет подключаться к серверу с использованием PSK. Идентификатор PSK будет "PSK 002".

(Чтобы минимизировать время простоя, смотрите, как изменить тип соединения в разделе Управление шифрованием соединений).

Настройте PSK для этого прокси в веб-интерфейсе Zabbix. Перейдите в *Administration→Proxies*, выберите прокси, откройте вкладку "Encryption". В поле "Connections from proxy" отметьте PSK. Вставьте в поле "PSK identity" значение "PSK 002", а в поле "PSK" — "e560cb0d918d26d31b4f642181f5f570ad89a390931102e5391d08327ba434e9". Нажмите "Update".

Перезапустите прокси. Он начнет использовать зашифрованные соединения с сервером на основе PSK. Проверьте журнальные файлы сервера и прокси на наличие сообщений об ошибках.

Для пассивного прокси процедура очень похожа. Единственное отличие — задайте `TLSAccept=psk` в файле конфигурации прокси и установите "Connections to proxy" в веб-интерфейсе Zabbix в значение PSK.

3 Устранение неполадок

Общие рекомендации

- Начните с понимания того, какой компонент выступает в роли TLS-клиента, а какой — в роли TLS-сервера в проблемном случае.
  Zabbix сервер, прокси и агенты, в зависимости от взаимодействия между ними, могут работать как TLS-серверы и клиенты.
  Например, Zabbix сервер, подключаясь к агенту для пассивной проверки, выступает в роли TLS-клиента. Агент при этом находится в роли TLS-сервера.
  Zabbix агент, запрашивая список активных проверок у прокси, выступает в роли TLS-клиента. Прокси при этом находится в роли TLS-сервера.
  Утилиты `zabbix_get` и `zabbix_sender` всегда выступают в роли TLS-клиентов.
- Zabbix использует взаимную аутентификацию.
  Каждая сторона проверяет своего партнера и может отклонить соединение.
  Например, Zabbix сервер, подключаясь к агенту, может немедленно закрыть соединение, если сертификат агента недействителен. И наоборот — Zabbix агент, принимая соединение от сервера, может закрыть соединение, если сервер не является доверенным для агента.
- Изучите журналы на обеих сторонах — у TLS-клиента и TLS-сервера.
  Сторона, которая отклоняет соединение, может записать точную причину отказа. Другая сторона часто сообщает лишь об общей ошибке (например, "Connection closed by peer", "connection was non-properly terminated").
- Иногда неправильно настроенное шифрование приводит к запутанным сообщениям об ошибках, которые никак не указывают на реальную причину.
  В подразделах ниже мы пытаемся привести подборку сообщений и возможных причин (далеко не исчерпывающую), которая может помочь в устранении неполадок.
  Обратите внимание, что разные криптографические библиотеки (OpenSSL, GnuTLS) часто выдают разные сообщения об ошибках в одинаковых проблемных ситуациях.
  Иногда сообщения об ошибках зависят даже от конкретного сочетания криптографических библиотек на обеих сторонах.

1 Проблемы с типом соединения или 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)

```
Get value from agent failed: zbx_tls_connect(): gnutls_handshake() failed: \
    -110 The TLS connection was non-properly terminated.
```

В журнале сервера или прокси (с *OpenSSL* 1.0.2c)

```
Get value from agent failed: TCP connection successful, cannot establish TLS to [[127.0.0.1]:10050]: \
    Connection closed by peer. Check allowed connection types and access rights
```

Одна сторона подключается с сертификатом, но другая сторона принимает только PSK, или наоборот

В любом журнале (с *GnuTLS*):

```
failed to accept an incoming connection: from 127.0.0.1: zbx_tls_accept(): gnutls_handshake() failed:\
    -21                              .
```

В любом журнале (с *OpenSSL* 1.0.2c):

```
failed to accept an incoming connection: from 127.0.0.1: TLS handshake returned error code 1:\
    file .\ssl\s3_srvr.c line 1411: error:1408A0C1:SSL routines:ssl3_get_client_hello:no shared cipher:\
    TLS write fatal alert "handshake failure"
```

Попытка использовать Zabbix sender, скомпилированный с поддержкой TLS, для отправки данных на сервер/прокси Zabbix, скомпилированный без TLS

В журнале на стороне подключения:

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:'                     '
...              : TCP     ,              TLS   [[localhost]:10051]:
```

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(): [0x00000000]                .'
...              : TCP     ,              TLS   [[192.168.1.2]:10051]:        -    SSL_connect(): [0x000000
```

В журнале принимающей стороны:

```
...                        :   127.0.0.1:       TLS
```

Одна сторона подключается с PSK, но другая сторона использует LibreSSL или была скомпилирована без поддержки шифрования

LibreSSL не поддерживает PSK.

В журнале подключающейся стороны:

```
...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 отключённой поддержкой PSK

В журнале подключающейся стороны:

```
...TCP successful, cannot establish TLS to [[192.168.1.2]:10050]: SSL_connect() set result code to SSL_ERR
```

В журнале принимающей стороны:

```
...failed to accept an incoming connection: from 192.168.1.2: TLS handshake set result code to 1: file ssl
```

2 Проблемы с сертификатом

OpenSSL используется с CRL, и для некоторого CA в цепочке сертификатов его CRL не включён в `TLSCRLFile`

В журнале TLS-сервера в случае узла *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 TLS-сервера в случае узла *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 истёк или истекает во время работы сервера

OpenSSL, в журнале сервера:

- до истечения срока:

```
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:

```
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 в том, что при действительном CRL отозванный сертификат отображается как ”certificate revoked”. Когда срок действия CRL истекает, сообщение об ошибке меняется на ”certificate expired”, что может вводить в заблуждение.

GnuTLS, в журнале сервера:

- до и после истечения срока 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 совпадают в корневом сертификате CA, но они не могут совпадать в сертификате сервера. (То же самое относится к сертификатам прокси и агента.)

Чтобы проверить, содержит ли сертификат одинаковые записи Issuer и Subject, выполните:

```
openssl x509 -in <yourcertificate.crt> -noout -text
```

Допустимо, чтобы корневой сертификат (верхнего уровня) имел одинаковые значения Issuer и Subject.

3 Проблемы с PSK

PSK содержит нечётное количество шестнадцатеричных цифр

Прокси или агент не запускается, в журнале прокси или агента появляется сообщение:

```
invalid PSK in file "/home/zabbix/zabbix_proxy.psk"
```

Строка идентификации PSK длиннее 128 байт передаётся в GnuTLS

В журнале на стороне TLS-клиента:

```
gnutls_handshake() failed: -110 TLS-                    .
```

В журнале на стороне TLS-сервера:

```
gnutls_handshake() failed: -90                SRP             .
```

Используется слишком длинное значение PSK с OpenSSL 1.1.1

В журнале на стороне, устанавливающей соединение:

```
...OpenSSL library (version OpenSSL 1.1.1  11 Sep 2018) initialized
...
...In zbx_tls_connect(): psk_identity:"PSK 1"
...zbx_psk_client_cb() requested PSK identity "PSK 1"
...End of zbx_tls_connect():FAIL error:'SSL_connect() set result code to SSL_ERROR_SSL: file ssl\statem\ex
```

В журнале на принимающей стороне:

```
...Message from 123.123.123.123 is missing header. Message ignored.
```

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 (64-байтный PSK, введённый как 128 шестнадцатеричных цифр).

См. также: Ограничения размера значения

16 Веб-интерфейс

Обзор Для легкого доступа к Zabbix из любого места и с любой платформы, предусмотрен веб-интерфейс.

Note:
Если Вы используете несколько веб-интерфейсов, убедитесь, что локали и необходимые библиотеки (LDAP, SAML и т.д.) установлены и настроены одинаково для каждого из веб-интерфейсов.

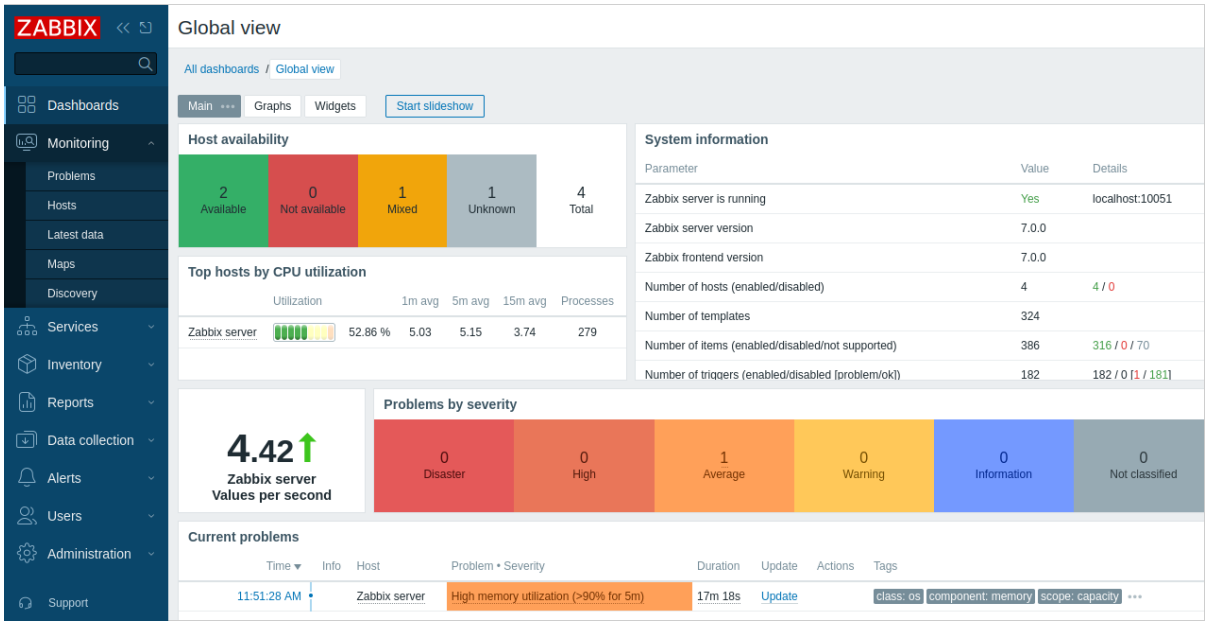
Помощь по веб-интерфейсу Ссылка на помощь предоставляется в формах веб-интерфейса Zabbix с прямыми ссылками на соответствующие части документации.

1 Меню

Обзор

Вертикальное меню на боковой панели обеспечивает доступ к различным разделам веб-интерфейса Zabbix.

Меню темно-синего цвета в теме по умолчанию.

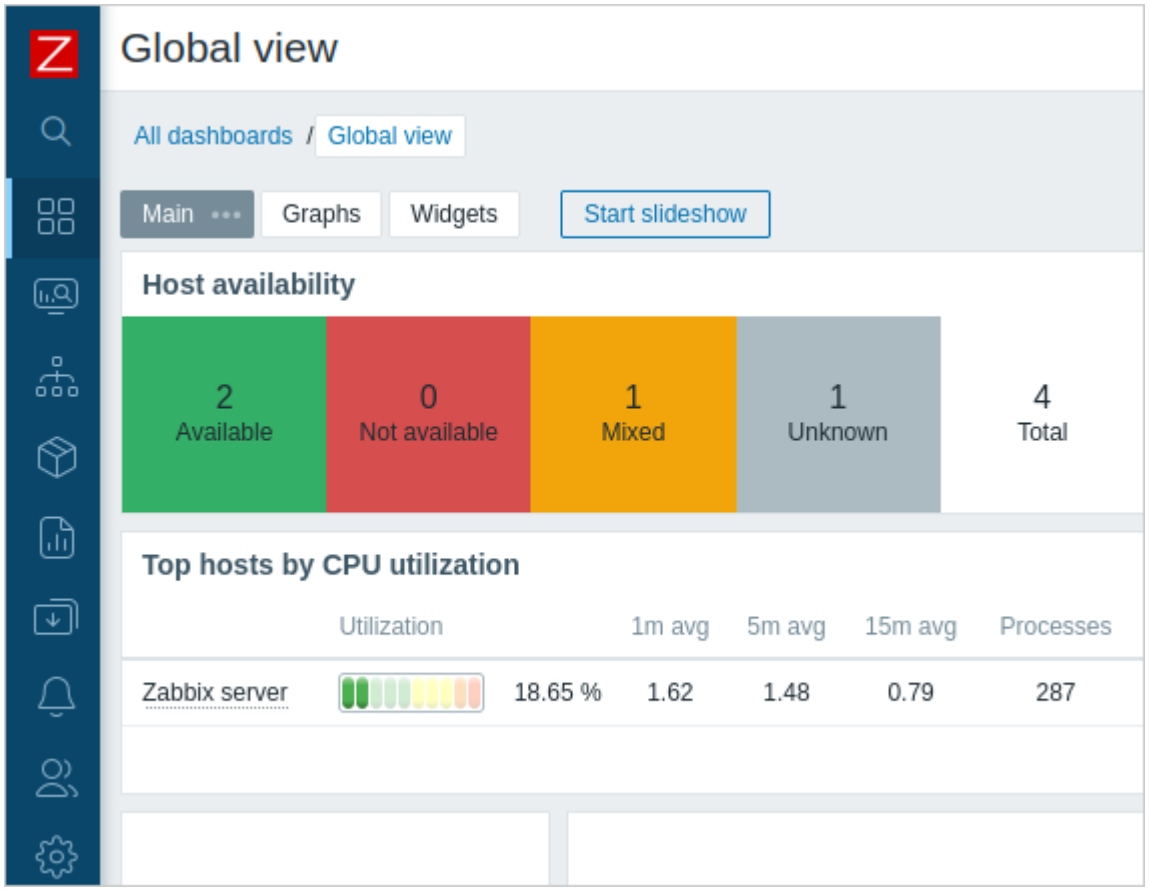


Работа с меню

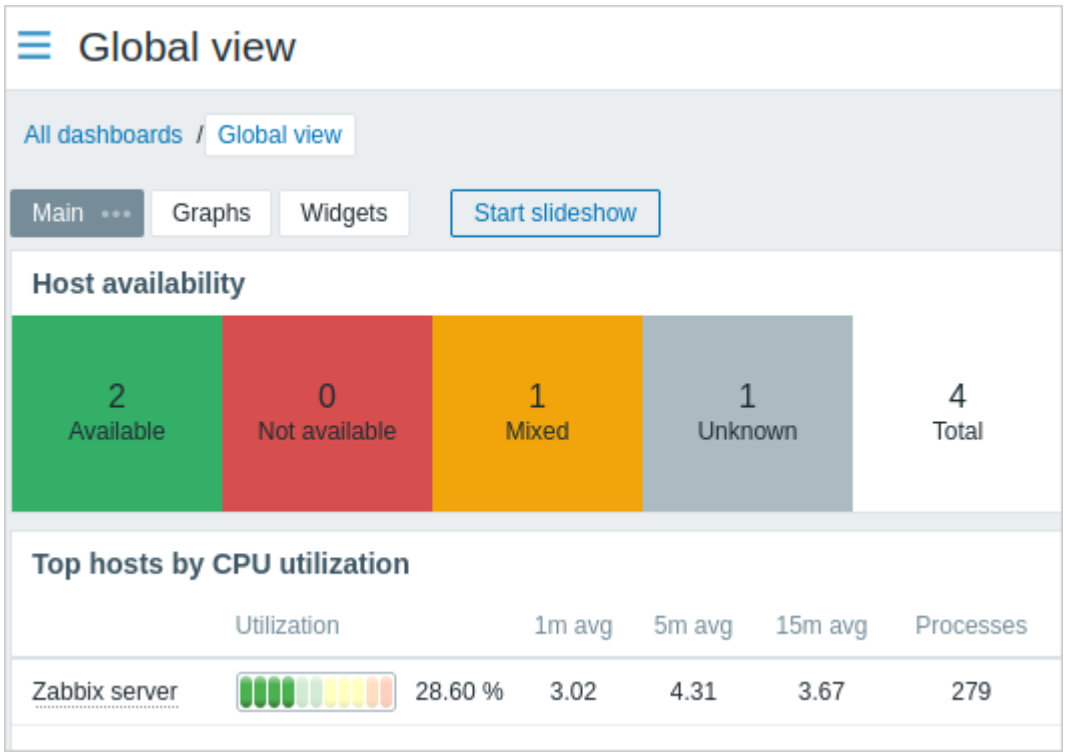
Глобальное окно поиска расположено под логотипом Zabbix.

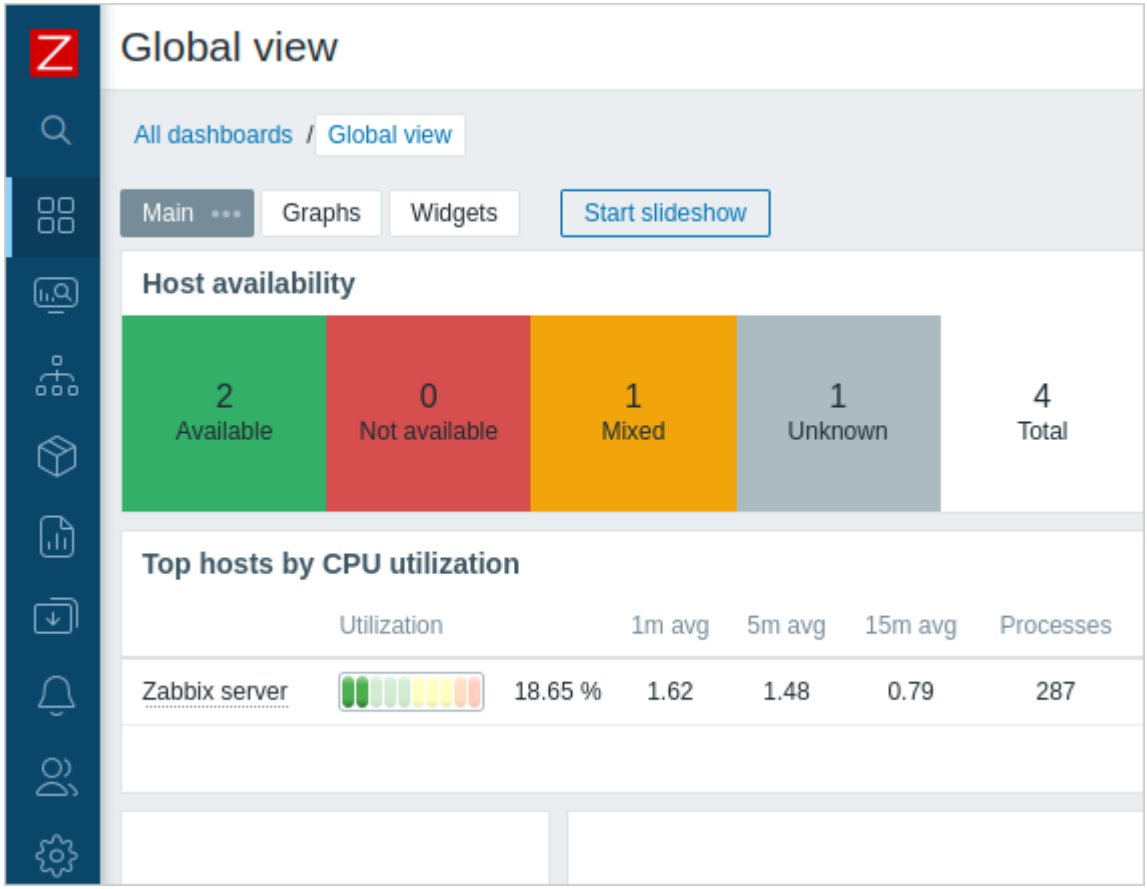
Меню может быть свернуто или скрыто полностью:

- Чтобы свернуть, нажмите на  рядом с логотипом Zabbix. В свернутом меню видны только значки.

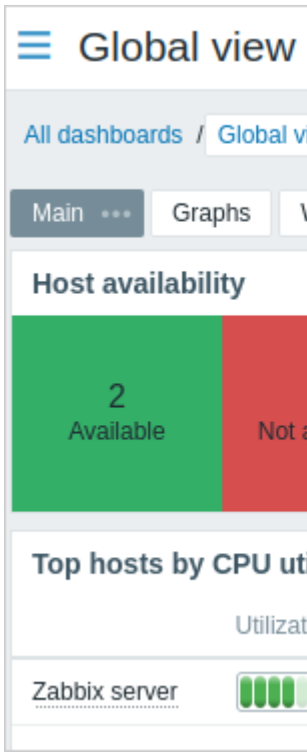


- Чтобы скрыть, нажмите на  рядом с логотипом Zabbix





Свернутое меню с иконками.



Скрытое меню.

Свернутое меню

Когда меню свернуто до значков, полное меню появляется снова при наведении курсора мыши. Обратите внимание, что оно появляется поверх содержимого страницы; чтобы переместить содержимое страницы вправо, нужно нажать на кнопку развернуть. Если переместить курсор мыши за пределы полного меню, меню снова свернется через две секунды.

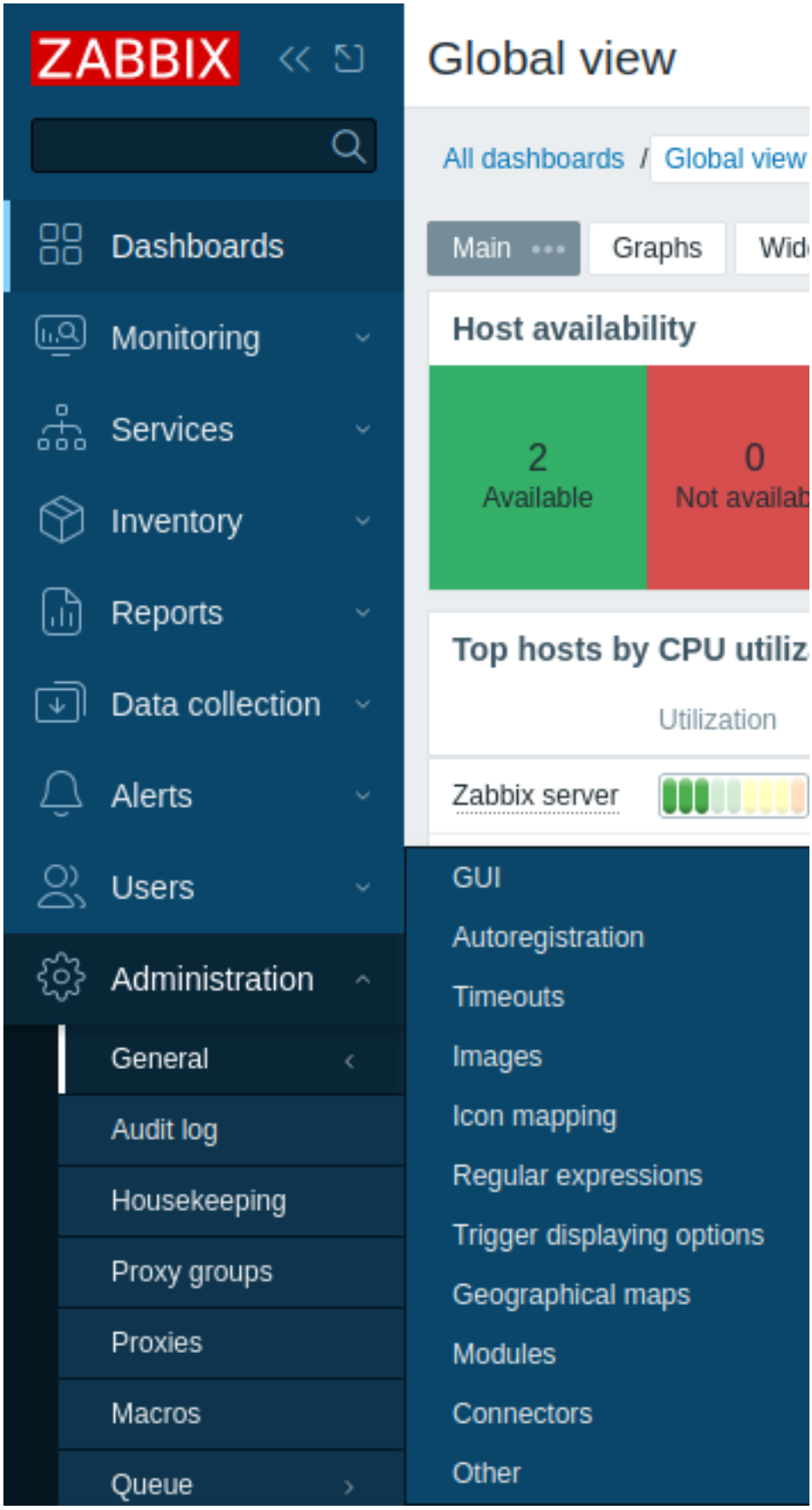
Вы также можете развернуть свернутое меню, нажав клавишу Tab. Повторное нажатие клавиши Tab выделит следующий элемент меню.

Скрытое меню

Даже когда меню полностью скрыто, полное меню находится всего в одном щелчке мыши — достаточно нажать на значок «бургер»  . Обратите внимание, что оно снова появляется поверх содержимого страницы; чтобы сдвинуть содержимое страницы вправо, необходимо отобразить меню, нажав кнопку показа боковой панели.

Уровни меню

В меню может быть до трех уровней.



Контекстные меню

В дополнение к главному меню Zabbix предоставляет контекстные меню для хоста, item иevent для быстрого доступа к часто используемым объектам, таким как последние значения, простой график, конфигурационная форма, связанные скрипты или внешние ссылки.

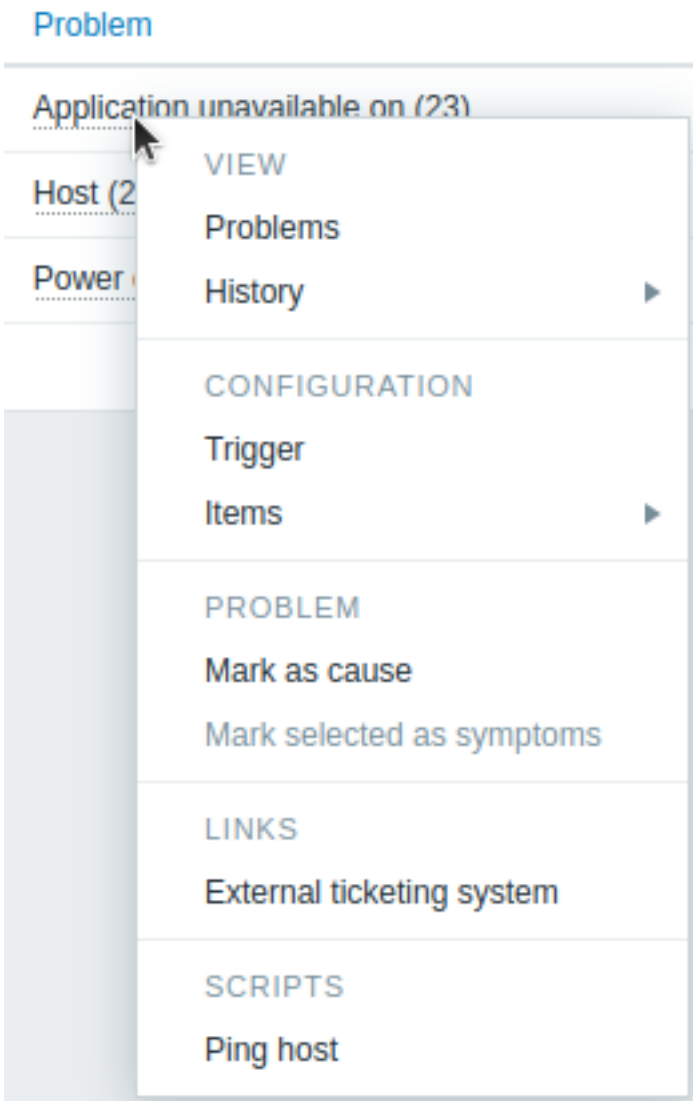
Контекстные меню доступны при нажатии на узел сети, элемент или имя проблемы/триггера в поддерживаемых местах.

1 Меню событий

Обзор

Меню событий содержит ярлыки для действий или разделов веб-интерфейса, которые часто требуются для события.

Меню событий можно открыть, щелкнув по названию события.



Содержимое

Контекстное меню события содержит шесть разделов: *Просмотр*, *Действия*, *Конфигурация*, *Проблема*, *Ссылки* и *Скрипты*. Для сущностей, которые не настроены, ссылки отключены и отображаются серым цветом. Разделы *Скрипты* и *Ссылки* отображаются, если соответствующие сущности настроены.

Раздел *Просмотр* содержит ссылки для:

- **Проблемы** — открывает список неустраненных проблем базового триггера;
- **История** — ведет к графику/истории элемента данных в разделе *Последние данные* для базового элемента данных (элементов данных). Если триггер использует несколько элементов данных, ссылки будут доступны для каждого из них.

Раздел *Действия* доступен только в виджетах *Обзор триггеров*. Он содержит ссылку для:

- **Обновить проблему** — открывает экран обновления проблемы.

Раздел *Конфигурация* содержит ссылки на конфигурацию:

- **Триггера**, вызвавшего проблему;
- **Элементов данных**, используемых в выражении триггера.

Note:
Обратите внимание, что раздел конфигурации доступен только пользователям Admin и Super admin.

Раздел *Проблема* содержит параметры для:

- **Отметить как причину** — отметить проблему как причину;
- **Отметить выбранные как симптомы** — отметить выбранные проблемы как симптомы этой проблемы.

Раздел *Ссылки* содержит ссылки для:

- доступа к настроенному URL триггера;
- доступа к пользовательским ссылкам, настроенным в Глобальных скриптах (с областью действия 'Manual event action' и типом 'URL');
- доступа к настроенному внешнему тикету для проблемы (см. параметр *Include event menu entry* при настройке вебхуков).

Раздел *Скрипты* содержит ссылки для выполнения глобального скрипта (с областью действия *Manual event action*). Эта возможность может быть полезна для запуска скриптов, используемых для управления тикетами проблем во внешних системах.

Поддерживаемые расположения

Контекстное меню события доступно при нажатии на проблему или имя события в различных разделах веб-интерфейса, например:

- Виджеты панелей мониторинга, такие как виджет *Проблемы*, виджет *Обзор триггеров* и т. д.
- Мониторинг → Проблемы
- Мониторинг → Проблемы → Сведения о событии
- Отчеты → Топ 100 триггеров (глобальные скрипты и доступ к внешнему тикету в этом расположении не поддерживаются)

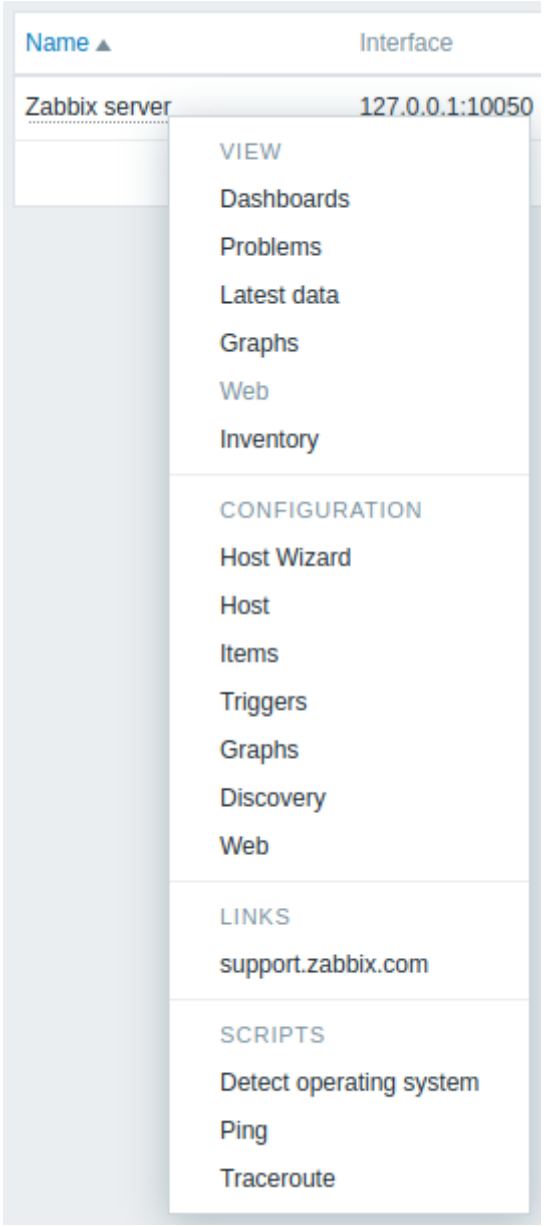
2 Меню узла сети

Обзор

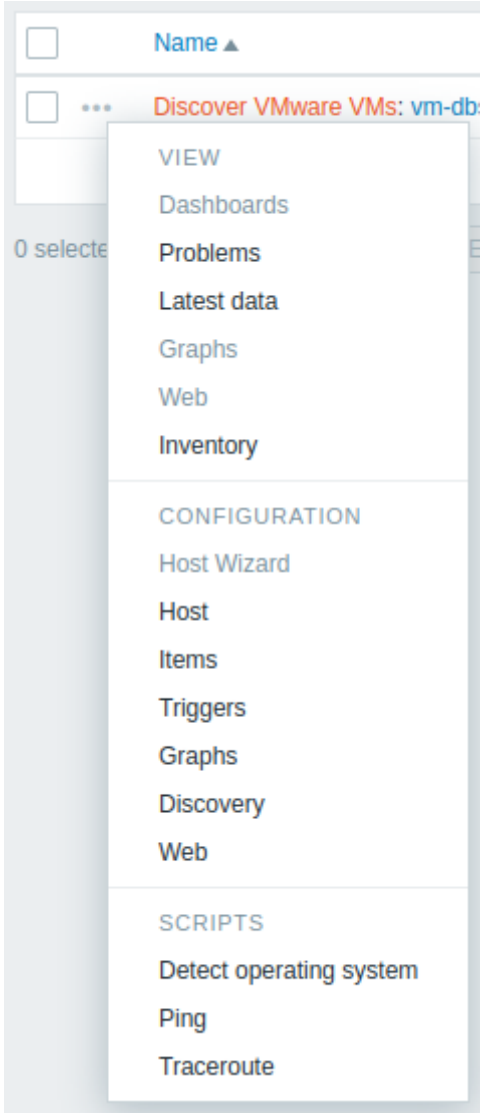
Меню узла сети содержит ярлыки для действий или разделов веб-интерфейса, которые часто требуются для узла сети.

Меню узла сети можно открыть, щелкнув по имени узла сети или по значку с тремя точками, в зависимости от раздела веб-интерфейса, например:

Мониторинг → *Последние данные*



Сбор данных → *Узлы сети*



Содержимое

Контекстное меню узла сети содержит четыре раздела: *Просмотр*, *Конфигурация*, *Ссылки* и *Скрипты*. Для сущностей, которые не настроены, ссылки отключены и отображаются серым цветом. Разделы *Скрипты* и *Ссылки* отображаются, если их сущности настроены.

Раздел *Просмотр* содержит ссылки на:

- **Панели** — открывает виджеты и графики.
- **Проблемы** — открывает раздел *Проблемы* со списком неустраненных проблем базового триггера.
- **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 инвентаризационные данные текущего узла сети.

Раздел *Конфигурация* содержит ссылки на:

- **Мастер узла сети** — открывает Host Wizard для текущего узла сети (отключено для обнаруженных узлов сети).
- **Узел сети** — форма 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 низкоуровневого обнаружения текущего узла сети.
- **Веб** — список веб-сценариев текущего узла сети.

> **Note:**
> Обратите внимание, что раздел конфигурации доступен только пользователям Admin и Super admin.

Раздел *Ссылки* содержит ссылки для:

- доступа к настроенному URL триггера.
- доступа к пользовательским ссылкам, настроенным в Глобальных скриптах (с областью действия *Manual host action* и типом 'URL').

Раздел *Скрипты* позволяет выполнять глобальные скрипты, настроенные для текущего узла сети. Чтобы быть доступными в меню узла сети, для этих скриптов должна быть задана область действия *Manual host action*.

Поддерживаемые расположения

Меню узла сети доступно по щелчку на имени узла сети в различных разделах веб-интерфейса, например:

- Виджеты панелей мониторинга, такие как Проблемы, Лучшие элементы данных, Обзор триггеров и т. д.
- Мониторинг → Проблемы
- Мониторинг → Проблемы → Сведения о событии
- Мониторинг → Узлы сети
- Мониторинг → Узлы сети → Веб-мониторинг
- Мониторинг → Последние данные
- Мониторинг → Карты сети
- Инвентаризация → Узлы сети
- Отчёты → Топ 100 триггеров

Меню узла сети доступно по щелчку на значке с тремя точками в Сбор данных → Узлы сети.

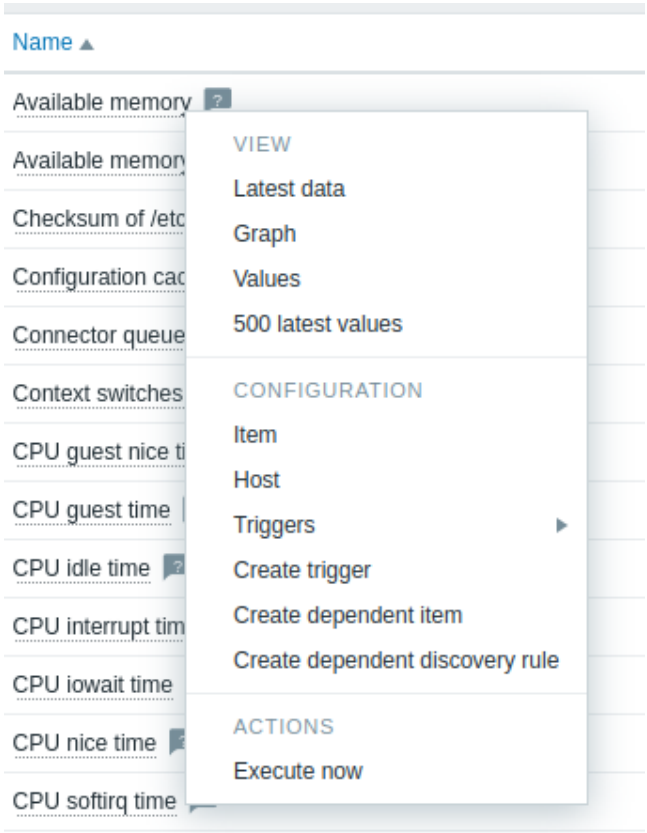
3 Меню элемента данных

Обзор

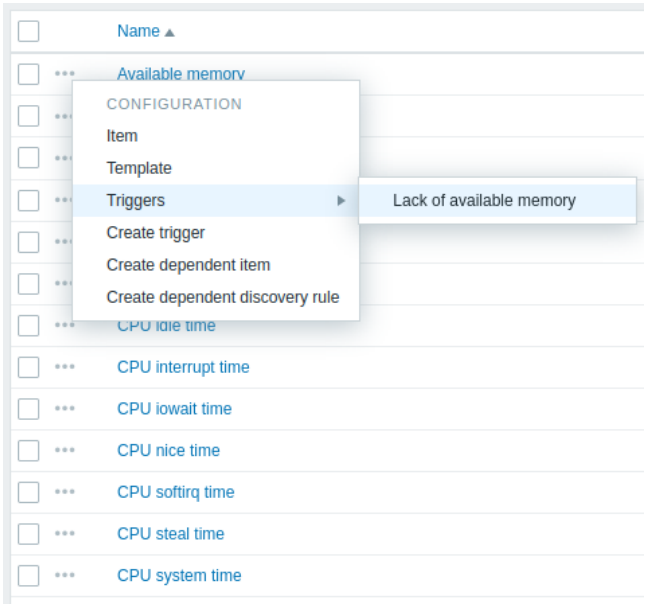
Меню элемента данных содержит ярлыки для действий или разделов веб-интерфейса, которые часто требуются для элемента данных.

Меню элемента данных можно открыть, щелкнув по имени элемента данных или по значку с тремя точками, в зависимости от раздела веб-интерфейса, например:

Мониторинг → Последние данные



Сбор данных → Шаблоны → Элементы данных



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 для текущего элемента данных.

Раздел *Конфигурация* (доступен только пользователям типа *Admin* и *Super admin*) содержит следующие параметры:

- **Элемент данных** — открывает форму настройки элемента данных текущего элемента данных;
- **Шаблон** — открывает форму настройки шаблона шаблона, к которому относится элемент данных (доступно только при открытии меню элемента данных из *Сбор данных → Шаблоны → Элементы данных*);
- **Узел сети** — открывает форму настройки узла сети узла сети, к которому относится элемент данных;
- **Триггеры** — при наведении курсора открывает список триггеров элемента данных, если они есть; щелчок по триггеру открывает его форму 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, где текущий элемент данных выступает как основной элемент данных;
- **Создать зависимое правило обнаружения** — открывает форму настройки правила обнаружения для создания зависимого правила обнаружения, где текущий элемент данных выступает как основной элемент данных.

Раздел *Действия* содержит следующий параметр:

- **Выполнить сейчас** — выполняет проверку для немедленного получения нового значения элемента данных.

Поддерживаемые расположения

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.

2 Разделы веб-интерфейса

Структура меню Меню веб-интерфейса Zabbix имеет следующую структуру:

- Панели
- Мониторинг
  - Проблемы
  - Узлы сети
  - Последние данные
  - Карты
  - Обнаружение
- Услуги
  - Услуги
  - SLA
  - Отчет по SLA
- Инвентаризация
  - Обзор
  - Узлы сети
- Отчеты
  - Системная информация
  - Отчеты по расписанию
  - Отчет о доступности
  - Топ 100 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 услуг
    - Действия обнаружения
    - Действия авторегистрации
    - Внутренние действия
  - Способы оповещения
  - Скрипты
- Пользователи
  - Группы пользователей
  - Роли пользователей
  - Пользователи
  - Токены API
  - Аутентификация
- Администрирование
  - Общие
    - GUI
    - Авторегистрация
    - Тайм-ауты
    - Изображения
    - Сопоставление иконок
    - Регулярные выражения
    - Параметры отображения триггеров

        * Географические карты
        * Модули
        * Коннекторы
        * Прочее
    - Журнал аудита
    - Очистка истории
    - Группы прокси
    - Прокси
    - Макросы
    - Очередь
        * Обзор очереди
        * Обзор очереди по прокси
        * Подробности очереди

1 Панели мониторинга

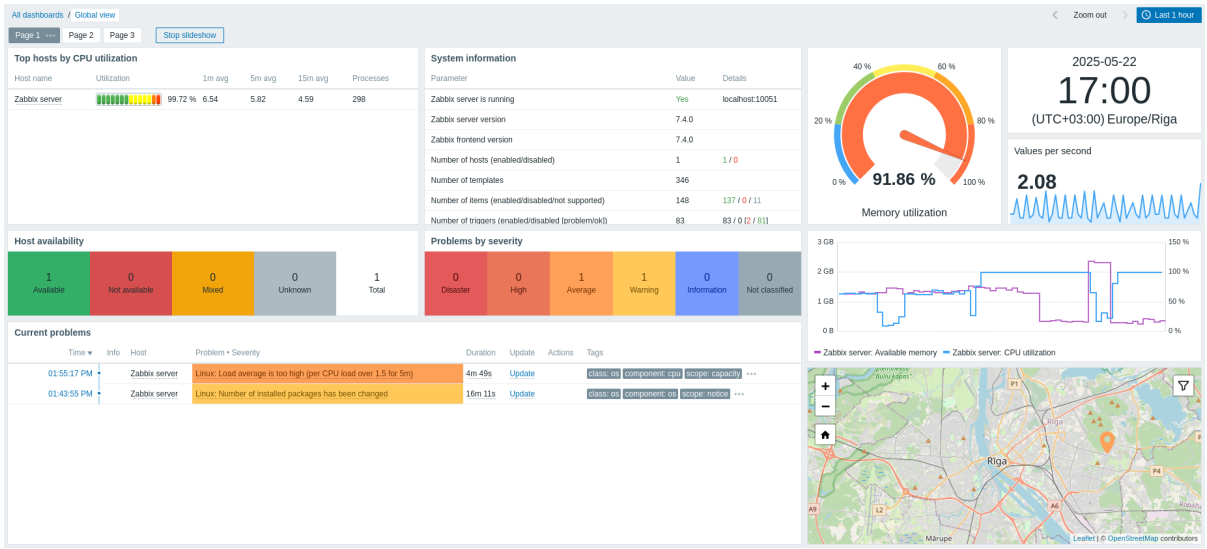
Обзор

Раздел *Панели* предназначен для отображения сводок всей важной информации на **панели**.

Хотя одновременно может отображаться только одна панель, можно настроить несколько панелей. Каждая панель может содержать одну или несколько страниц, которые можно переключать в режиме слайд-шоу.

Страница панели состоит из виджетов, и каждый виджет предназначен для отображения информации определенного типа и из определенного источника, например сводки, карты, графика, часов и т. д.

Доступ к узлам сети в виджетах зависит от прав доступа.



Страницы и виджеты добавляются на панель и редактируются в режиме редактирования панели. Страницы можно просматривать и переключать в режиме просмотра.

Период времени, отображаемый в виджетах графиков, управляется селектором периода времени, расположенным над виджетами. На его метке отображается текущий выбранный период времени. Щелчок по метке позволяет развернуть или свернуть селектор периода времени.

Размер панели

Минимальная ширина панели составляет 1200 пикселей. Панель не будет сжиматься меньше этой ширины; вместо этого, если окно браузера меньше, будет отображаться горизонтальная полоса прокрутки.

Максимальная ширина панели равна ширине окна браузера. Виджеты панели растягиваются по горизонтали, чтобы соответствовать размеру окна. В то же время виджет панели не может быть растянут по горизонтали за пределы окна.

По горизонтали панель состоит из 72 столбцов всегда одинаковой ширины, которые динамически растягиваются/сжимаются (но не менее чем до общей ширины 1200 пикселей).

По вертикали панель может содержать максимум 64 строки; каждая строка имеет фиксированную высоту 70 пикселей.

Таким образом, виджет может иметь ширину до 72 столбцов и высоту до 64 строк.

Просмотр панелей

Чтобы просмотреть все настроенные панели, нажмите *All dashboards* сразу под заголовком раздела.



Панели отображаются с меткой sharing:

- *My* — указывает на панель, принадлежащую текущему пользователю
- *Shared* — указывает на общедоступную панель или частную панель, предоставленную любому пользователю или группе пользователей

Обратите внимание, что метки общего доступа показываются только для панелей, принадлежащих текущему пользователю. Панели, принадлежащие другим пользователям, отображаются без меток общего доступа.

Фильтр, расположенный справа над списком, позволяет фильтровать панели по имени и по признаку создания текущим пользователем.

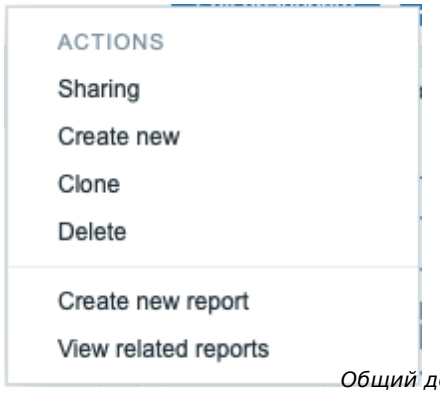
Чтобы удалить одну или несколько панелей, установите флажки у соответствующих панелей и нажмите *Delete* под списком.

Просмотр панели

Чтобы просмотреть отдельную панель, нажмите на её имя в списке панелей.

При **просмотре** панели доступны следующие параметры:

Edit dashboard	Переключиться в режим **редактирования** панели. Режим редактирования также открывается при создании новой панели и при нажатии кнопки редактирования у виджета.
	Открыть меню действий (см. описание действий ниже).

	*Общий доступ* — изменить настройки общего доступа для панели. *Создать новую* — создать новую панель. *Клонировать* — создать новую панель, скопировав свойства существующей. Сначала будет предложено ввести параметры панели. Затем новая панель откроется в режиме редактирования со всеми виджетами исходной панели. *Удалить* — удалить панель. *Создать новый отчёт* — открыть всплывающее окно с формой настройки отчёта. Недоступно, если у пользователя нет прав на управление отчётами по расписанию. *Просмотреть связанные отчёты* — открыть всплывающее окно со списком существующих отчётов, основанных на текущей панели. Недоступно, если нет связанных отчётов или у пользователя нет прав на просмотр отчётов по расписанию.
	Отображать только содержимое страницы (режим киоска). К режиму киоска также можно перейти с помощью следующих параметров URL: `/zabbix.php?action=dashboard.view&kiosk=1`. Чтобы вернуться в обычный режим: `/zabbix.php?action=dashboard.view&kiosk=0`.

Общий доступ

Панели мониторинга могут быть общедоступными или приватными.

Общедоступные панели мониторинга видны всем пользователям. Приватные панели мониторинга видны только их владельцу, а также пользователям/группам пользователей, добавленным в настройках общего доступа.

Чтобы изменить статус общего доступа панели мониторинга, при просмотре отдельной панели мониторинга нажмите пункт *Общий доступ* в меню действий:

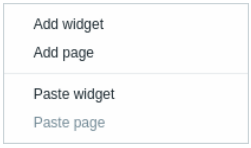
Parameter	Description
Type	Выберите тип панели мониторинга: **Private** — панель мониторинга видна только выбранным группам пользователей и пользователям. **Public** — панель мониторинга видна всем.
List of user group shares	Выберите группы пользователей, которым будет доступна панель мониторинга. Можно разрешить доступ только на чтение или на чтение и запись.
List of user shares	Выберите пользователей, которым будет доступна панель мониторинга. Можно разрешить доступ только на чтение или на чтение и запись.

См. раздел Права доступа к панелям мониторинга, чтобы узнать, как параметры общего доступа ограничивают действия, доступные пользователям.

Редактирование панели

При **редактировании** панели доступны следующие параметры:

	Изменить общие параметры панели.
	Добавить новый виджет.
	Нажатие на кнопку со стрелкой откроет меню действий (см. описание действий ниже).

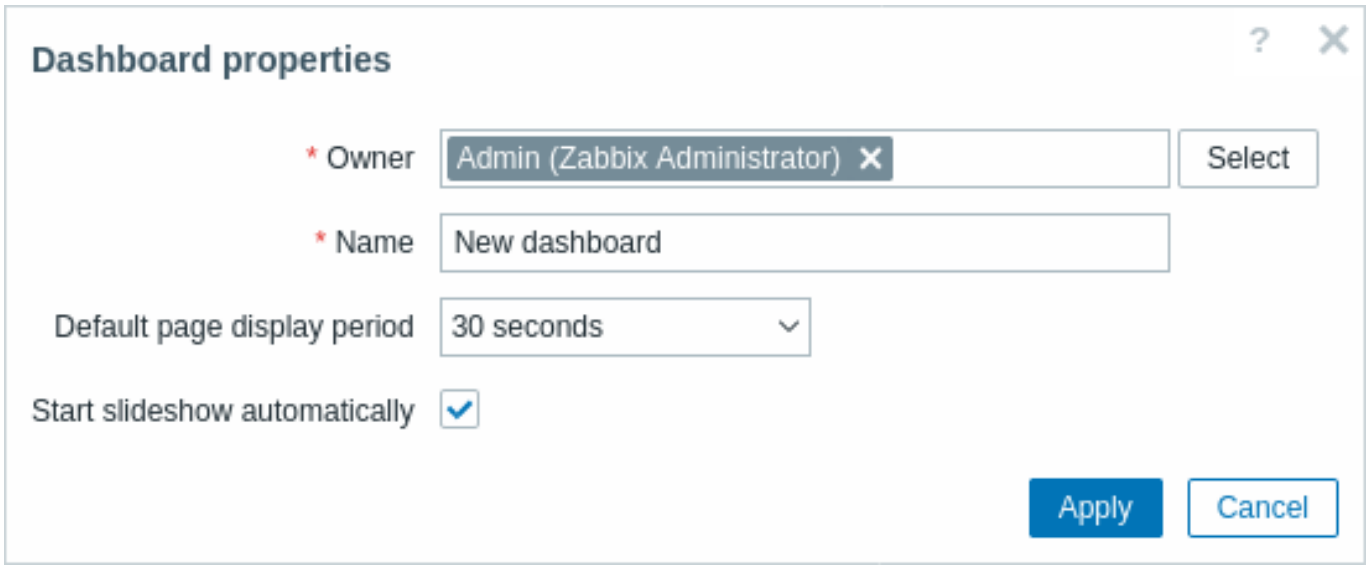
	*Добавить виджет* — добавить новый виджет. *Добавить страницу* — добавить новую страницу. *Вставить виджет* — вставить скопированный виджет. Этот параметр недоступен, если ни один виджет не был скопирован. Одновременно можно скопировать только один объект (виджет или страницу). *Вставить страницу* — вставить скопированную страницу. Этот параметр недоступен, если ни одна страница не была скопирована.
	Сохранить изменения панели.
	Отменить изменения панели.

Создание панели мониторинга

Новую панель мониторинга можно создать двумя способами:

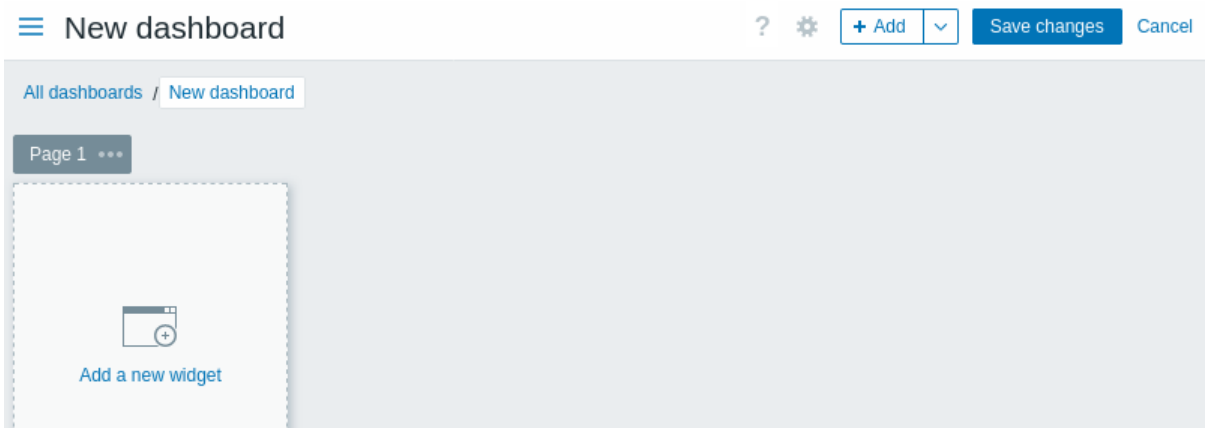
- Нажмите *Создать панель мониторинга* при просмотре всех панелей мониторинга
- Выберите *Создать новую* в меню действий при просмотре одной панели мониторинга

Сначала вам будет предложено ввести общие параметры панели мониторинга:



Параметр	Описание
Владелец	Выберите пользователя системы, который будет владельцем панели мониторинга.
Имя	Введите имя панели мониторинга.
Период отображения страницы по умолчанию	Выберите период, в течение которого страница панели мониторинга будет отображаться перед переключением на следующую страницу в слайд-шоу.
Запускать слайд-шоу автоматически	Установите этот флажок, чтобы слайд-шоу запускалось автоматически, если существует более одной страницы панели мониторинга.

После нажатия *Применить* откроется пустая панель мониторинга:



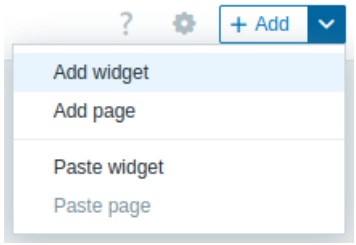
Чтобы заполнить панель мониторинга, вы можете добавить виджеты и страницы.

Нажмите кнопку *Сохранить изменения*, чтобы сохранить панель мониторинга.
Если вы нажмете *Отмена*, панель мониторинга не будет создана.

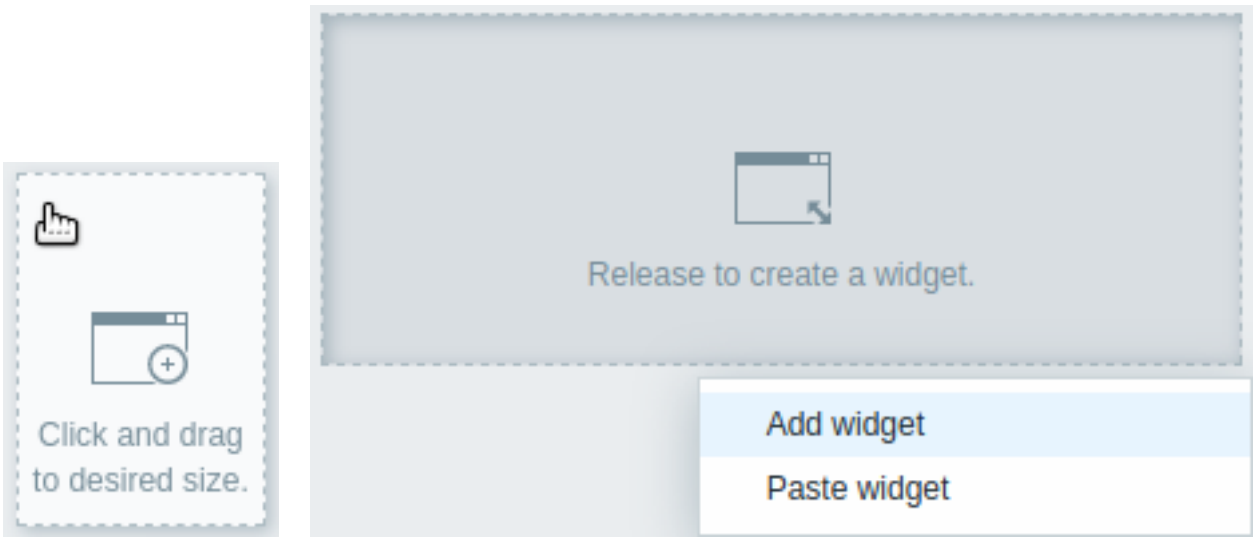
Добавление виджетов

Вы можете добавлять на панель различные виджеты (журнал действий, часы, статус обнаружения и т. д.). Виджеты можно добавлять двумя способами:

- Нажмите кнопку *Добавить* или стрелку рядом с ней, затем выберите *Добавить виджет* в меню действий. После настройки виджет будет добавлен с размером по умолчанию и размещен после всех существующих виджетов.

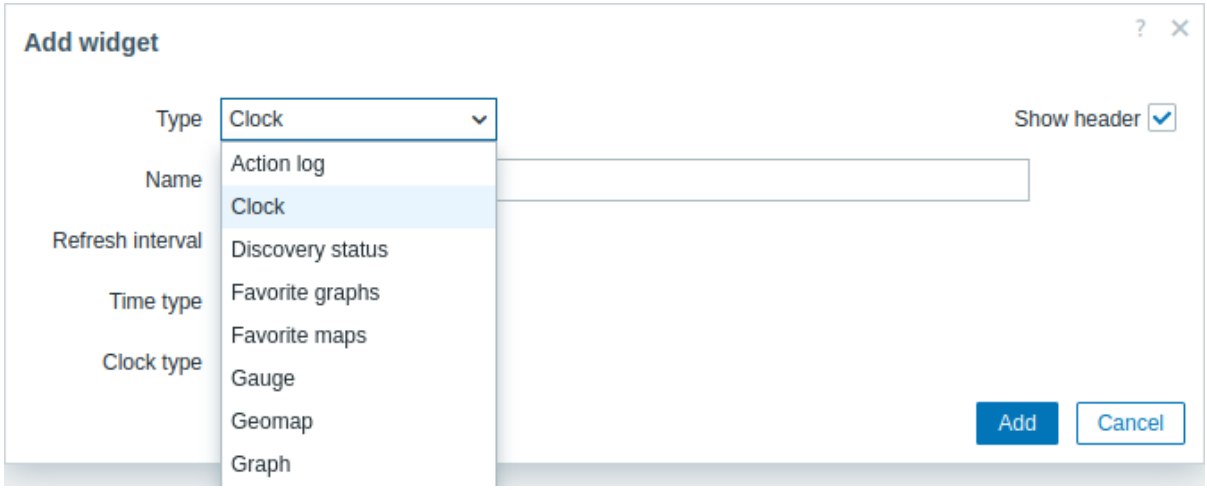


- Переместите указатель мыши на пустое место на панели. Обратите внимание, что появится заполнитель; затем нажмите на него, чтобы открыть форму настройки виджета. После настройки виджет будет добавлен с размером по умолчанию или изменен в размере так, чтобы соответствовать доступному пространству. Либо перетащите заполнитель, чтобы определить размер виджета, и отпустите его, чтобы открыть форму настройки виджета (если в буфер обмена скопирован виджет, сначала будет предложено выбрать между *Добавить виджет* и *Вставить виджет*).



В форме настройки виджета:

1. Выберите тип виджета.
2. Настройте параметры виджета.
3. Нажмите *Добавить*, чтобы добавить виджет на панель.



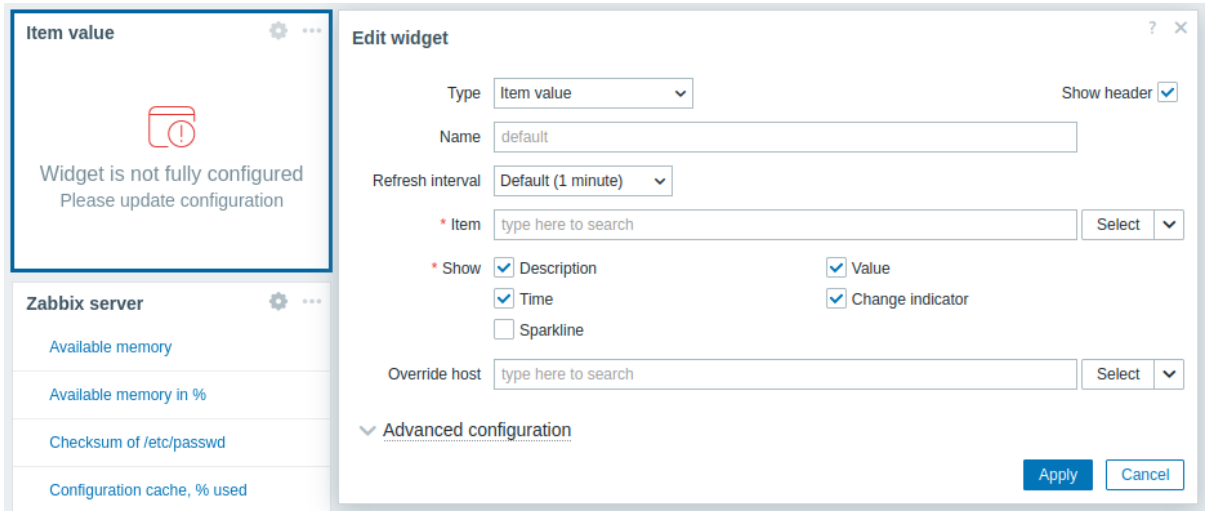
Редактирование виджетов

Размер виджетов можно изменять, а сами виджеты можно перемещать в режиме редактирования панели, перетаскивая их строку заголовка. Вы также можете перемещать форму настройки виджета, перетаскивая ее строку заголовка.

Каждый виджет содержит элементы управления в правом верхнем углу:

- - редактировать виджет
- - открыть меню виджета

Во время редактирования виджет выделяется, а изменения отображаются в режиме реального времени. Если обязательные параметры виджета не настроены, виджет переходит в состояние *Виджет настроен не полностью*.



После редактирования нажмите *Применить* в виджете, затем *Сохранить изменения* в правом верхнем углу панели, чтобы применить изменения.

Копирование/вставка виджетов

Вы можете копировать и вставлять виджеты, чтобы быстро создавать новые виджеты с той же конфигурацией. Виджеты можно копировать и вставлять в пределах одной панели или между панелями, открытыми в разных вкладках.

Чтобы скопировать виджет, используйте меню виджета. Чтобы вставить скопированный виджет (доступно в режиме редактирования панели):

- Нажмите на стрелку рядом с кнопкой *Добавить* и выберите *Вставить виджет*.
- Либо щелкните по пустой области панели и выберите *Вставить виджет*.

Вы также можете вставить скопированный виджет поверх существующего виджета, используя опцию *Вставить* в его меню виджета.

Создание слайд-шоу

Слайд-шоу будет запускаться автоматически, если панель содержит две или более страниц (см. Добавление страниц) и если выполняется одно из следующих условий:

- В свойствах панели отмечена опция *Запускать слайд-шоу автоматически*
- URL панели содержит параметр `slideshow=1`

Страницы переключаются в соответствии с интервалами, заданными в свойствах панели и отдельных страниц. Нажмите:

- *Остановить слайд-шоу* — чтобы остановить слайд-шоу
- *Запустить слайд-шоу* — чтобы запустить слайд-шоу



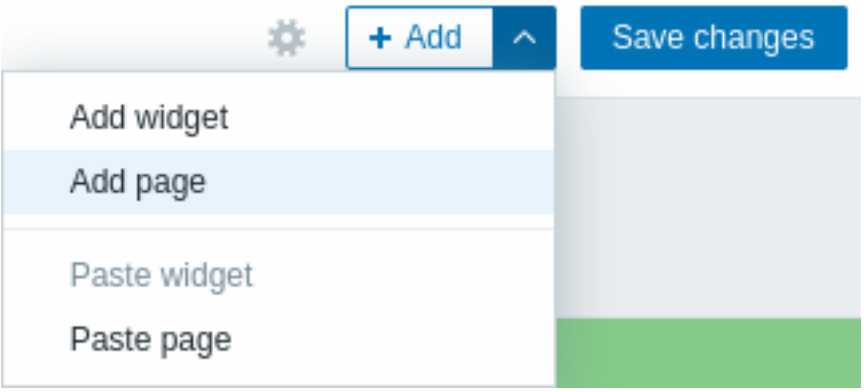
Элементы управления слайд-шоу также доступны в режиме киоска (где отображается только содержимое страницы):

- — остановить слайд-шоу
- — запустить слайд-шоу
- — вернуться на одну страницу назад
- — перейти к следующей странице

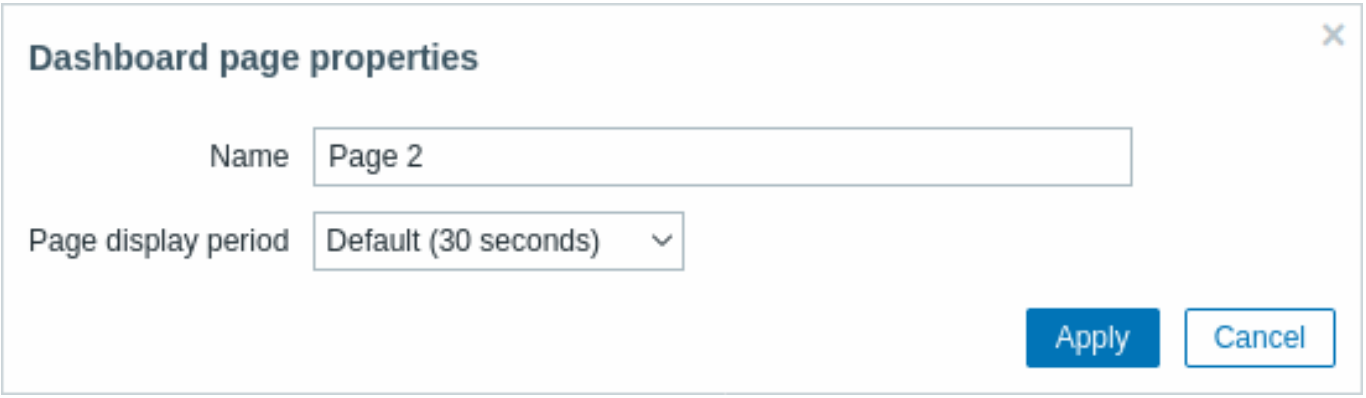
Добавление страниц

Чтобы добавить новую страницу на панель:

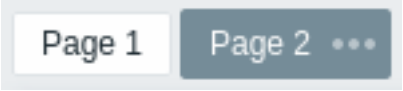
- Убедитесь, что панель находится в режиме редактирования
- Нажмите на стрелку рядом с кнопкой *Добавить* и выберите опцию *Добавить страницу*



- Заполните общие параметры страницы и нажмите *Применить*. Если оставить имя пустым, страница будет добавлена с именем `Page N`, где 'N' — порядковый номер страницы. Период отображения страницы позволяет настроить, как долго страница будет отображаться в слайд-шоу.



Будет добавлена новая страница, обозначенная новой вкладкой (*Page 2*).



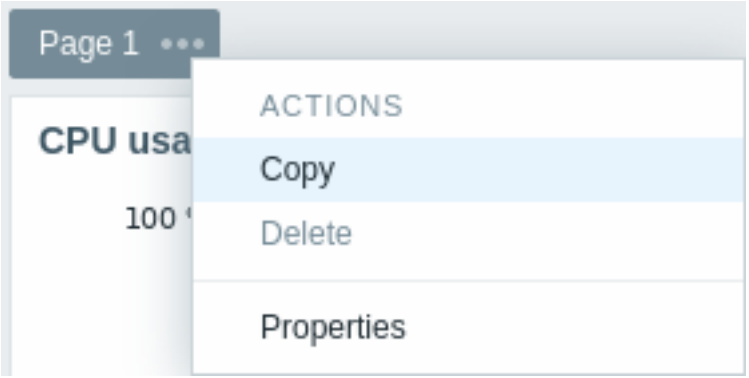
Страницы можно менять местами, перетаскивая их вкладки. При изменении порядка исходные имена страниц сохраняются. Всегда можно перейти к любой странице, щёлкнув по её вкладке.

Когда новая страница добавлена, она пуста. Вы можете добавить на неё виджеты, как описано выше.

Копирование/вставка страниц

Страницы панели могут быть скопированы и вставлены, что позволяет создать новую страницу со свойствами существующей. Их можно вставлять как из той же панели, так и из другой панели.

Чтобы вставить существующую страницу на панель, сначала скопируйте ее, используя меню страницы:

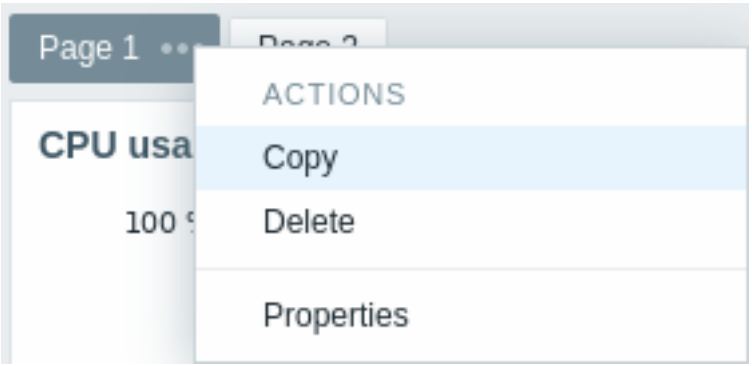


Чтобы вставить скопированную страницу:

- Убедитесь, что панель находится в режиме редактирования
- Нажмите на стрелку рядом с кнопкой *Добавить* и выберите опцию *Вставить страницу*

Меню страницы

Меню страницы можно открыть, щелкнув по трем точкам рядом с именем страницы:

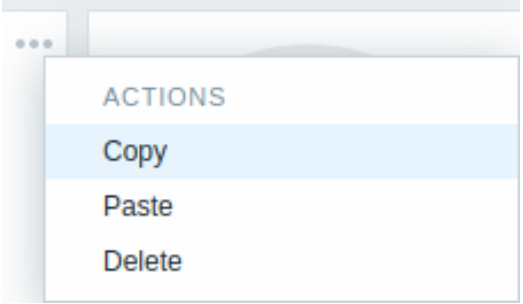


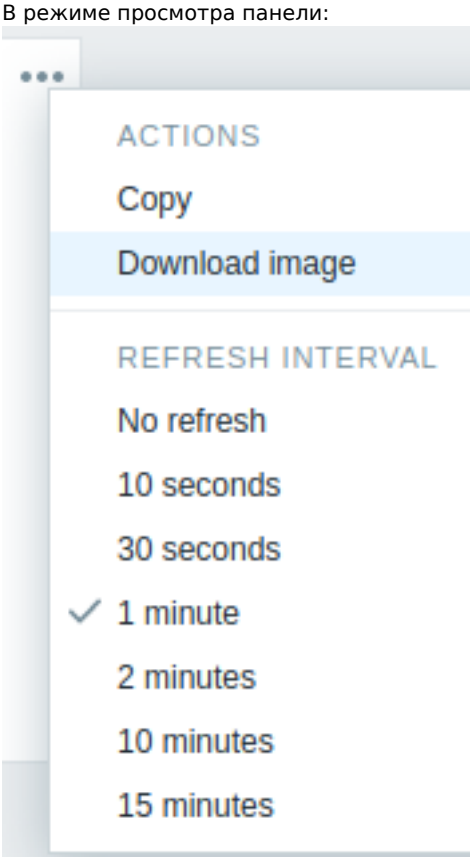
Оно содержит следующие параметры:

- *Копировать* - копировать страницу
- *Удалить* - удалить страницу (страницы можно удалять только в режиме редактирования панели)
- *Свойства* - настроить параметры страницы (имя и период отображения страницы в слайд-шоу)

Меню виджета

Меню виджета содержит различные параметры в зависимости от того, находится ли панель в режиме редактирования или просмотра:

Меню виджета	Параметры
В режиме редактирования панели: 	*Копировать* — копировать виджет. *Вставить* — вставить скопированный виджет поверх этого виджета. Этот параметр неактивен, если ни один виджет не был скопирован. *Удалить* — удалить виджет.

Меню виджета	Параметры
В режиме просмотра панели: 	*Копировать* — копировать виджет. *Скачать изображение* — скачать виджет как изображение PNG. Этот параметр доступен для виджетов, поддерживающих эту возможность (см. страницы отдельных виджетов). В зависимости от настроек вашего браузера изображение либо автоматически сохраняется в папку Downloads, либо открывается диалоговое окно сохранения, позволяющее выбрать расположение. *Интервал обновления* — выбрать частоту обновления содержимого виджета

Права доступа к панелям

Права доступа к панелям для обычных пользователей и пользователей типа 'Admin' (при условии, что для назначенной им роли пользователя включен доступ к панелям и опция *Создавать и редактировать панели*) ограничены следующим образом.

- Они могут просматривать и клонировать общедоступную панель, даже если в настройках общего доступа панели не добавлены пользователи или группы пользователей.
- Они могут просматривать и клонировать частную панель, если им предоставлены как минимум права *Чтение* с помощью настроек общего доступа.
- Они могут редактировать и удалять панель, только если им предоставлены права *Чтение-запись* с помощью настроек общего доступа.
- Они не могут изменить владельца панели.

1 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.

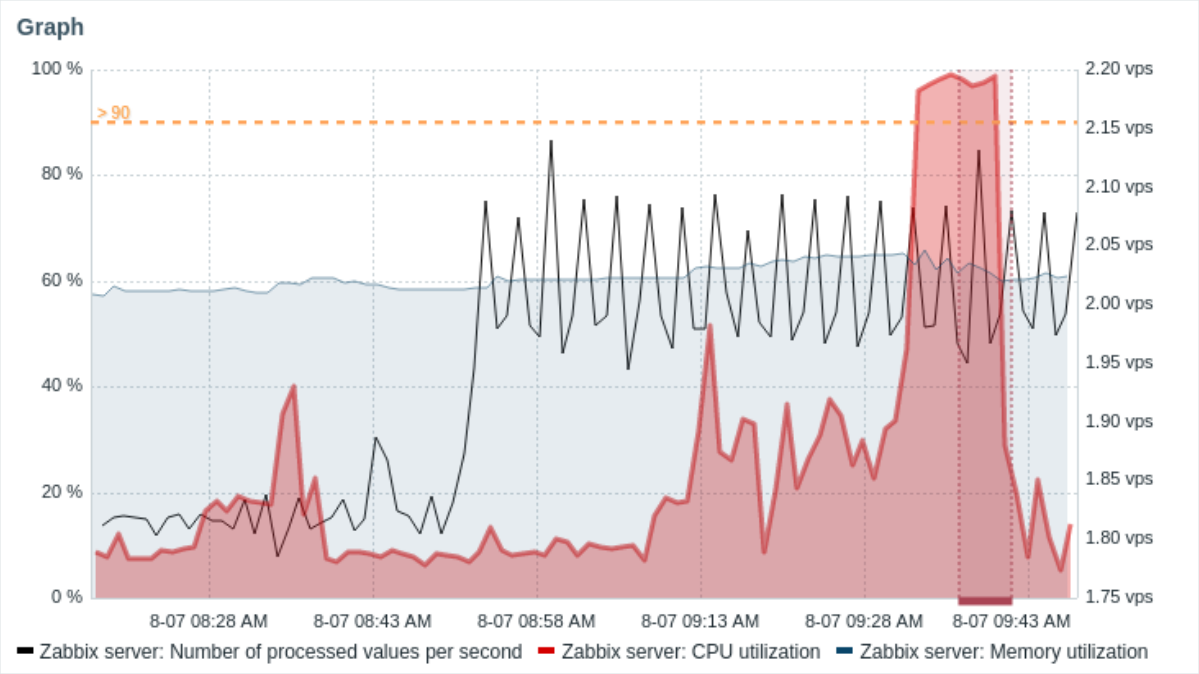
Подробные сведения о настройке виджетов см. в разделе Параметры виджетов панели мониторинга.

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.

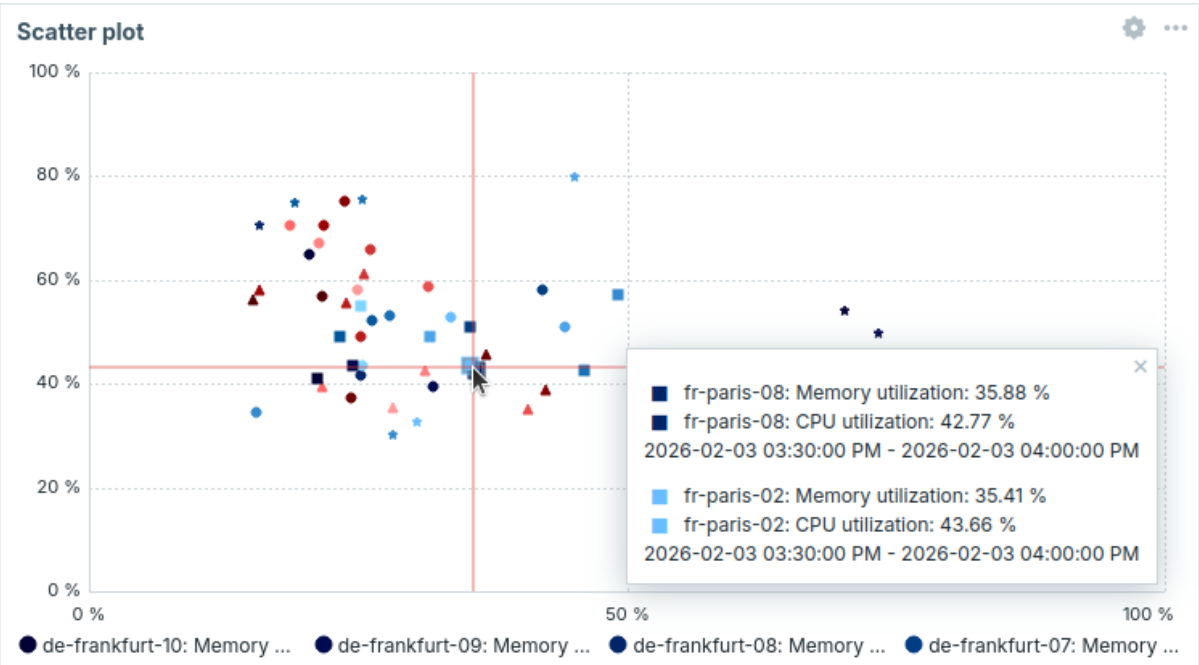


См. также:

- 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.

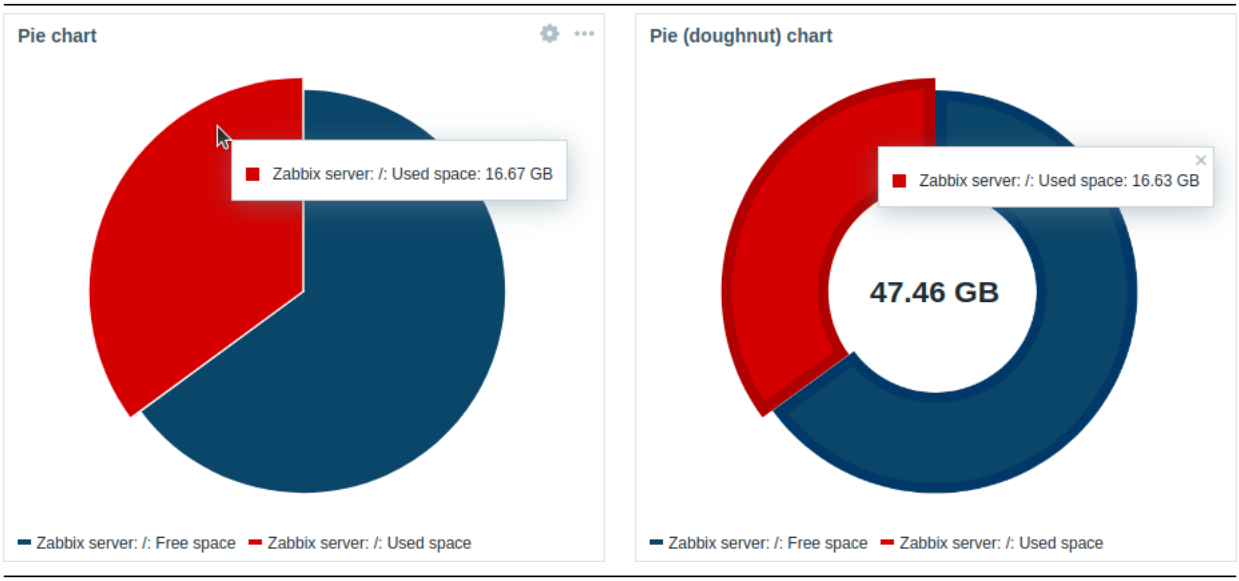
Диаграмма рассеяния

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

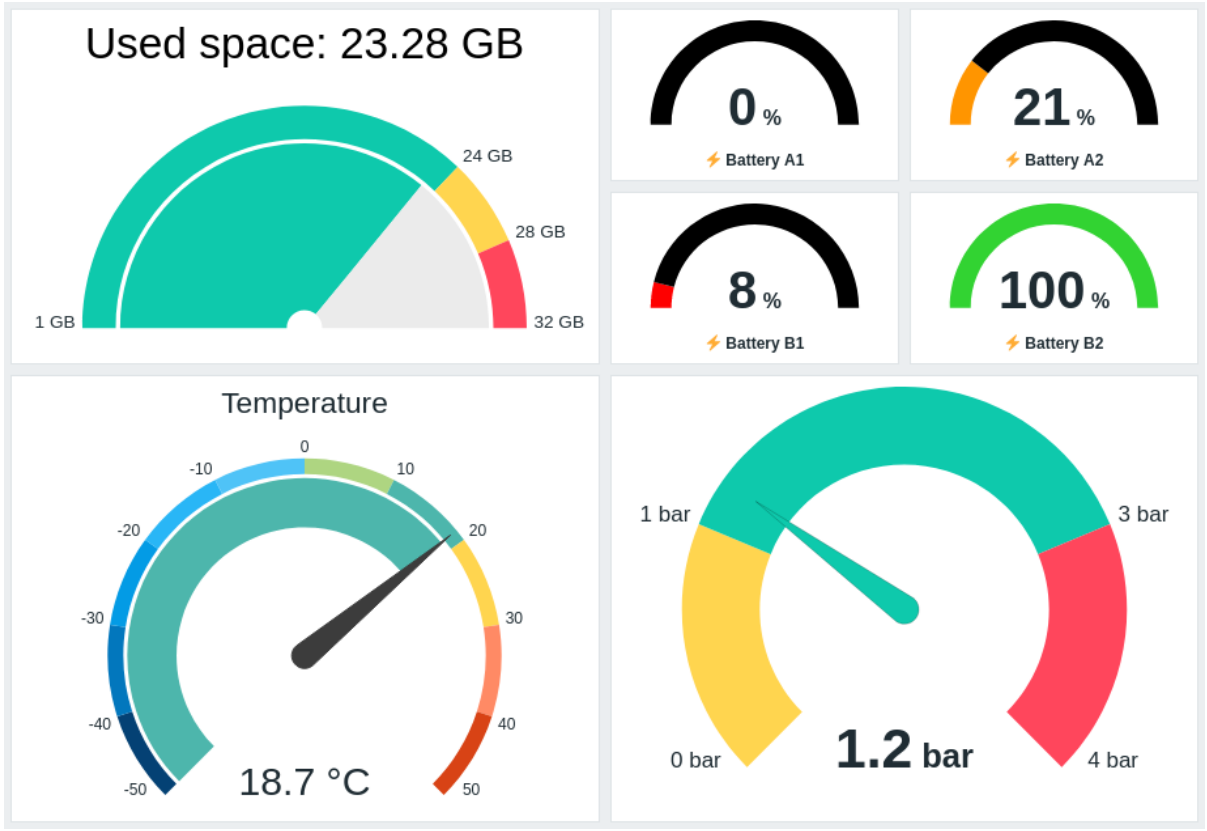
Виджет Круговая диаграмма отображает числовые данные элементов данных в виде векторной круговой или кольцевой диаграммы. Этот виджет помогает наглядно представить, как элементы данных или узлы сети вносят вклад в общий набор данных.



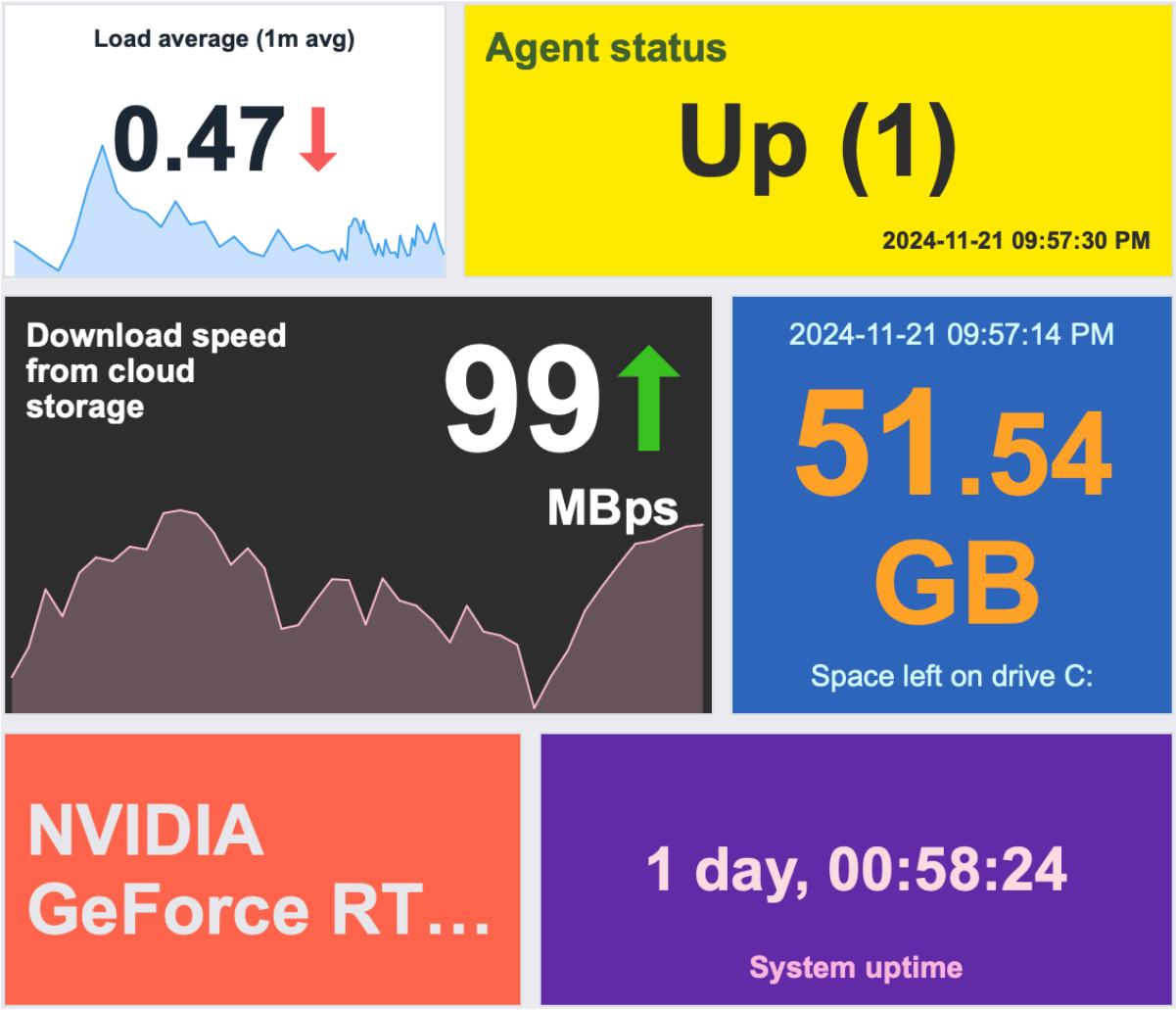
Gauge/Значение элемента данных

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:

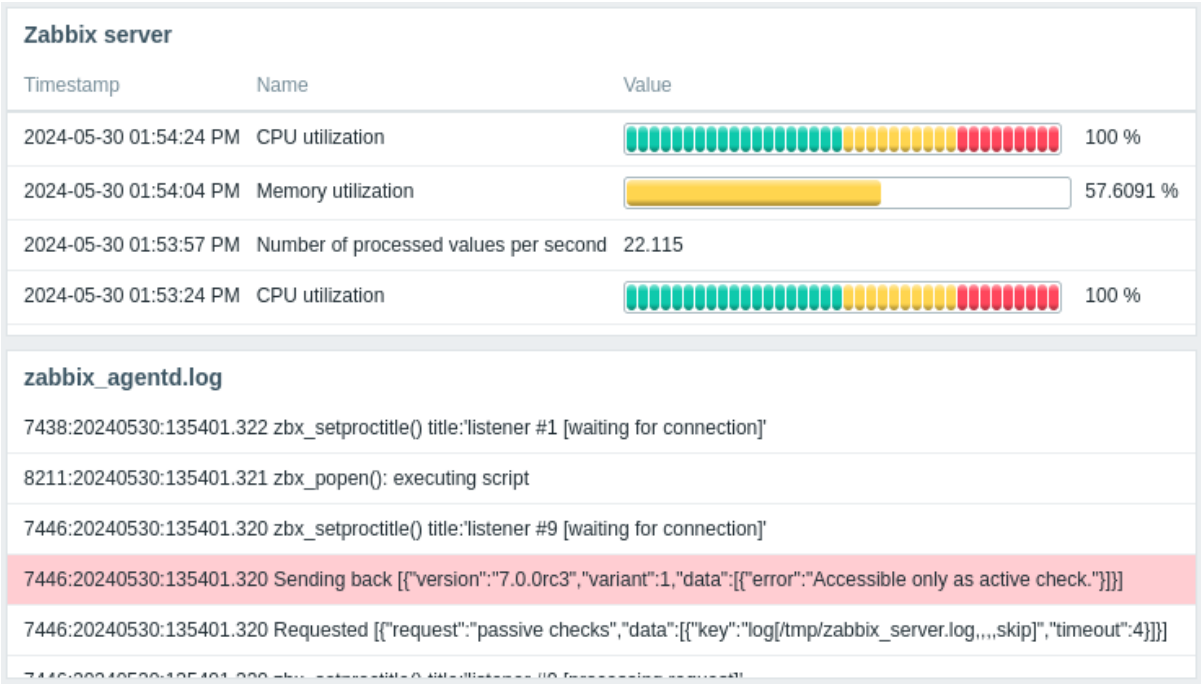


Виджет «Значение элемента данных»:



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 %

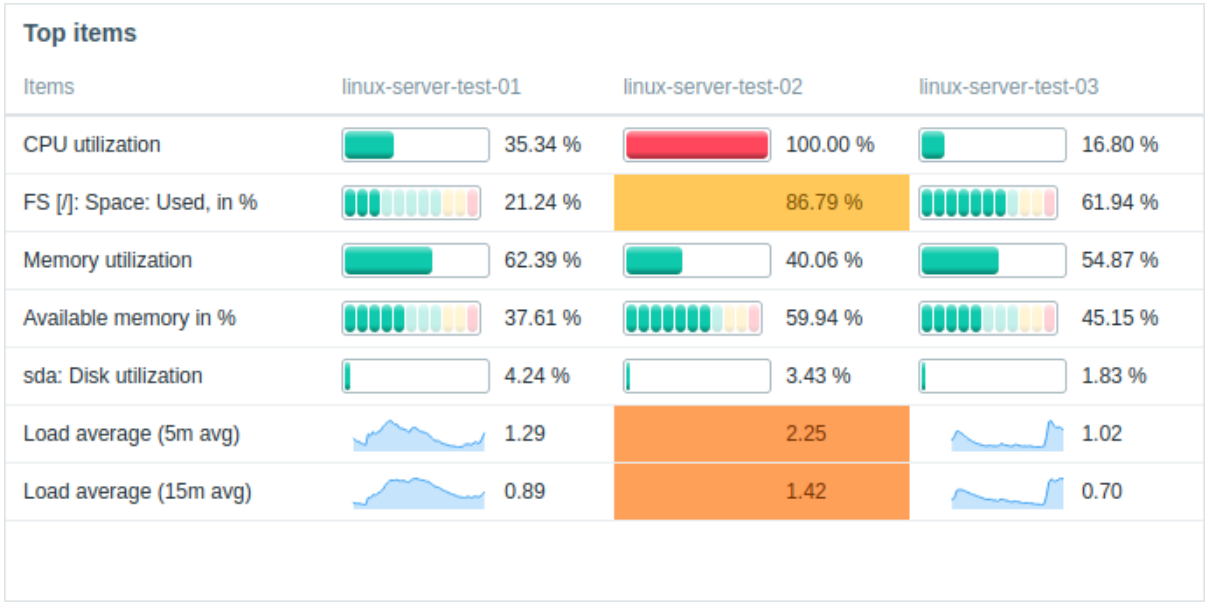
zabbix_agentd.log

```
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.



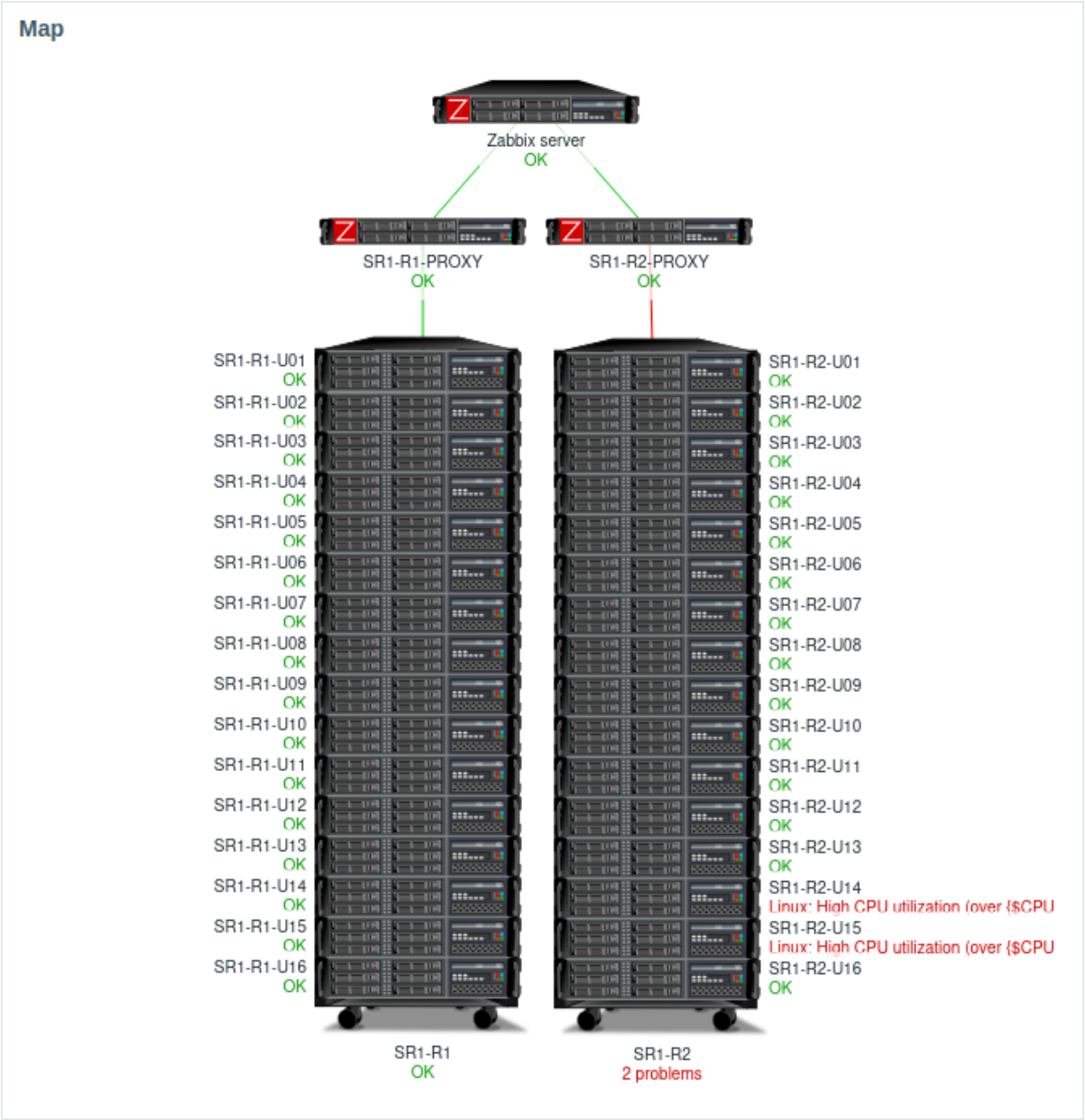


Карты

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.

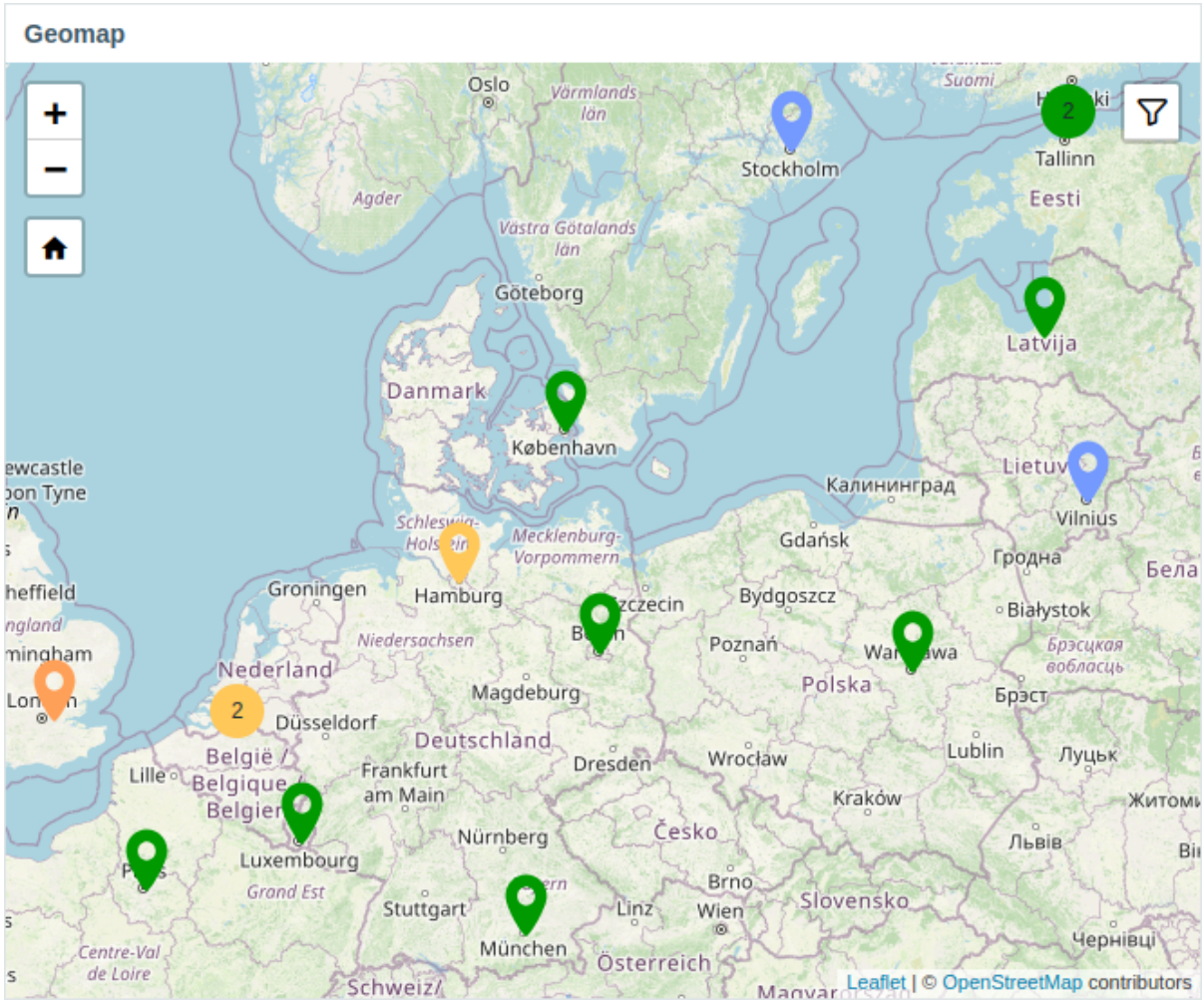


См. также:

- 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.

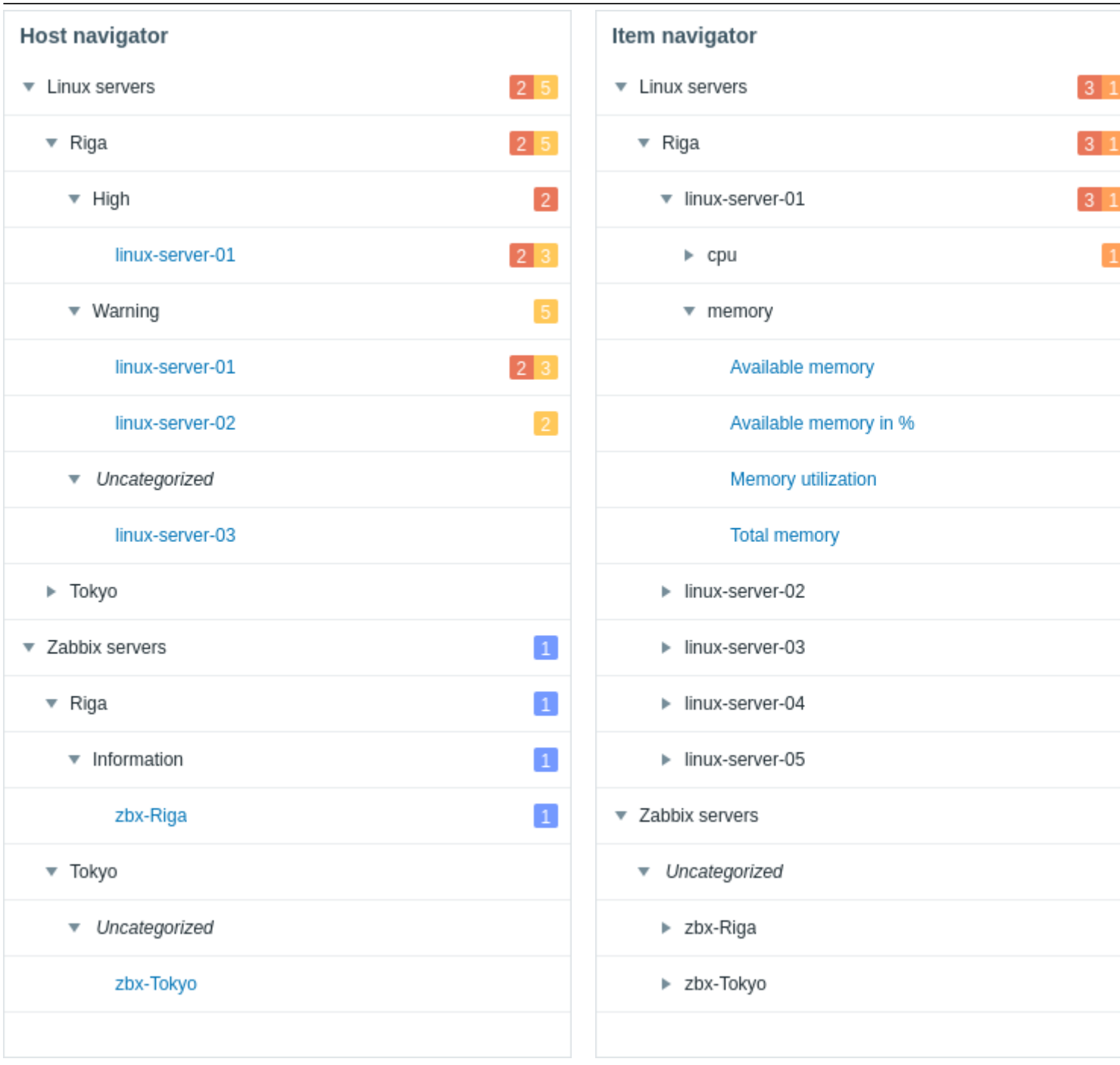


Сведения об узле сети/элементе данных

Эти виджеты предоставляют списки и подробные представления узлов сети или элементов данных.

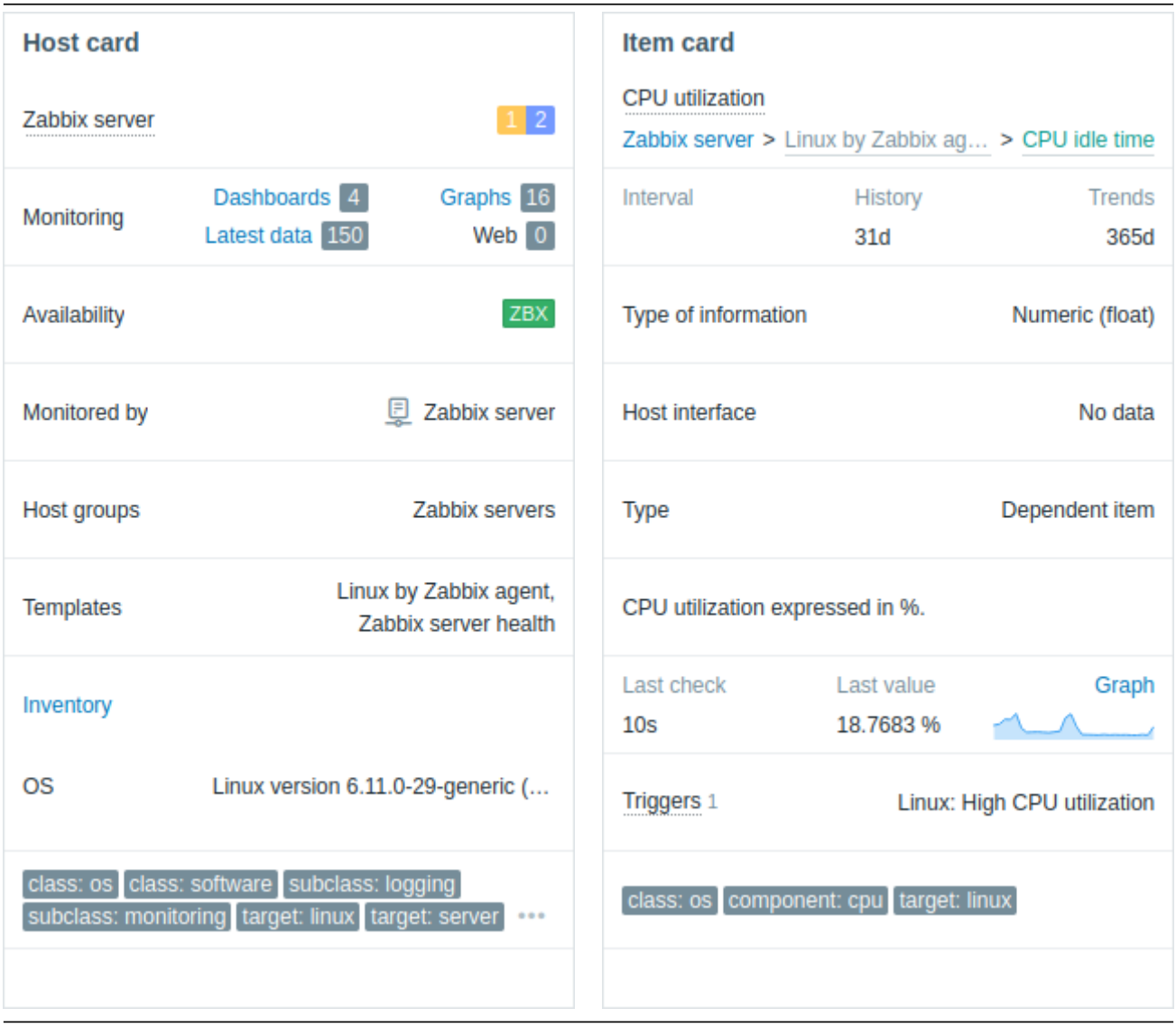
Навигатор узлов сети/элементов данных

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

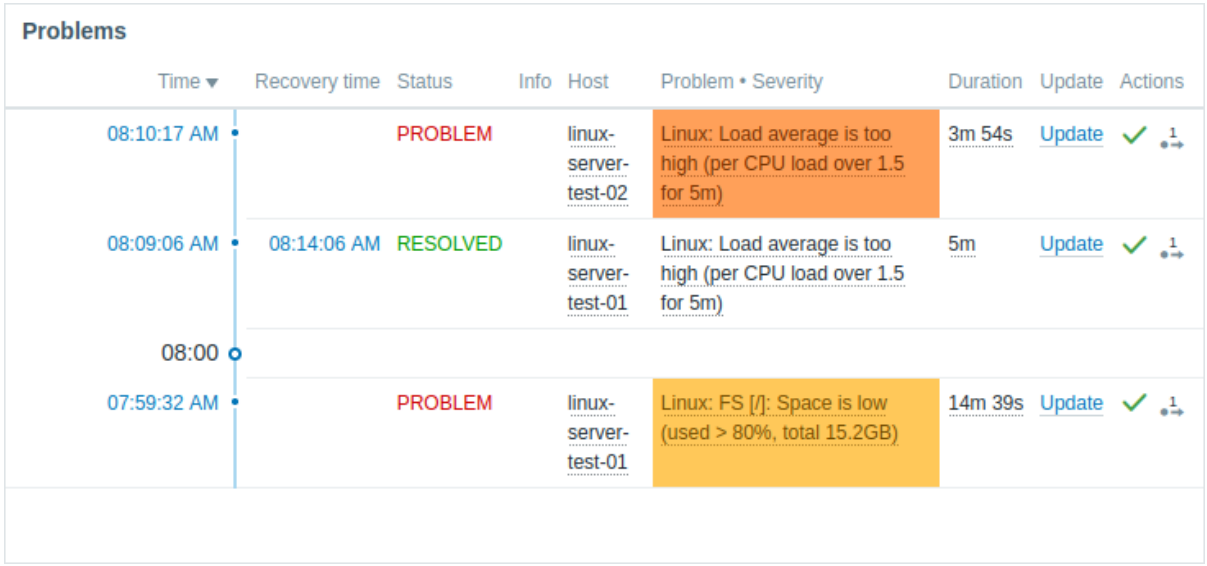
Виджет Доступность узлов сети отображает доступность узлов сети в выбранных группах узлов сети, позволяя отслеживать количество узлов сети, которые доступны или недоступны.

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

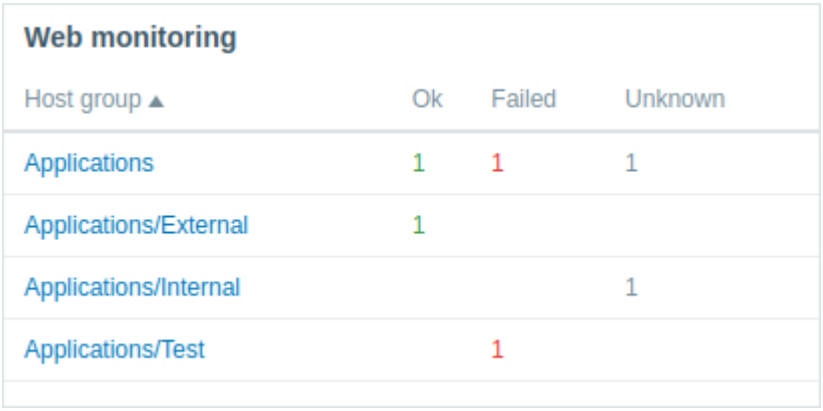
Виджет Отчет SLA отображает отчеты SLA, помогая вам контролировать производительность сервисов относительно заданных целевых показателей.

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

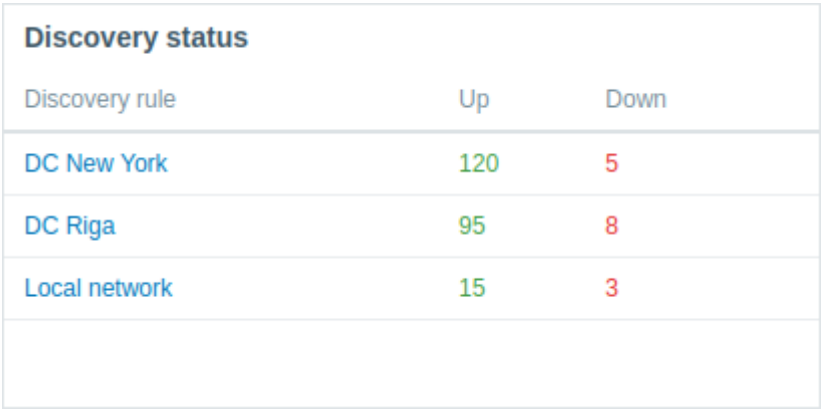
Веб-мониторинг

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

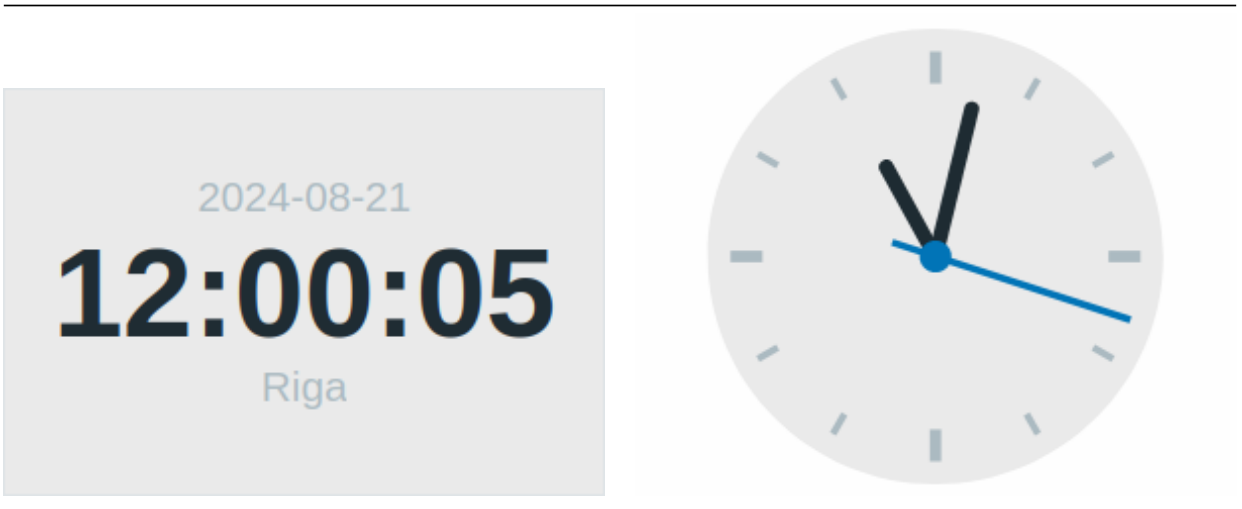
Виджет Системная информация отображает сводку по ключевым данным сервера Zabbix и системы или сведения об узлах высокой доступности.

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	

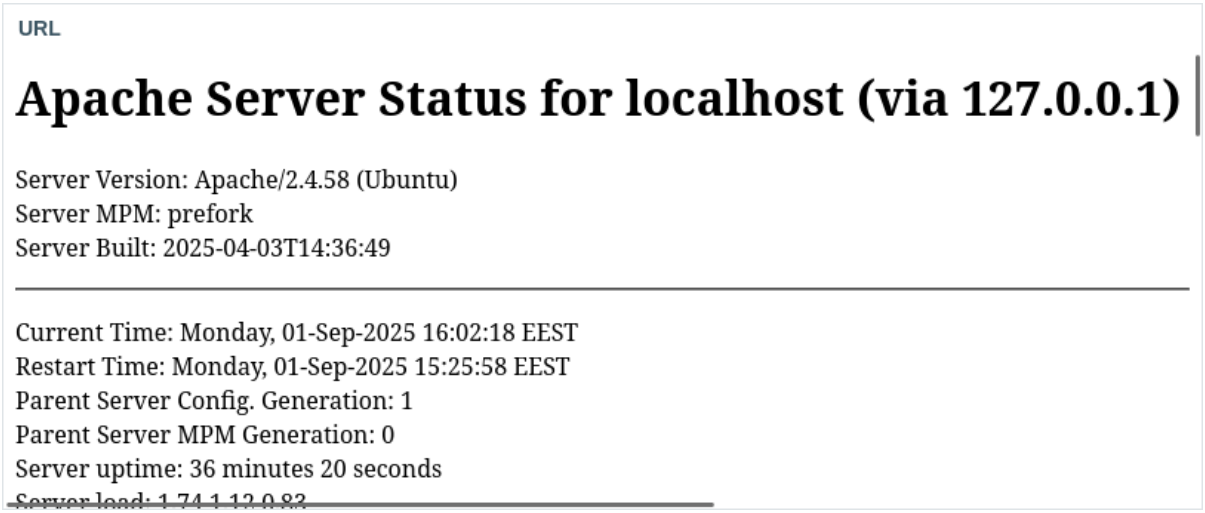
Часы

Виджет Clock отображает локальное время, время сервера или узла сети в аналоговом или цифровом формате.



URL

Виджет URL отображает содержимое, полученное по URL.



2 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:

Name	Введите имя виджета.
Refresh interval	Настройте интервал обновления по умолчанию. Интервалы обновления по умолчанию для виджетов варьируются от *No refresh* до *15 minutes* в зависимости от типа виджета. Например: - *No refresh* для виджета *URL*; - *1 minute* для виджета *Action log*; - *15 minutes* для виджета *Clock*. Интервалы обновления можно задать в качестве значения по умолчанию для всех пользователей. Переключите панель на режим редактирования, нажмите кнопку редактирования виджета и выберите нужный интервал обновления из выпадающего списка. Каждый пользователь также может задать собственный интервал обновления виджета. В режиме просмотра панели нажмите кнопку с тремя точками на виджете и выберите нужный интервал обновления из выпадающего списка. Обратите внимание, что индивидуальный интервал обновления пользователя имеет приоритет над настройкой виджета и сохраняется даже при изменении настройки виджета.
Show header	Установите флажок, чтобы заголовок виджета отображался постоянно. Если флажок снят, заголовок скрыт для экономии места и становится видимым только при наведении указателя мыши на виджет (как в режиме просмотра, так и в режиме редактирования). Заголовок также частично отображается 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 самой панелью. После этого виджеты будут автоматически обновляться на основе выбранных значений или широковещательных сообщений из других частей панели.

Совместимость виджетов

Некоторые виджеты могут передавать данные конфигурации другим виджетам, некоторые могут получать данные, а некоторые могут делать и то, и другое. Например:

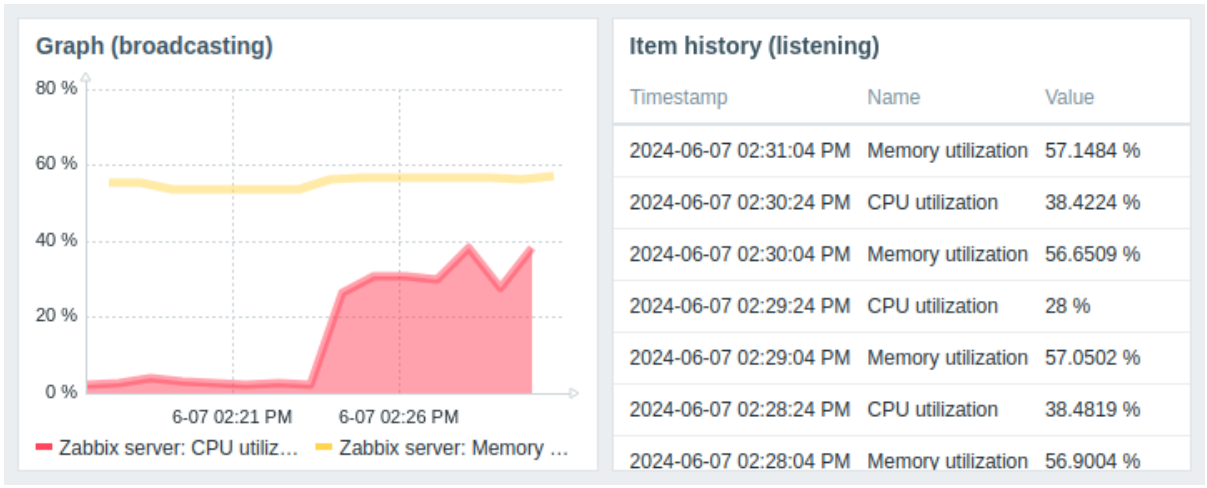
- Виджет *Журнал действий* может получать данные только о периоде времени из виджетов *График*, *График (классический)* и *Прототип графика*.
- Виджет *Geomap* может передавать данные об узлах сети виджетам, которые могут их получать (*Honeycomb*, *Top items* и т. д.), а также может получать данные о группах узлов сети и узлах сети от виджетов, которые могут их передавать (*Honeycomb*, *Problem hosts* и т. д.).
- Виджет *Часы* не может ни передавать, ни получать какие-либо данные.

В следующей таблице приведены возможности передачи и получения данных для каждого виджета.

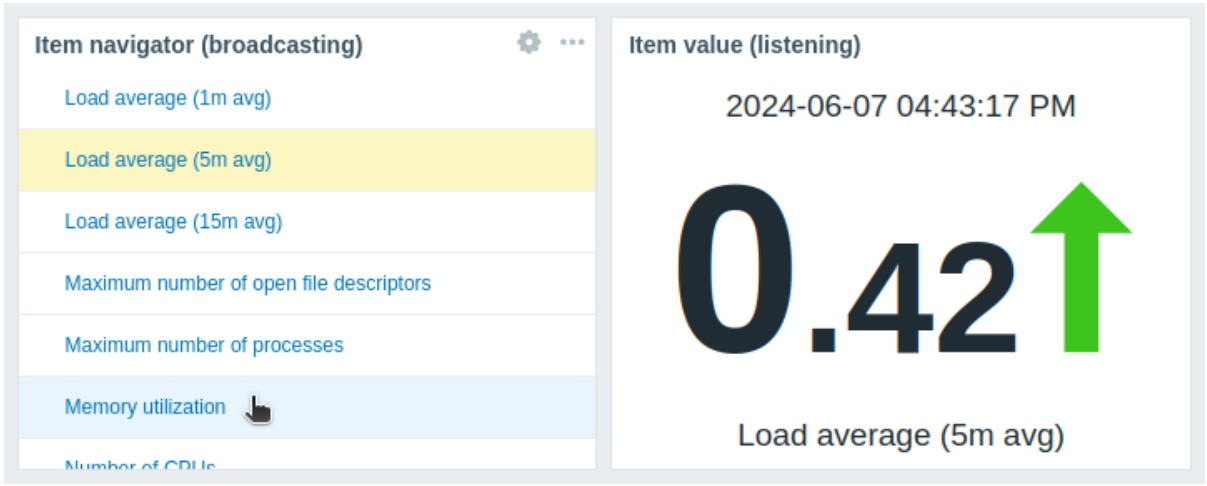
Виджет	Передает	Получает
Журнал действий	-	Период времени
Часы	-	-
Состояние обнаружения	-	-
Избранные графики	-	-
Избранные карты	-	-
Индикатор	-	Узлы сети, Элементы данных
Geomap	Узлы сети	Группы узлов сети, Узлы сети
График	Период времени, Элементы данных	Узлы сети, Элементы данных, Период времени
График (классический)	Период времени	Узлы сети, Элементы данных, Графики, Период времени
Прототип графика	Период времени	Узлы сети, Период времени
Honeycomb	Узлы сети, Элементы данных	Группы узлов сети, Узлы сети
Доступность узла сети	-	Группы узлов сети
Карточка узла сети	-	Узлы сети
Навигатор узлов сети	Узлы сети	Группы узлов сети
Карточка элемента данных	-	Узлы сети, Элементы данных, Период времени
История элемента данных	Элементы данных	Узлы сети, Период времени
Навигатор элементов данных	Элементы данных	Группы узлов сети, Узлы сети
Значение элемента данных	-	Узлы сети, Элементы данных, Период времени
Карта	Группы узлов сети, Узлы сети	Карты
Дерево навигации по картам	Карты	-
Круговая диаграмма	-	Элементы данных, Период времени
Узлы сети с проблемами	Группы узлов сети	Группы узлов сети, Узлы сети
Проблемы	События	Группы узлов сети, Узлы сети
Проблемы по серьезности	Группы узлов сети	Группы узлов сети, Узлы сети
Точечная диаграмма	Период времени, Элементы данных	Узлы сети, Элементы данных, Период времени
Отчет SLA	-	-
Системная информация	-	-
Топ узлов сети	Узлы сети	Группы узлов сети, Узлы сети
Топ элементов данных	-	Группы узлов сети, Узлы сети
Топ триггеров	-	Период времени
Обзор триггеров	-	Группы узлов сети, Узлы сети
URL	-	Узлы сети
Веб-мониторинг	Группы узлов сети	Группы узлов сети, Узлы сети

Виджеты различаются по тому, как они **транслируют** данные другим виджетам.

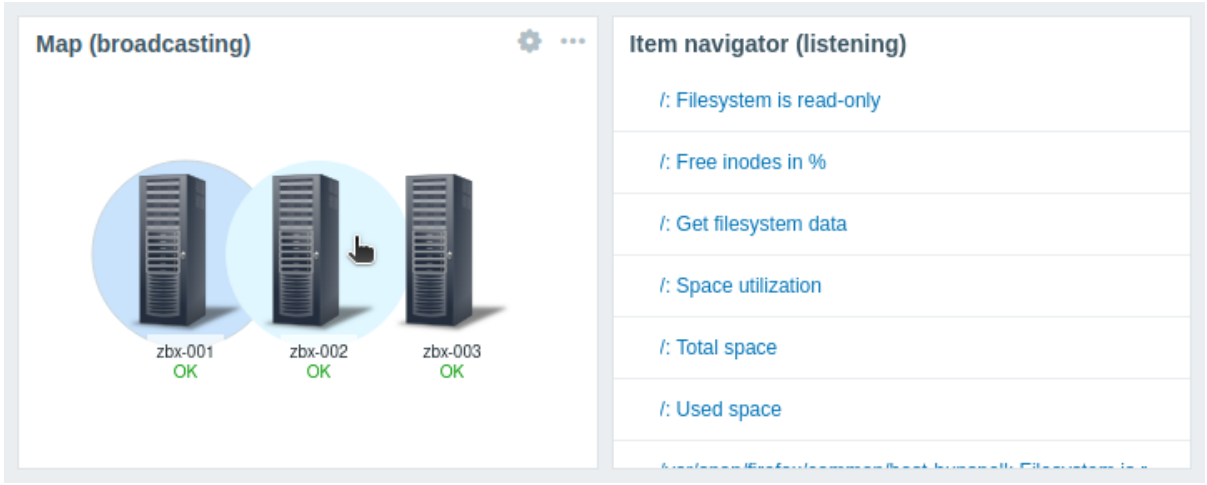
Виджеты, поддерживающие трансляцию, начинают делать это автоматически после создания. Например, виджет *Graph* сразу транслирует данные о периоде времени виджетам, которые их принимают.



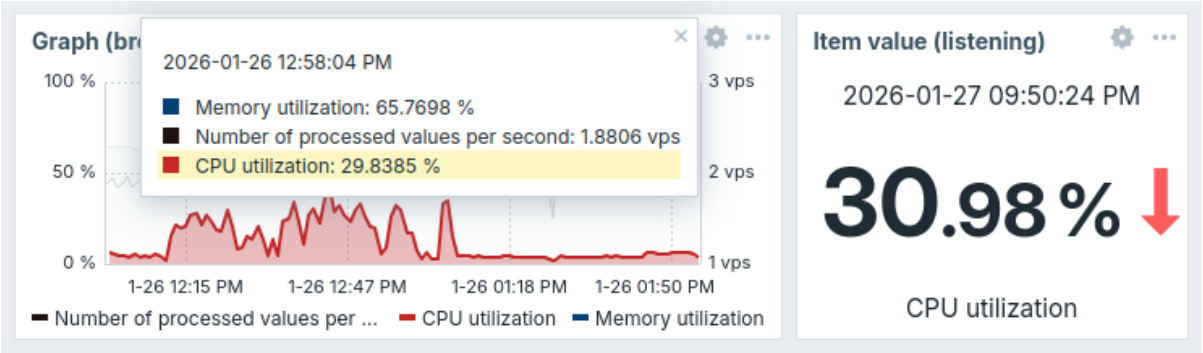
Виджеты, поддерживающие выбор сущностей, автоматически транслируют данные для первой доступной сущности. Например, виджет *Item navigator* транслирует данные для первого элемента данных в своем списке элементов данных. При выборе другого элемента данных он транслирует данные для этого элемента данных. При наведении указателя мыши элемент данных подсвечивается светло-синим цветом; при выборе — желтым.



Виджет *Map* ведет себя аналогично, транслируя данные для первого элемента, ближайшего к левому верхнему углу видимой области виджета (для виджета *Geomap* — ближайшего к центру). При выборе другого элемента он транслирует данные для этого элемента. При наведении указателя мыши элемент подсвечивается светло-синим цветом; при выборе — темно-синим.

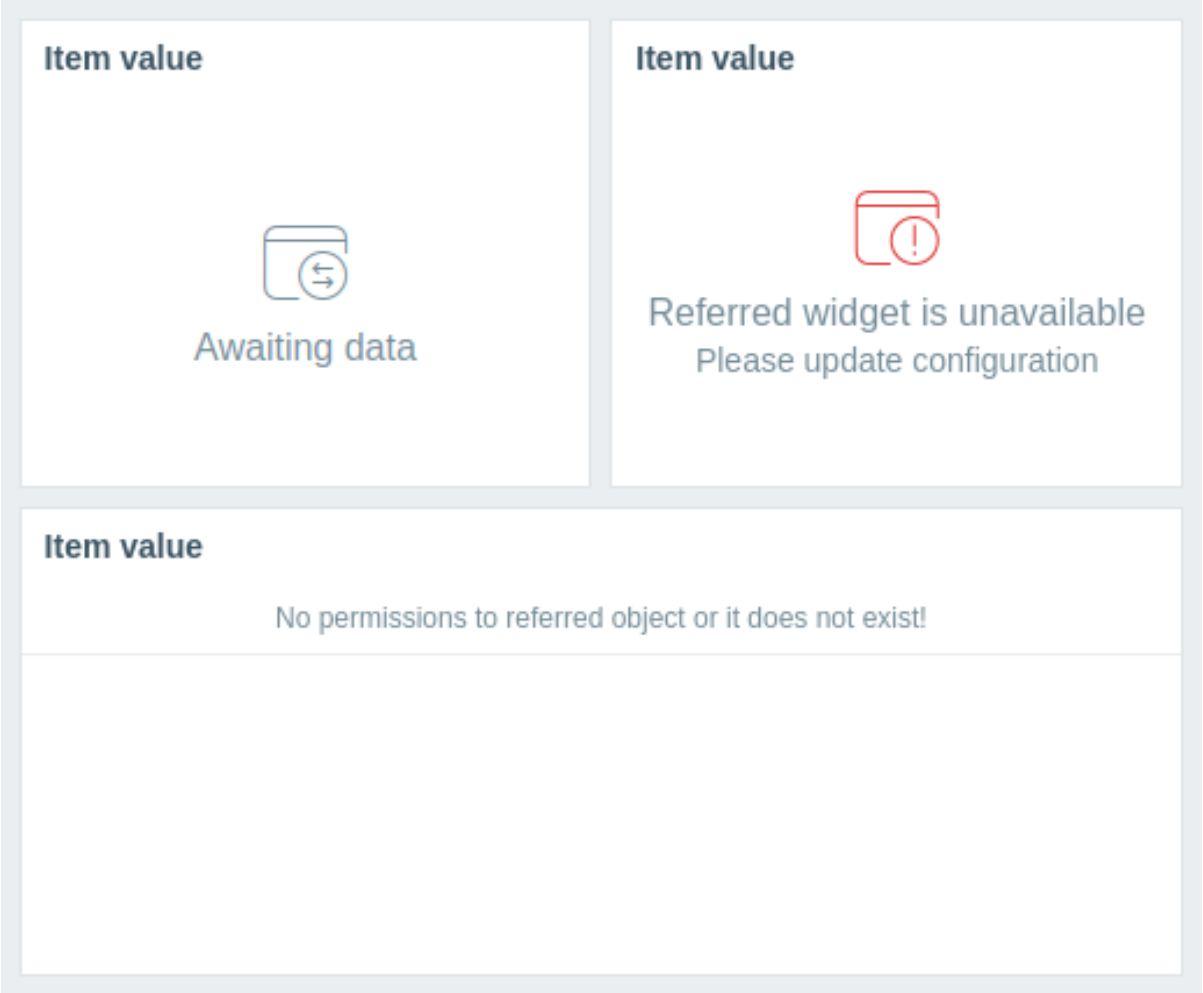


Такие виджеты, как *Graph* и *Scatter plot*, могут транслировать данные элемента данных, когда элемент данных выбран во всплывающих подсказках:



Виджеты также имеют определенное поведение при **получении** данных от других виджетов:

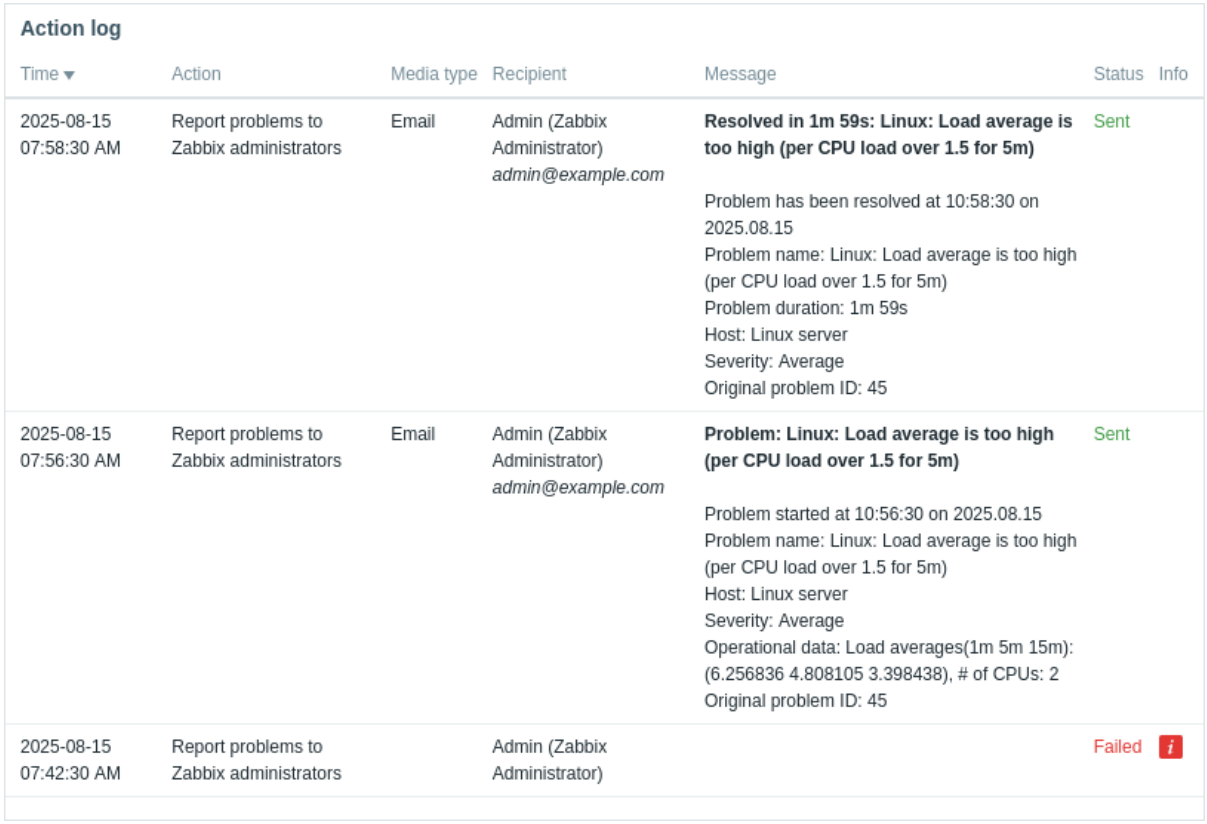
- Если виджет-источник данных не транслирует данные, принимающий виджет переходит в состояние *Awaiting data*.
- Если виджет-источник данных был удален, заменен несовместимым виджетом или перемещен на другую страницу панели, принимающий виджет переходит в состояние *Referred widget is unavailable*.
- Если указанный узел сети в источнике данных (виджете или панели) не содержит сущность, настроенную в принимающем виджете (элемент данных, график, карта 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!"*



1 Журнал действий

Обзор

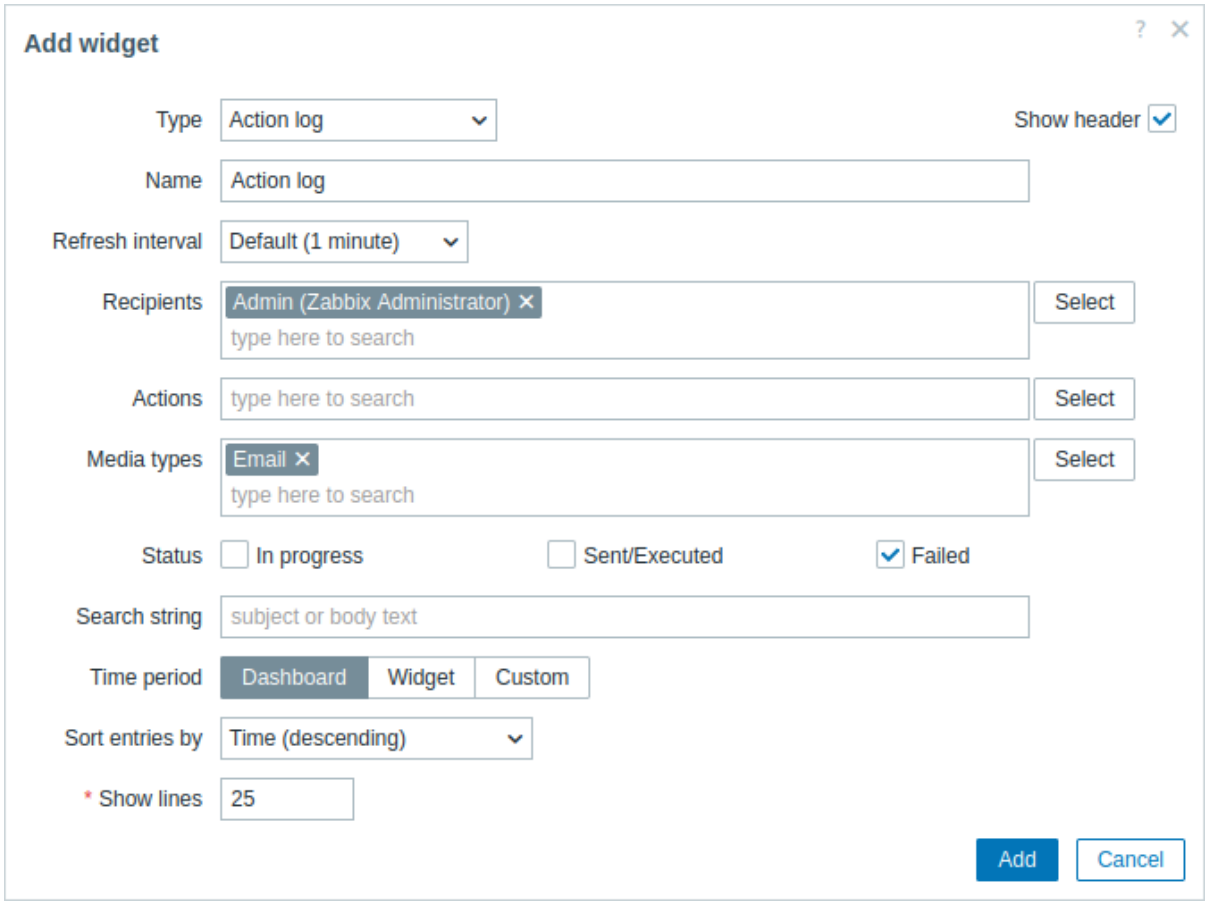
Виджет *Журнал действий* отображает сведения об операциях (уведомлениях, удаленных командах), выполненных в рамках действия. Подробный журнал помогает при аудите, устранении неполадок и мониторинге выполнения каждого действия.



Он отображает те же данные, что и *Отчеты* > *Журнал действий*, и может показывать до 1000 записей.

Настройка

Для настройки выберите *Журнал действий* в качестве типа:



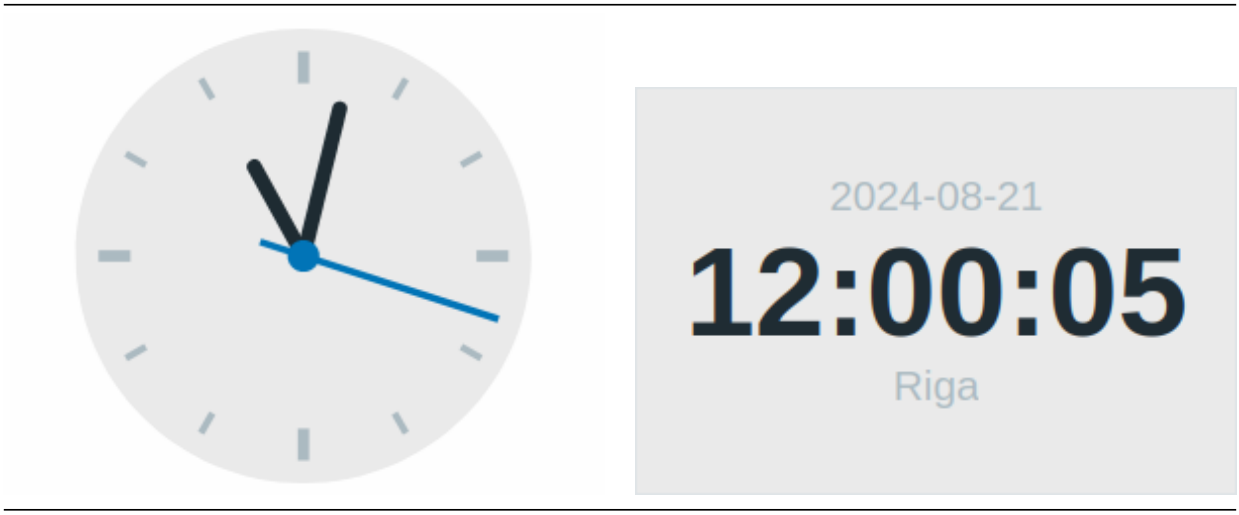
В дополнение к параметрам, которые являются общими для всех виджетов, вы можете задать следующие специальные параметры:

Получатели	Фильтрация записей по получателям. Это поле поддерживает автодополнение, поэтому при вводе имени получателя будет предложен выпадающий список подходящих получателей. Если получатели не выбраны, будут отображаться сведения об операциях действий для всех получателей.
Действия	Фильтрация записей по действиям. Это поле поддерживает автодополнение, поэтому при вводе имени действия будет предложен выпадающий список подходящих действий. Если действия не выбраны, будут отображаться сведения об операциях действий для всех действий.
Способы оповещения	Фильтрация записей по способам оповещения. Это поле поддерживает автодополнение, поэтому при вводе имени способа оповещения будет предложен выпадающий список подходящих способов оповещения. Если способы оповещения не выбраны, будут отображаться сведения об операциях действий для всех способов оповещения.
Статус	Установите флажок, чтобы фильтровать записи по соответствующему статусу: **В процессе** — отображаются операции действий, которые находятся в процессе выполнения; **Отправлено/Выполнено** — отображаются операции действий, для которых уведомление было отправлено или которые были выполнены; **Не удалось** — отображаются операции действий, завершившиеся ошибкой.
Строка поиска	Фильтрация записей по содержимому сообщения/удалённой команды. Если вы введёте здесь строку, будут отображаться только те операции действий, чьё сообщение/удалённая команда содержит введённую строку. Макросы не раскрываются.
Период времени	Фильтрация записей по периоду времени. 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 задан, в правом верхнем углу виджета будет отображаться значок часов, указывающий установленное время при наведении курсора. Обратите внимание, что независимо от настройки *Период времени* у виджета, совместимые виджеты всё 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".
Сортировать записи по	Сортировка записей по: **Время** (по убыванию или по возрастанию); **Тип** (по убыванию или по возрастанию); **Статус** (по убыванию или по возрастанию); **Получатель** (по убыванию или по возрастанию).
Показывать строк	Укажите, сколько строк журнала действий будет отображаться в виджете.

2 Часы

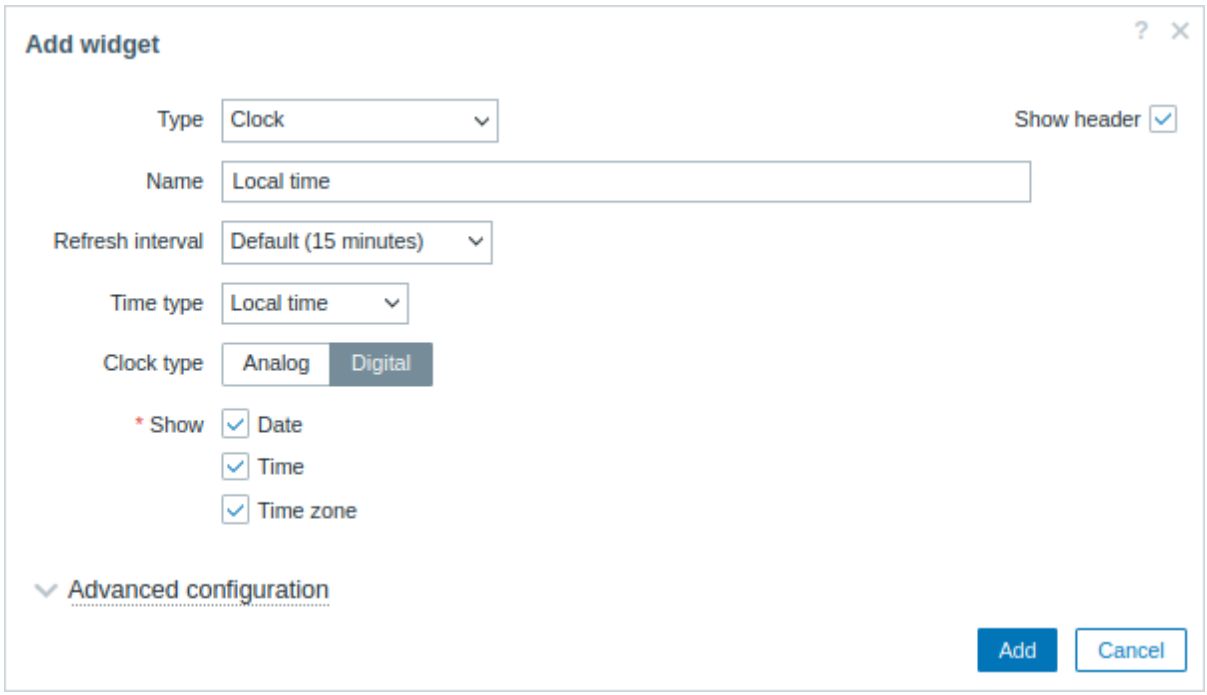
Обзор

Виджет *Clock* отображает локальное время, время сервера или узла сети в аналоговом или цифровом формате.



Настройка

Для настройки выберите *Clock* в качестве типа:



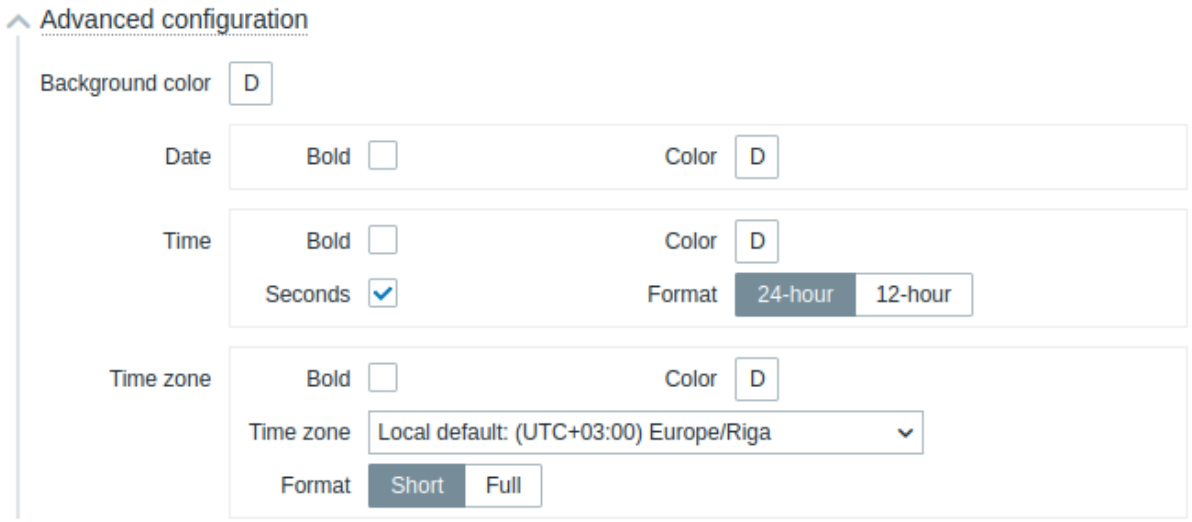
В дополнение к параметрам, которые являются общими для всех виджетов, вы можете задать следующие специальные параметры:

Тип времени	Выберите источник времени: **Локальное** — системное время устройства, на котором запущен браузер; **Сервер** — часовой пояс, заданный глобально или для пользователя Zabbix; **Узел сети** — время, полученное из элемента данных узла сети.
Элемент данных	Выберите элемент данных для времени узла сети (используйте элемент данных system.localtime[local]). Этот параметр доступен, если для *Тип времени* установлено значение «Время узла сети».
Тип часов	Выберите тип часов (аналоговые или цифровые).
Показывать	Выберите одну или несколько единиц информации (дата, время, часовой пояс) для отображения в цифровых часах. Размер единиц автоматически подстраивается под отображаемые единицы, размер виджета и масштаб отображения. Этот параметр доступен, если для *Тип часов* установлено значение «Цифровые».
Расширенная настройка	Нажмите на метку *Расширенная настройка*, чтобы отобразить параметры расширенной настройки для цифровых часов. Этот раздел доступен, если для *Тип часов* установлено значение «Цифровые».

Расширенная настройка

Параметры расширенной настройки доступны в сворачиваемом разделе *Расширенная настройка*, и только для тех элементов данных, которые выбраны в поле *Показать* (см. выше).

Кроме того, расширенная настройка позволяет изменить цвет фона для всего виджета.

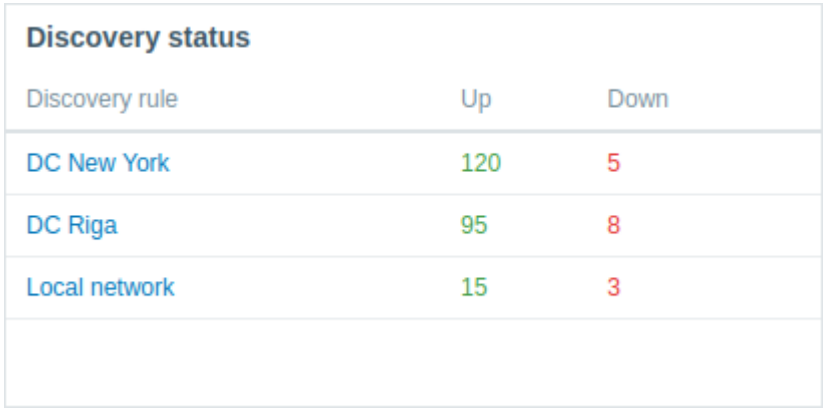


Цвет фона	Выберите цвет фона в палитре цветов. D обозначает цвет по умолчанию (зависит от темы веб-интерфейса). Чтобы вернуть значение по умолчанию, нажмите кнопку *Использовать по умолчанию* в палитре цветов.
Дата	
Жирный	Установите флажок, чтобы отображать дату жирным шрифтом.
Цвет	Выберите цвет даты в палитре цветов. D обозначает цвет по умолчанию (зависит от темы веб-интерфейса). Чтобы вернуть значение по умолчанию, нажмите кнопку *Использовать по умолчанию* в палитре цветов.
Время	
Жирный	Установите флажок, чтобы отображать время жирным шрифтом.
Цвет	Выберите цвет времени в палитре цветов. D обозначает цвет по умолчанию (зависит от темы веб-интерфейса). Чтобы вернуть значение по умолчанию, нажмите кнопку *Использовать по умолчанию* в палитре цветов.
Секунды	Установите флажок, чтобы отображать секунды. В противном случае будут отображаться только часы и минуты.
Формат	Выберите отображение времени в 24-часовом или 12-часовом формате.
Часовой пояс	
Жирный	Установите флажок, чтобы отображать часовой пояс жирным шрифтом.
Цвет	Выберите цвет часового пояса в палитре цветов. D обозначает цвет по умолчанию (зависит от темы веб-интерфейса). Чтобы вернуть значение по умолчанию, нажмите кнопку *Использовать по умолчанию* в палитре цветов.
Часовой пояс	Выберите часовой пояс.
Формат	Выберите отображение часового пояса в кратком формате (например, `New York`) или полном формате (например, `(UTC-04:00) America/New York`).

3 Состояние обнаружения

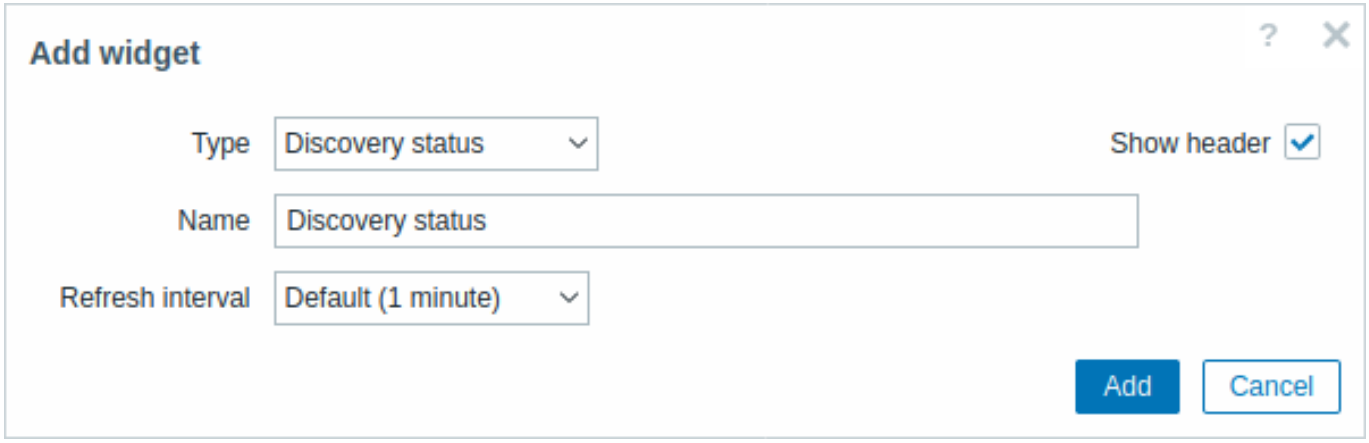
Обзор

Виджет *Состояние обнаружения* отображает состояние устройств, обнаруженных включёнными правилами сетевого обнаружения. Он помогает контролировать, корректно ли работает сетевое обнаружение, а также количество устройств, которые доступны или недоступны.



Настройка

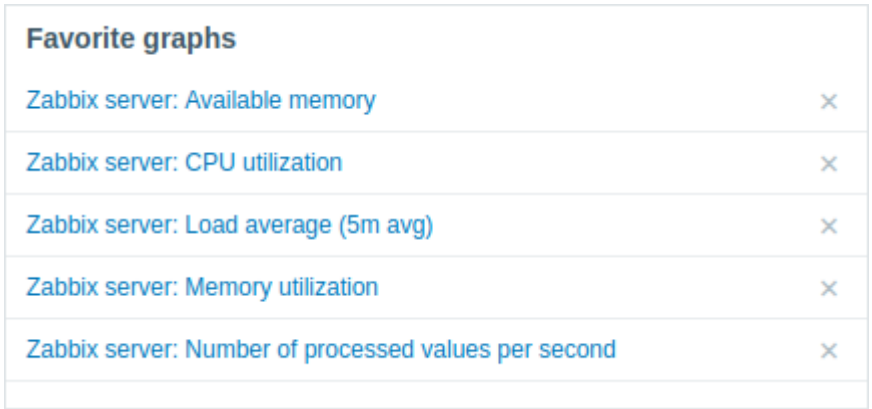
Все параметры настройки этого виджета являются общими для всех виджетов.



4 Избранные графики

Обзор

Виджет *Избранные графики* отображает алфавитный список ярлыков графиков, отмеченных текущим пользователем как избранные.



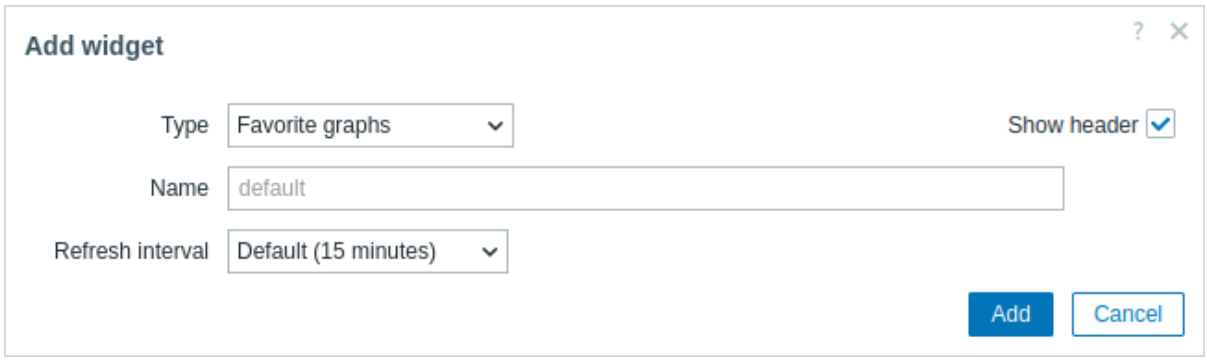
Нажатие x удаляет график из списка и из избранного.

Графики помечаются как избранные нажатием кнопки  *Добавить в избранное* при просмотре графиков элементов данных в разделе *Мониторинг* > *Последние данные*.



Настройка

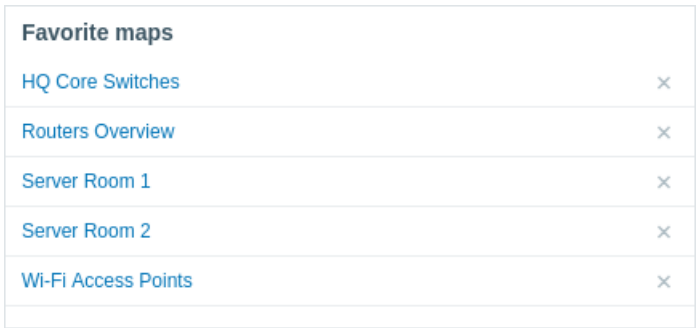
Все параметры настройки этого виджета являются общими для всех виджетов.



5 Избранные карты

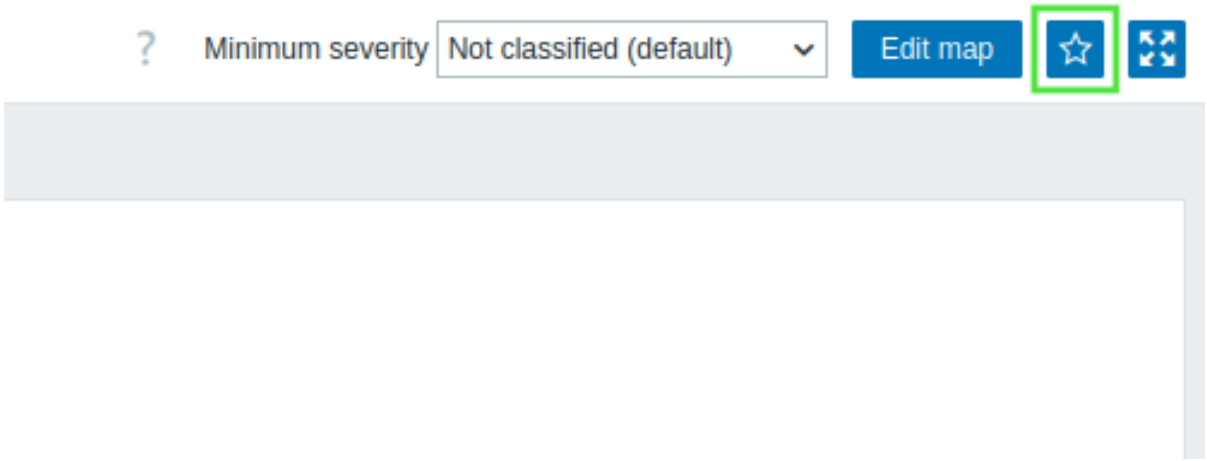
Обзор

Виджет *Избранные карты* отображает список ярлыков на карты, отмеченные текущим пользователем как избранные, отсортированный в алфавитном порядке.



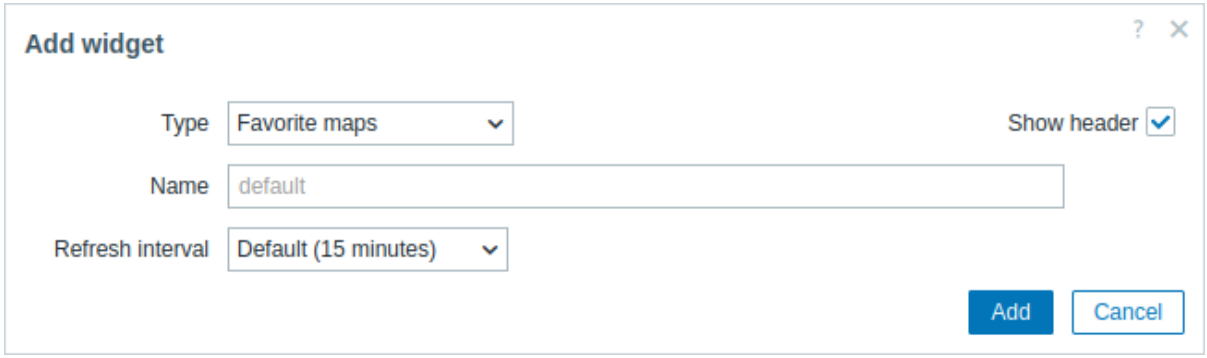
Нажатие x удаляет карту из избранного.

Карты отмечаются как избранные нажатием кнопки  *Добавить в избранное* при просмотре карт в разделе *Мониторинг* > *Карты*.



Настройка

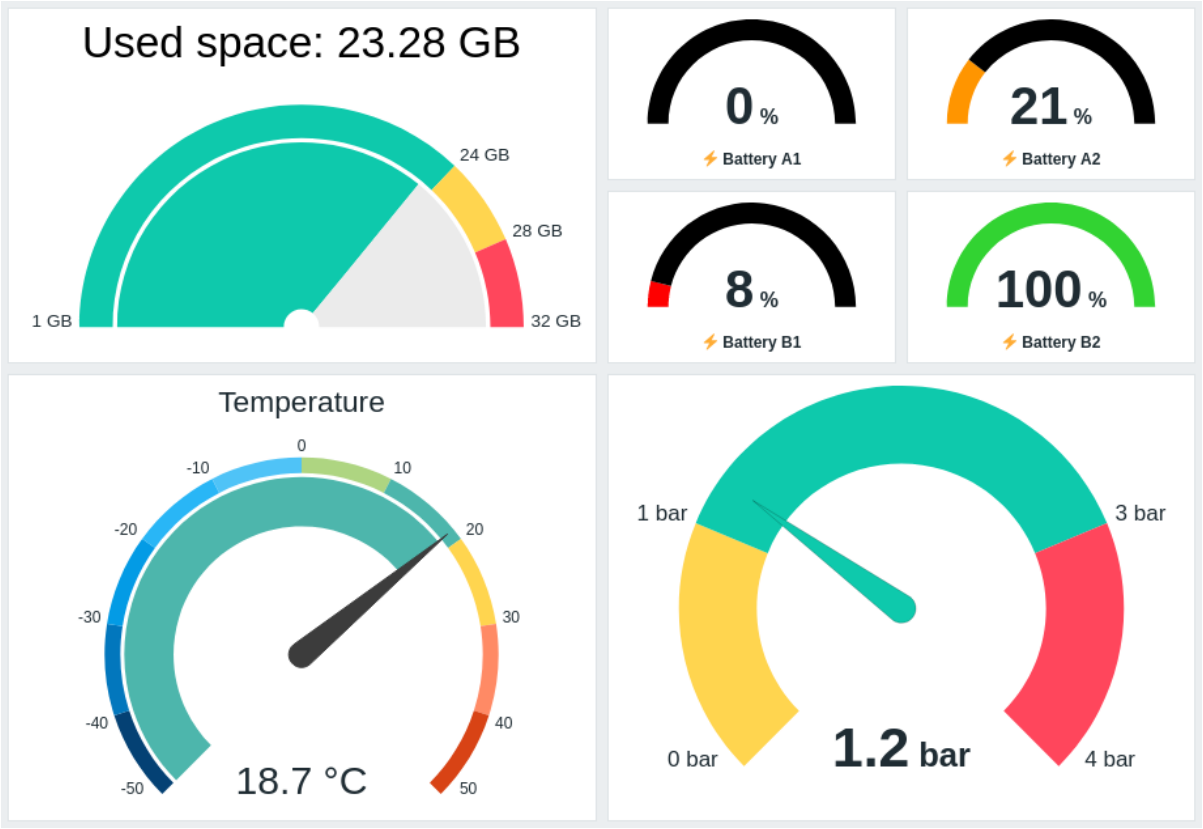
Все параметры настройки этого виджета являются общими для всех виджетов.



6 Калибр

Обзор

Виджет *Gauge* отображает значение одного числового элемента данных в виде шкалы. Он полезен для отслеживания важных метрик, визуализации пороговых значений и обнаружения резких изменений.



Вы можете настроить виджет для отображения следующего:

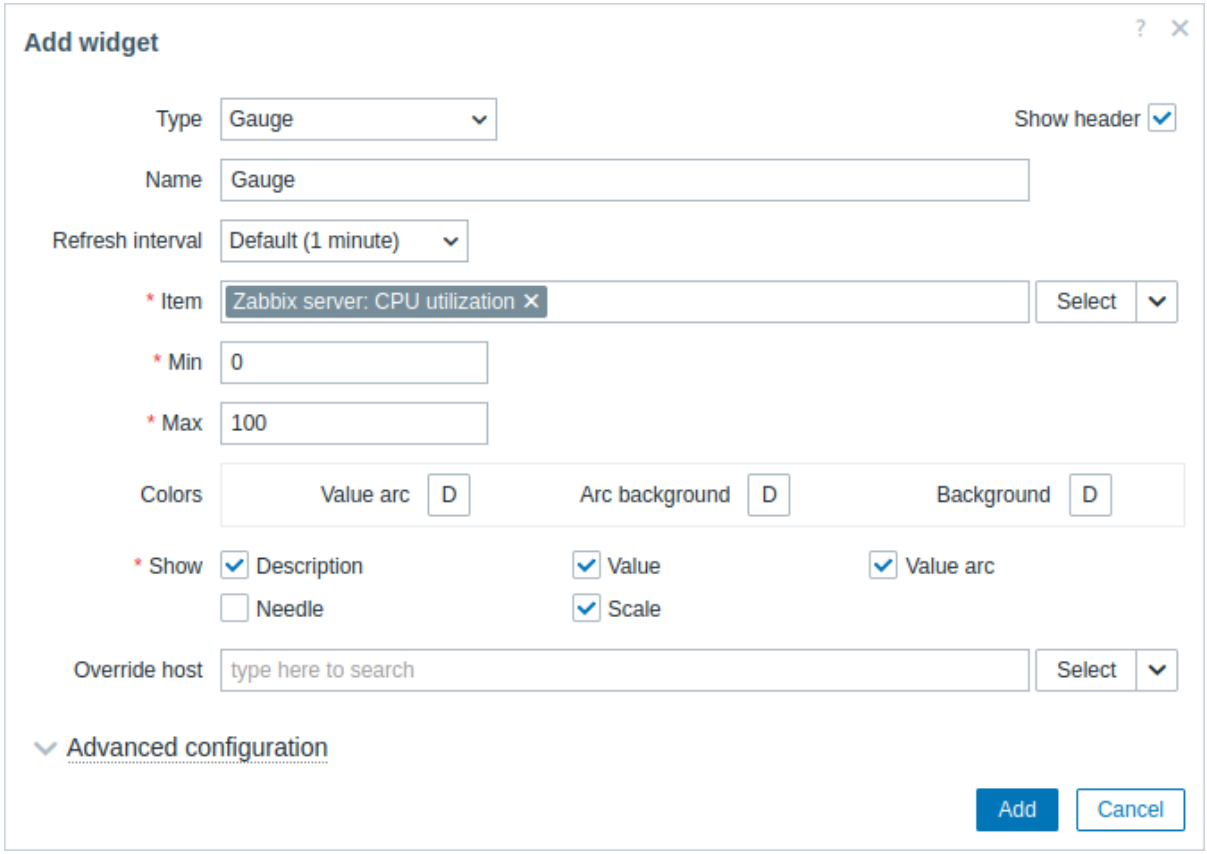
- Описание элемента данных (Used space: {ITEM.LASTVALUE}, Temperature)
- Значение элемента данных и единицы измерения (21 %, 18.7 °C, 1.2 bar)
- Шкала индикатора (1GB/24GB/28GB/32GB, -50/-40/-30/etc.)
- Дуга индикатора (дуга значения индикатора и дуга пороговых значений индикатора)
- Стрелка индикатора

Щелчок по виджету открывает простой график для элемента данных.

Информацию, отображаемую в виджете *Gauge*, можно загрузить как изображение PNG, выбрав параметр *Download image* в меню виджета.

Конфигурация

Чтобы настроить виджет, выберите *Gauge* в качестве типа:



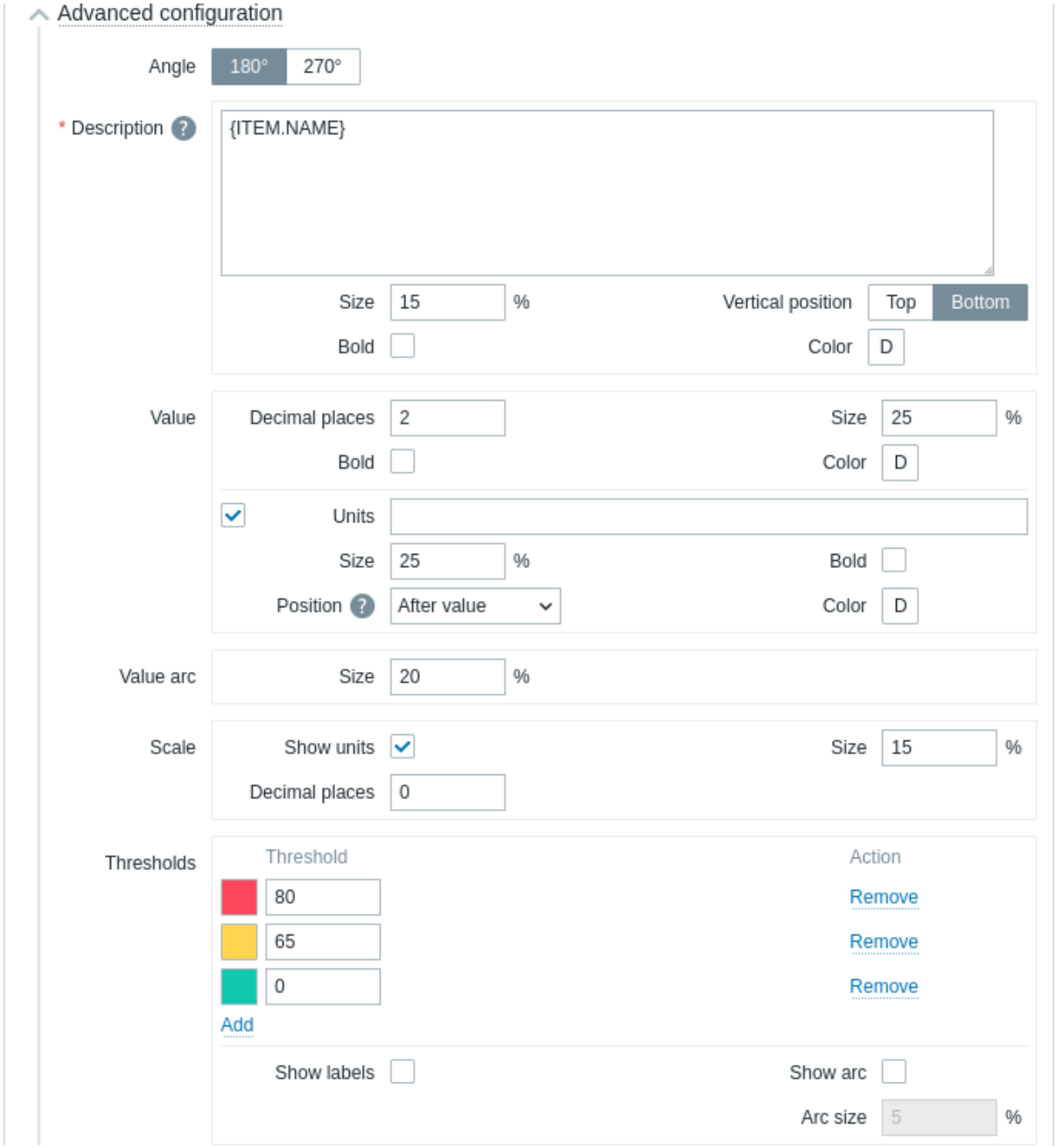
Помимо параметров, общих для всех виджетов, можно задать следующие специальные параметры:

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 предложен выпадающий список подходящих элементов данных. Обратите внимание, что можно выбирать только элементы данных, возвращающие числовые данные (кроме двоичных данных).
Min	Введите минимальное значение индикатора. Поддерживаются суффиксы (например, ”1d”, ”2w”, ”4K”, ”8G”). Поддерживаются преобразования значений.
Max	Введите максимальное значение индикатора. Поддерживаются суффиксы (например, ”1d”, ”2w”, ”4K”, ”8G”). Поддерживаются преобразования значений.
Colors	Выберите цвет в палитре цветов: **Value arc** — выберите цвет дуги значения индикатора; **Arc background** — выберите цвет фона дуги значения индикатора и дуги порогов индикатора; **Background** — выберите цвет фона виджета. ”D” обозначает 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 индикатора — описание, значение, дугу значения, стрелку, шкалу (минимальное и максимальное значение индикатора в начале и в конце дуги индикатора). Снимите флажок, чтобы скрыть элемент. Должен быть выбран как минимум один элемент. Обратите внимание, что стрелка и шкала индикатора могут отображаться, если отображается дуга значения индикатора или дуга порогов индикатора (см. параметры расширенной конфигурации). Также обратите внимание, что если отображается стрелка индикатора, значение размещается под стрелкой; если стрелка скрыта, значение выравнивается по нижнему краю дуги индикатора.
Override host	Выберите совместимый виджет или селектор узла сети панели как источник данных для узлов сети. Этот параметр недоступен при настройке виджета на панели шаблона.

Advanced configuration	Нажмите на метку *Advanced configuration*, чтобы отобразить параметры расширенной конфигурации. Здесь же можно будет настроить элементы индикатора, выбранные в поле *Show*.

Расширенная настройка

Параметры расширенной настройки доступны в сворачиваемом разделе *Расширенная настройка*:



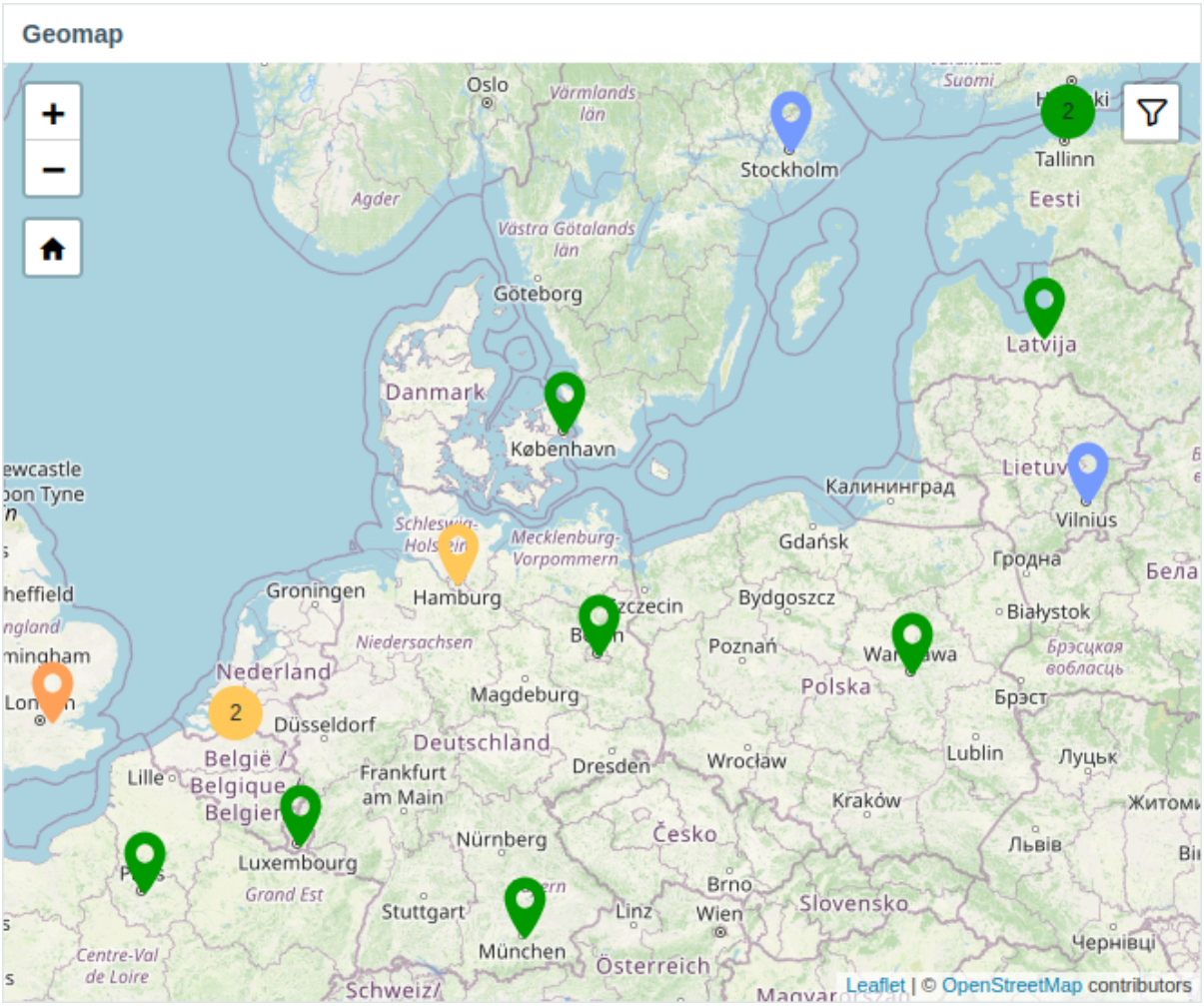
Угол	Выберите угол шкалы (180° или 270°).
Описание	
Описание	Введите описание элемента данных. Это описание может переопределить имя элемента данных по умолчанию. Поддерживаются многострочные описания. Возможно сочетание текста и поддерживаемых макросов. Поддерживаются {HOST.*}, {ITEM.*}, {INVENTORY.*} и пользовательские макросы.
Размер	Укажите высоту размера шрифта для описания элемента данных (в процентах относительно общей высоты виджета).
Вертикальное положение	Выберите вертикальное положение описания элемента данных (сверху или снизу относительно дуги шкалы).
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, который зависит от темы веб-интерфейса. Чтобы вернуть цвет по умолчанию, нажмите кнопку *Use default* в палитре цветов.
Значение	
Десятичные знаки	Укажите количество знаков после запятой для отображения значения. Этот параметр влияет только на элементы данных, возвращающие данные типа numeric (float).
Размер	Укажите высоту размера шрифта для значения (в процентах относительно высоты дуги шкалы).
Жирный	Установите этот флажок, чтобы отображать значение жирным шрифтом.
Цвет	Выберите цвет значения в палитре цветов. ”D” обозначает цвет по умолчанию, который зависит от темы веб-интерфейса. Чтобы вернуть цвет по умолчанию, нажмите кнопку *Use default* в палитре цветов.
Единицы измерения	
Единицы измерения	Установите этот флажок, чтобы отображать единицы измерения вместе со значением элемента данных. Если вы введете название единицы измерения, оно переопределит единицы измерения, заданные в настройке элемента данных.
Размер	Укажите высоту размера шрифта для единиц измерения элемента данных (в процентах относительно высоты дуги шкалы).
Жирный	Установите этот флажок, чтобы отображать единицы измерения элемента данных жирным шрифтом.
Положение	Выберите положение единиц измерения элемента данных (сверху, снизу, перед или после относительно значения элемента данных). Этот параметр игнорируется для следующих единиц измерения, связанных со временем: unixtime, uptime, s.
Цвет	Выберите цвет единиц измерения элемента данных в палитре цветов. ”D” обозначает цвет по умолчанию, который зависит от темы веб-интерфейса. Чтобы вернуть цвет по умолчанию, нажмите кнопку *Use default* в палитре цветов.
Дуга значения	
Размер дуги	Укажите высоту размера дуги значения шкалы (в процентах относительно радиуса дуги шкалы).
Стрелка	
Цвет	Выберите цвет стрелки шкалы в палитре цветов. ”D” обозначает цвет по умолчанию, который зависит от темы веб-интерфейса. Если заданы *Пороги*, цвет стрелки по умолчанию зависит от цвета порога. Чтобы вернуть цвет по умолчанию, нажмите кнопку *Use default* в палитре цветов.
Шкала	
Показывать единицы измерения	Установите этот флажок, чтобы отображать единицы измерения вместе с минимальным и максимальным значением шкалы.
Размер	Укажите высоту размера шрифта для минимального и максимального значения шкалы (в процентах относительно высоты дуги шкалы).
Десятичные знаки	Укажите количество знаков после запятой для отображения минимального и максимального значения шкалы. Этот параметр влияет только на элементы данных, возвращающие данные типа numeric (float).
Пороги	
Пороги	Нажмите *Add*, чтобы добавить порог, выберите цвет порога в палитре цветов и укажите числовое значение. При сохранении список порогов будет отсортирован по возрастанию. Обратите внимание, что цвета, настроенные как пороги, будут корректно отображаться только для числовых элементов данных. Поддерживаются суффиксы (например, ”1d”, ”2w”, ”4K”, ”8G”). Поддерживаются преобразования значений.
Показывать подписи	Установите этот флажок, чтобы отображать значения порогов как подписи на шкале.
Показывать дугу	Установите этот флажок, чтобы отображать дугу порогов шкалы.
Размер дуги	Укажите высоту размера дуги порогов шкалы (в процентах относительно радиуса дуги шкалы).

7 Географическая карта

Обзор

Виджет *Geomap* отображает узлы сети в виде маркеров на интерактивной карте на базе Leaflet, при этом цвет маркера указывает на наиболее серьёзную проблему каждого узла сети.

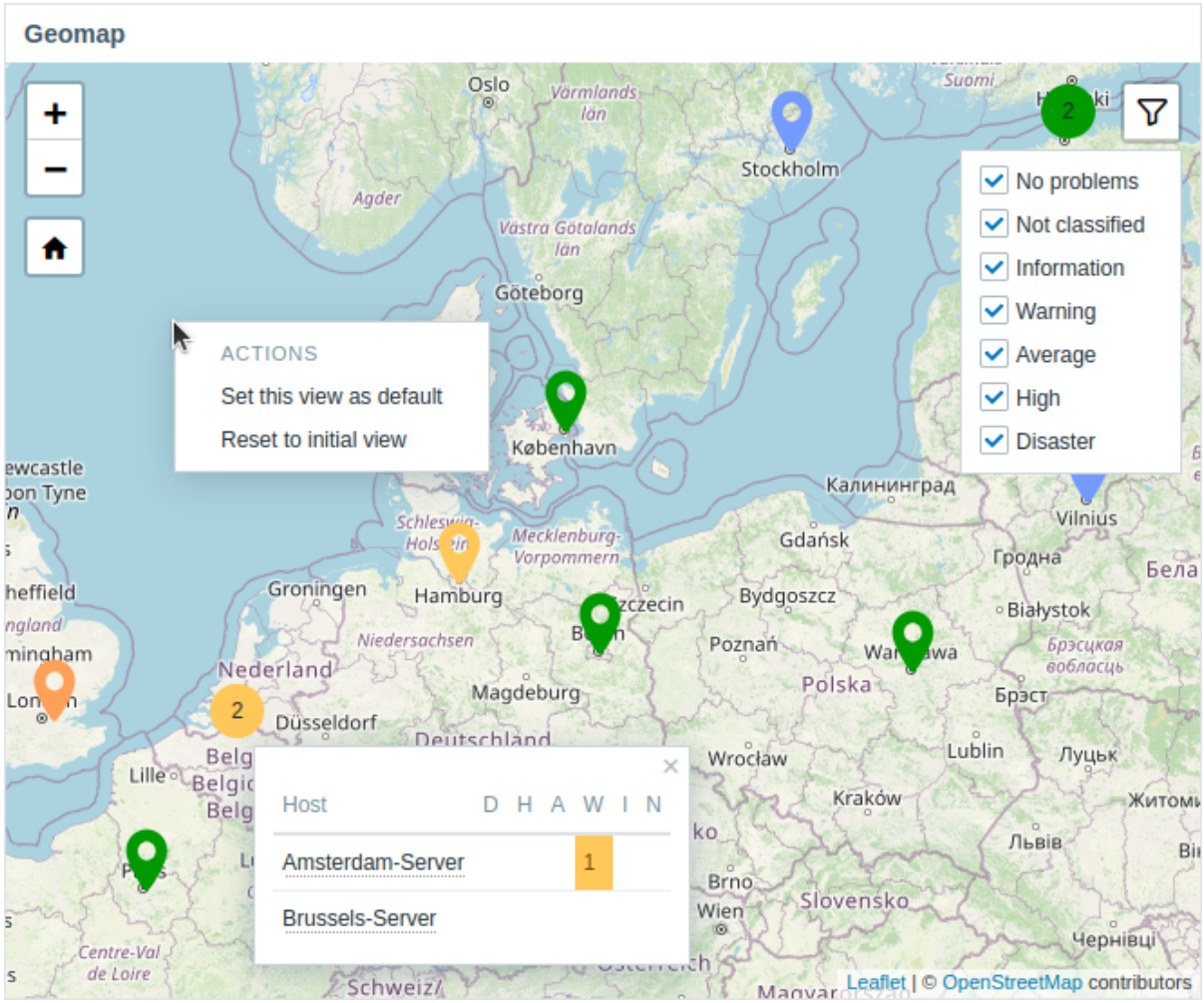


Note:
Zabbix предлагает несколько предопределённых поставщиков тайлов карты, а также позволяет добавлять пользовательских в *Administration* > *General* > *Geographical maps*.

По умолчанию виджет отображает все включённые узлы сети, имеющие корректные координаты (широта от -90 до 90, долгота от -180 до 180) в своём инвентаре.

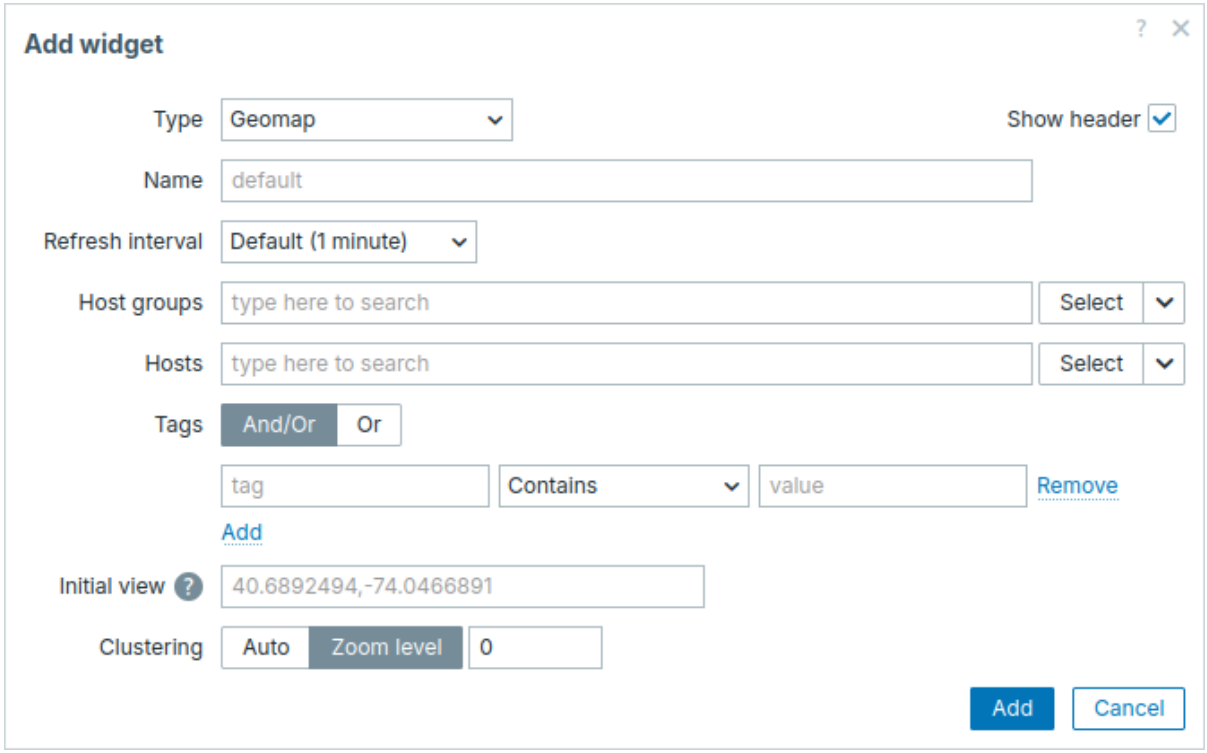
Для взаимодействия с виджетом используйте следующие возможности:

- При щелчке по маркеру отображаются имя узла сети и неустранённые проблемы; щелчок по имени узла сети открывает меню узла сети.
- Фильтруйте узлы сети по серьёзности проблем с помощью значка фильтра в правом верхнем углу.
- Щёлкните правой кнопкой мыши по карте, чтобы задать или сбросить вид по умолчанию; используйте значок дома, чтобы вернуться к виду по умолчанию.



Настройка

Чтобы добавить виджет, выберите *Geomap* в качестве типа.



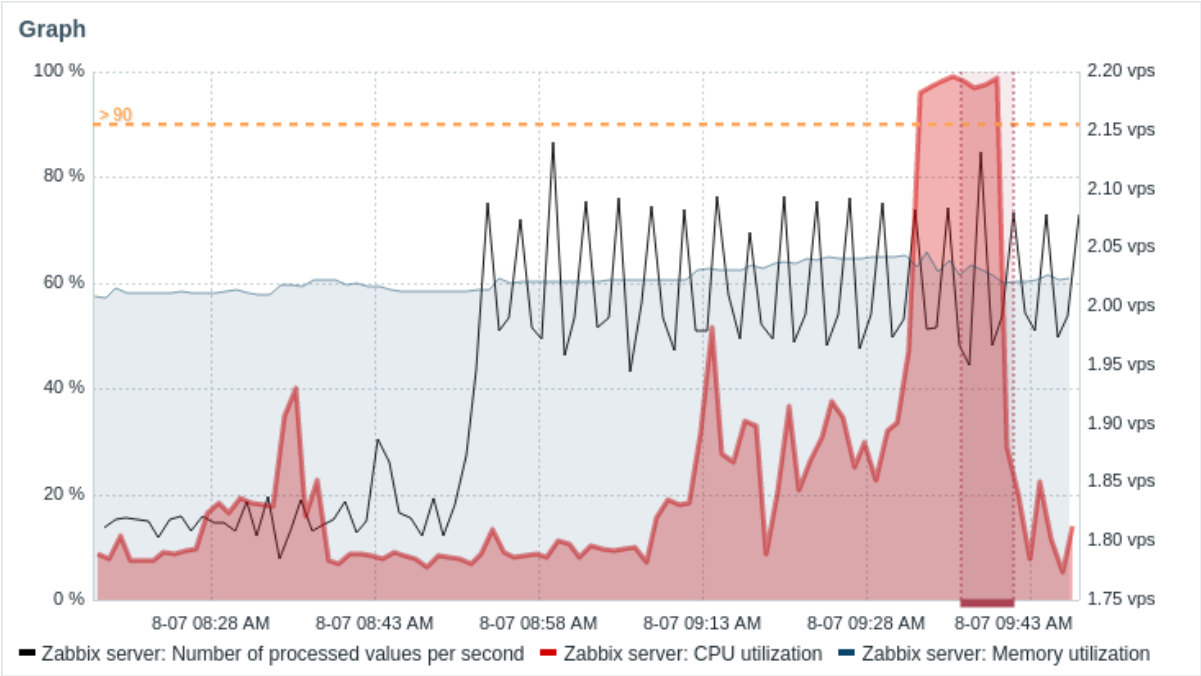
В дополнение к параметрам, которые являются общими для всех виджетов, вы можете задать следующие специальные параметры:

Группы узлов сети	Выберите группы узлов сети, которые будут отображаться на карте. 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. Если в полях *Группы узлов сети* и *Узлы сети* ничего не выбрано, будут отображены все узлы сети с корректными координатами. Этот параметр недоступен при настройке виджета на панели шаблона.
Узлы сети	Выберите узлы сети, которые будут отображаться на карте. Либо выберите совместимый виджет или панель в качестве источника данных для узлов сети. Это поле поддерживает автодополнение, поэтому при вводе имени узла сети будет предложен выпадающий список подходящих узлов сети. Если в полях *Группы узлов сети* и *Узлы сети* ничего не выбрано, будут отображены все узлы сети с корректными координатами. Этот параметр недоступен при настройке виджета на панели шаблона.
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 имен тегов всегда чувствительно к регистру. Для каждого условия доступны несколько операторов: **Существует** — включить указанные имена тегов; **Равно** — включить указанные имена и значения тегов (с учетом регистра); **Содержит** — включить указанные имена тегов, у которых значения содержат введенную строку (поиск подстроки, без учета регистра); **Не существует** — исключить указанные имена тегов; **Не равно** — исключить указанные имена и значения тегов (с учетом регистра); **Не содержит** — исключить указанные имена тегов, у которых значения 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 панели шаблона.
Начальный вид	Координаты центра, разделенные запятыми, и необязательный уровень масштабирования, которые будут отображаться при первоначальной загрузке виджета, в формате `<latitude>,<longitude>,<zoom>` Если начальный масштаб указан, виджет Geomap загружается с заданным уровнем масштабирования. В противном случае начальный масштаб вычисляется как половина 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 со счетчиком: **Auto** — маркеры объединяются автоматически; **Zoom level** — маркеры объединяются на основе указанного уровня масштабирования (0 — максимальное отдаление, 30 — максимальное приближение). Например, если параметр установлен в 0, маркеры не объединяются ни на одном уровне масштабирования; если установлено 15, маркеры объединяются начиная с максимального отдаления до уровня масштабирования 15; если установлено 30, маркеры объединяются на всех уровнях масштабирования. Обратите внимание, что максимальный уровень приближения определяется поставщиком тайлов карты.

8 График

Обзор

Виджет *Graph* отображает числовые данные элементов данных в виде векторного графика. Он помогает отслеживать тенденции, выявлять проблемы и сравнивать значения во времени и между узлами сети.



При наведении курсора на график отображается всплывающая подсказка со списком элементов данных, отсортированных по значению в порядке убывания.

При наведении курсора на элемент данных во всплывающей подсказке его график выделяется (остальные приглушаются); при выборе элемента данных его данные передаются другим виджетам.

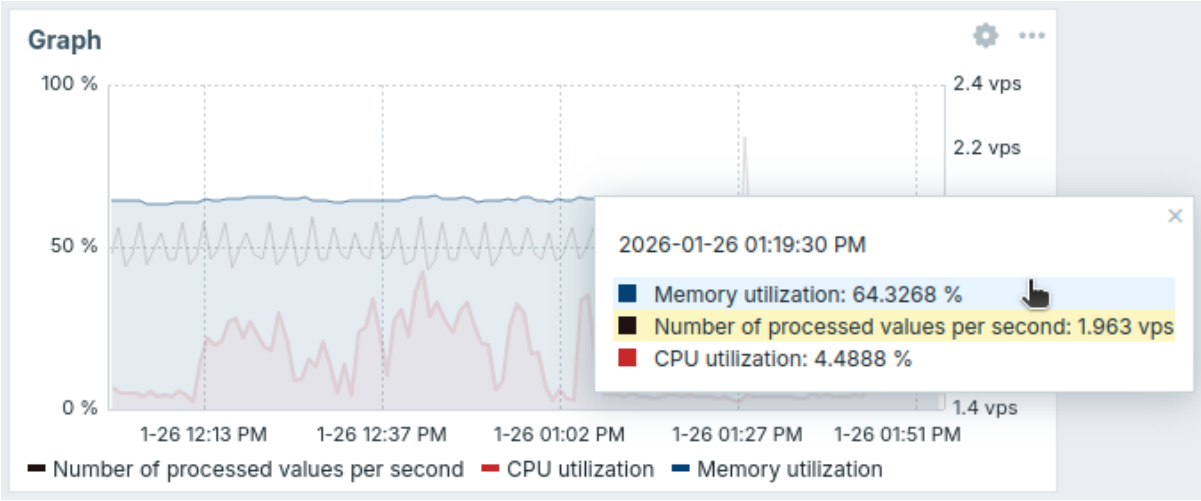
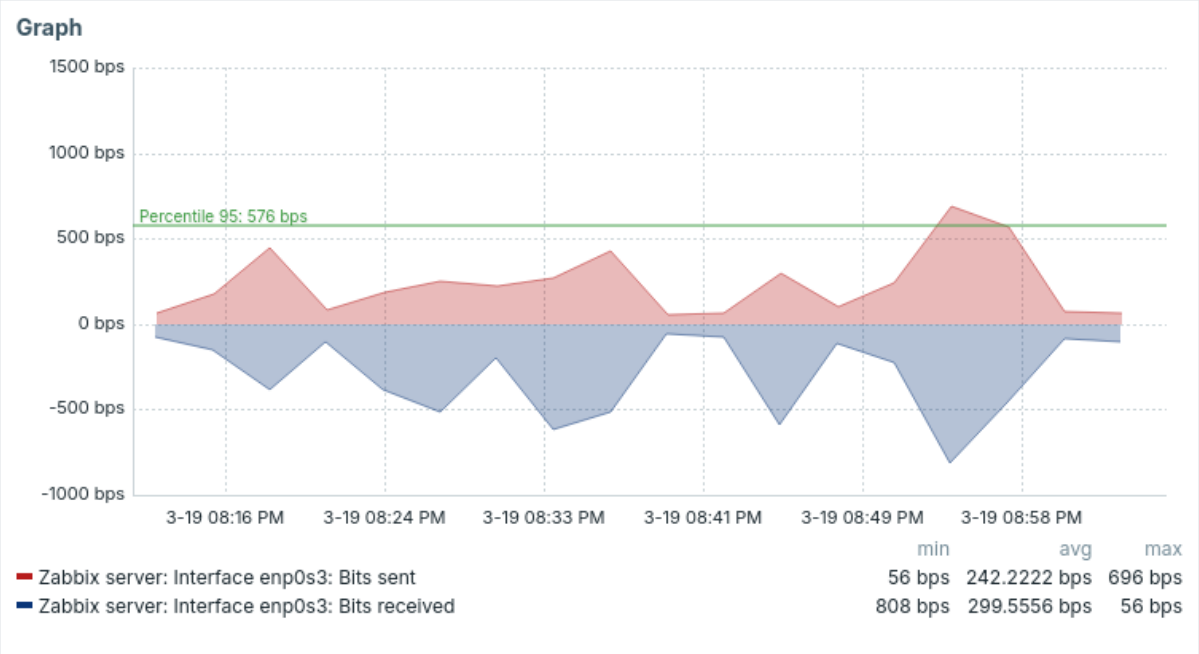


График может отображаться с инвертированными значениями по оси Y; это влияет только на визуальное представление и не изменяет исходные значения.

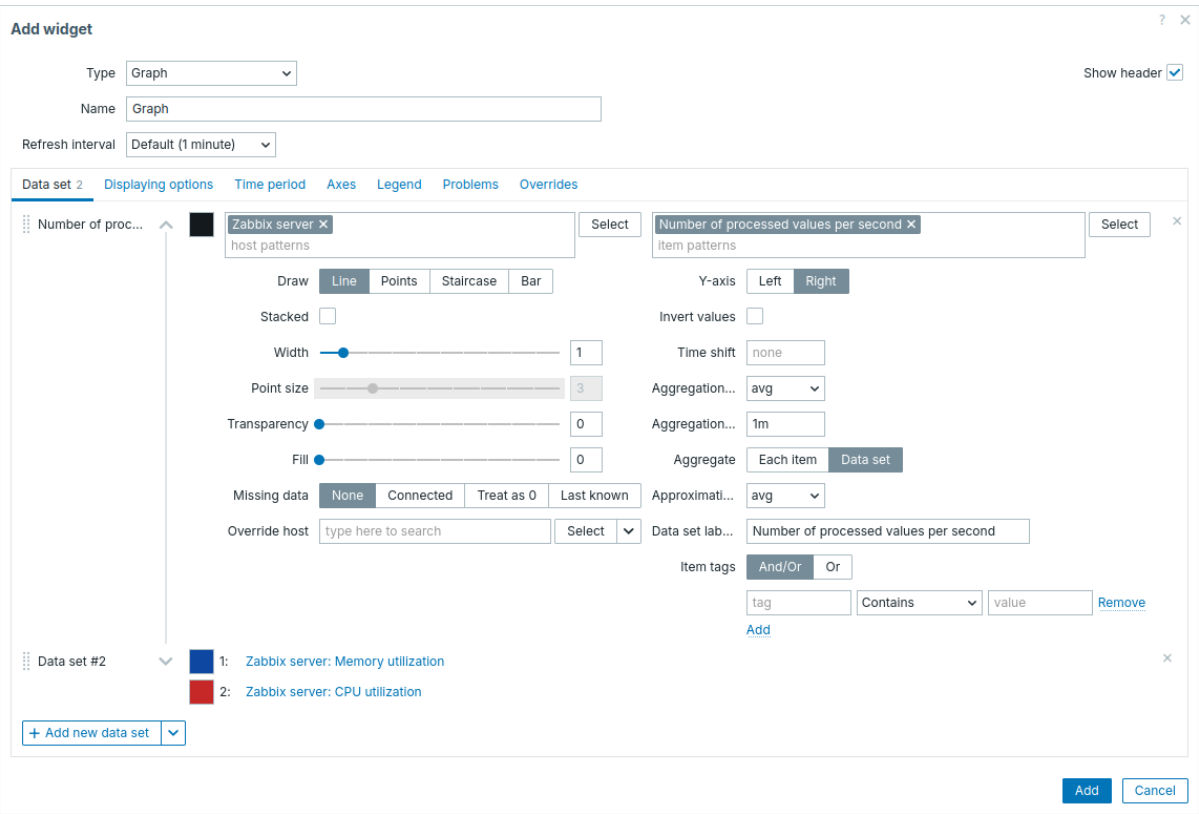


Также доступен устаревший виджет Graph (classic).

Информацию, отображаемую в виджете *Graph*, можно загрузить как изображение PNG, выбрав параметр *Download image* в меню виджета.

Настройка

Чтобы настроить, выберите *Graph* в качестве типа:



Набор данных

Вкладка **Набор данных** позволяет выбирать данные для графика, добавляя наборы данных. Можно добавить два типа наборов данных:

- *Шаблоны элементов данных* — отображаются данные из соответствующих элементов данных. Можно выбрать один базовый цвет или строку палитры, чтобы назначить разные цвета каждому найденному элементу данных.
- *Список элементов данных* — отображаются данные из выбранных элементов данных. Для каждого элемента данных можно выбрать цвет 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 узла сети и элемента данных; данные элементов данных, соответствующих этим шаблонам, будут отображаться на графике; может отображаться 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 интерпретируется всегда, поэтому невозможно, например, добавить отдельно элемент данных с именем *item** , если существуют другие подходящие элементы данных (например, *item2*, *item3*). Указание шаблонов узла сети и элемента данных обязательно для наборов данных типа "Шаблоны элементов данных". См. также: Подробности настройки набора данных. Для набора данных **Список элементов данных**: Выберите элементы данных для графика, нажав кнопку *Добавить элемент данных*. Также можно выбрать совместимые виджеты как источник данных для элементов данных, нажав кнопку *Добавить виджет*. Указание элементов данных или виджетов обязательно для наборов данных типа "Список элементов данных". См. также: Подробности настройки набора данных. Обратите внимание, что допускаются 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.
Отрисовка	Выберите тип отрисовки метрики. Возможные типы отрисовки: *Линия* (по умолчанию), *Точки*, *Ступенчатая*, *Столбцы*. Обратите внимание, что если на линейном/ступенчатом графике есть только одна точка данных, она отображается как точка независимо от типа отрисовки. Размер точки вычисляется на основе ширины линии, но не может быть меньше 3 пикселей, даже если ширина линии меньше.
С накоплением	Установите этот флажок, чтобы отображать данные с накоплением (с отображением заполненных областей). Этот параметр недоступен, если выбран тип отрисовки *Точки*.
Ширина	Укажите ширину линии. Этот параметр доступен, если выбран тип отрисовки *Линия* или *Ступенчатая*.
Размер точки	Укажите размер точки. Этот параметр доступен, если выбран тип отрисовки *Точки*.
Прозрачность	Укажите уровень прозрачности.
Заполнение	Укажите уровень заполнения. Этот параметр доступен, если выбран тип отрисовки *Линия* или *Ступенчатая*.
Отсутствующие данные	Выберите вариант отображения отсутствующих данных: **Нет** — разрыв остается пустым; **Соединять** — два граничных значения соединяются; **Считать как 0** — отсутствующие данные отображаются как значения 0; **Последнее известное** — отсутствующие данные отображаются с тем же значением, что и последнее известное значение; не применяется для типа отрисовки *Точки* и *Столбцы*.
Переопределить узел сети	Выберите совместимый виджет или селектор узла сети панели как источник данных для узлов сети. Этот параметр недоступен при настройке виджета на панели шаблона.
Ось Y	Выберите сторону графика, на которой будет отображаться ось Y.

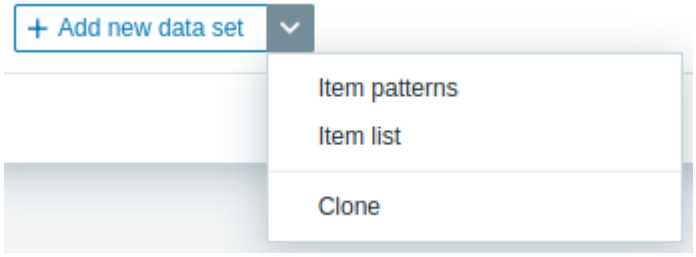
Инвертировать значения	Установите этот флажок, чтобы перед отображением на графике умножить все значения оси Y на -1. Это влияет только на визуальное представление; исходные данные остаются без изменений. Этот параметр не влияет на всплывающую подсказку, легенду или линию процентиля. Если для оси выбран параметр *Простые триггеры*, пороговые значения триггеров будут отображаться инвертированными.
Сдвиг времени	При необходимости укажите сдвиг времени. В этом поле можно использовать суффиксы времени. Допускаются отрицательные значения.
Функция агрегации	Укажите, какую функцию агрегации использовать: **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; **none** — отображать все значения (без агрегации). Агрегация позволяет отображать агрегированное значение для выбранного интервала (5 минут, час, день) вместо всех значений. См. также: Агрегация на графиках.
Интервал агрегации	Укажите интервал для агрегации значений. В этом поле можно использовать суффиксы времени. Числовое значение без суффикса будет рассматриваться как секунды. Обратите внимание, что если виджет настроен на отображение исторических данных на основе трендов (*Выбор исторических данных* установлен в *Тренды* или *Авто*), рекомендуется использовать интервал агрегации, кратный 1 часу (например, 3600, 60m, 1h, 3h и т. д.). Тренды хранят почасовые агрегированные значения, поэтому использование интервала агрегации, не кратного 1 часу (например, 100s, 7min, 15min, 90min и т. д.), может привести к результатам, которые трудно интерпретировать.
Агрегировать	Укажите, как выполнять агрегацию: **Каждый элемент данных** — каждый элемент данных в наборе данных будет агрегирован и отображен отдельно; **Набор данных** — все элементы данных набора будут агрегированы и отображены 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, за год значений, собираемых каждые 10 минут).
Метка набора данных	Укажите метку набора данных, которая отображается в настройке графика *Набор данных* и в *Легенде* графика (для агрегированных наборов данных). Нумеруются все наборы данных, включая те, для которых указана *Метка набора данных*. Если метка не указана, набор данных будет автоматически помечен в соответствии со своим номером (например, ”Набор данных #2”, ”Набор данных #3” и т. д.). Нумерация наборов данных пересчитывается после изменения порядка/перетаскивания наборов данных. Слишком длинные метки набора данных будут сокращены по размеру доступного места при отображении (например, ”Количество proc...”).

Теги элементов данных	Укажите теги для фильтрации элементов данных, отображаемых 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 несколько операторов: **Существует** — включить указанные имена тегов; **Равно** — включить указанные имена и значения тегов (с учетом регистра); **Содержит** — включить указанные имена тегов, если значения тегов содержат введенную строку (поиск подстроки, без учета регистра); **Не существует** — исключить указанные имена тегов; **Не равно** — исключить указанные имена и значения тегов (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, при этом условия с одинаковым именем тега будут сгруппированы по условию Или; **Или** — достаточно выполнения одного условия.

Сведения о настройке наборов данных

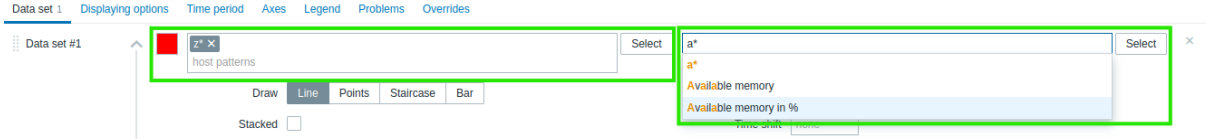
Существующие наборы данных отображаются в списке. Вы можете:

- Нажать на значок перемещения и перетащить набор данных на новое место в списке.
- Нажать на значок разворачивания, чтобы развернуть сведения о наборе данных. При разворачивании этот значок меняется на значок сворачивания.
- Нажать на значок цвета, чтобы открыть палитру выбора цвета. Вы можете ввести hex-код, выбрать сплошной образец цвета или перейти на вкладку *Palette* и выбрать строку предопределённых цветов. Выбранный цвет применяется напрямую для наборов данных *Item list* или используется как основа для генерируемых оттенков в *Item patterns*. Используйте Tab для перехода между элементами управления диалогового окна, клавиши со стрелками для навигации по образцам цвета или строкам палитры, Enter для выбора и Esc для отмены.
- Нажать кнопку *Add new data set*, чтобы добавить пустой набор данных, позволяющий выбрать шаблоны узлов сети и элементов данных. Если вы нажмёте на значок со стрелкой вниз рядом с кнопкой *Add new data set*, появится выпадающее меню, позволяющее добавить новый набор данных *Item patterns* или *Item list*, либо *Clone* текущий открытый набор данных. Если все наборы данных свернуты, опция *Clone* недоступна.

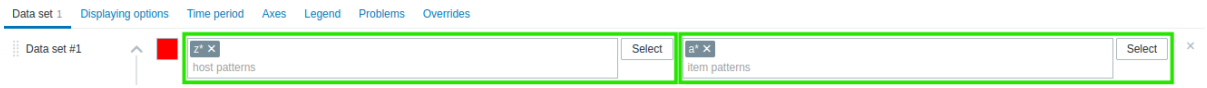


Набор данных **Шаблоны элементов данных** содержит поля *Шаблоны узлов сети* и *Шаблоны элементов данных*, которые оба распознают полные имена или шаблоны, содержащие символ подстановки (*). Эта функциональность позволяет выбрать все имена узлов сети и имена элементов данных, содержащие выбранный шаблон. При вводе имени элемента данных или шаблона элемента данных в поле *Шаблоны элементов данных* в раскрывающемся списке отображаются только элементы данных, принадлежащие выбранному имени(именам) узла сети.

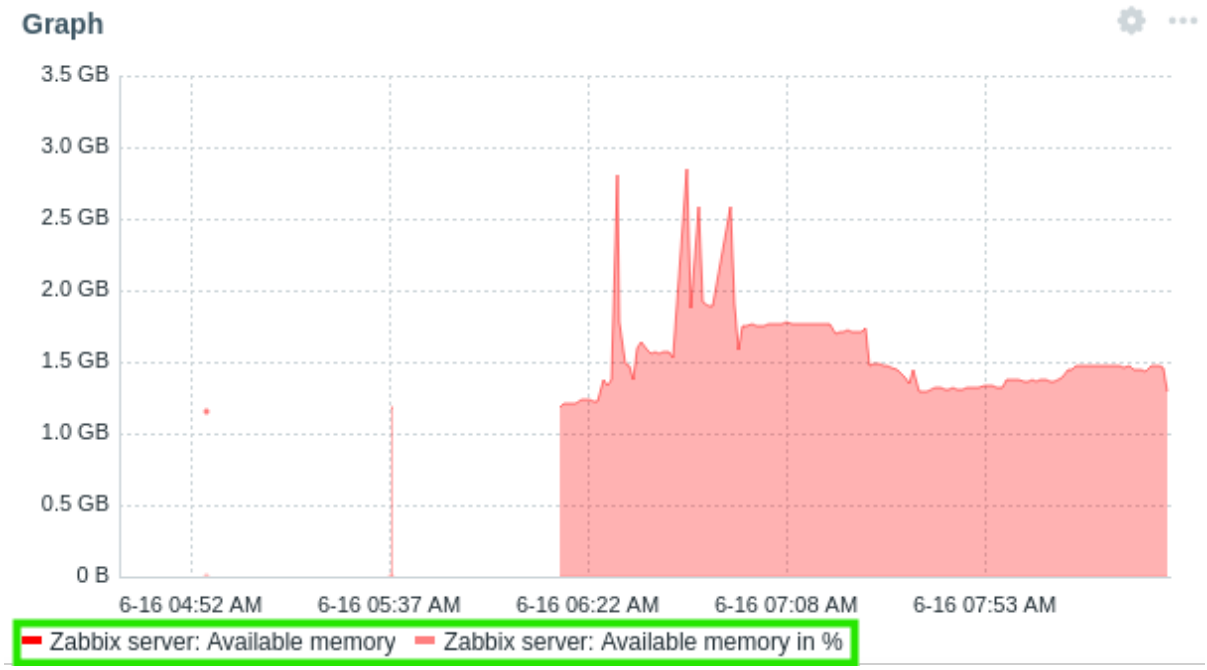
Например, если в поле *Шаблоны узлов сети* ввести шаблон **z***, в раскрывающемся списке будут показаны все имена узлов сети, содержащие этот шаблон: **z*, Zabbix server, Zabbix proxy**. После нажатия *Enter* этот шаблон принимается и отображается как **z***. Аналогично, если в поле *Шаблоны элементов данных* ввести шаблон **a***, в раскрывающемся списке будут показаны все имена элементов данных, содержащие этот шаблон: **a*, Available memory, Available memory in %**.



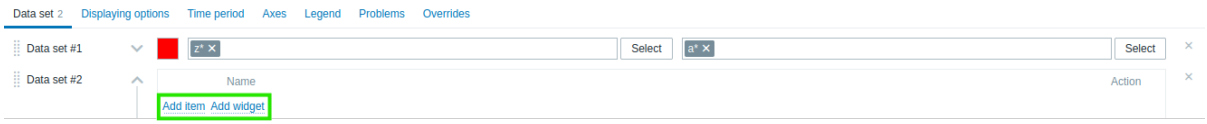
После нажатия *Enter* шаблон принимается и отображается как **a***.



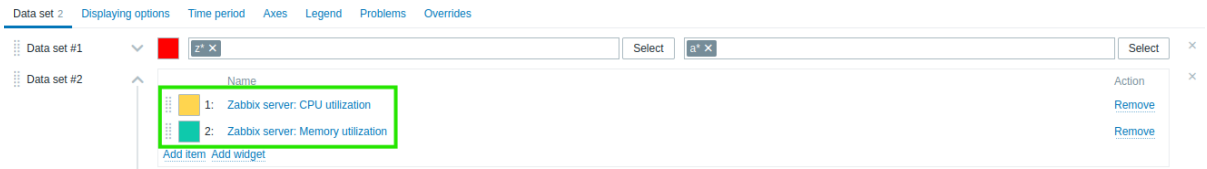
Затем график отображает все элементы данных, принадлежащие выбранному имени(именам) узла сети.



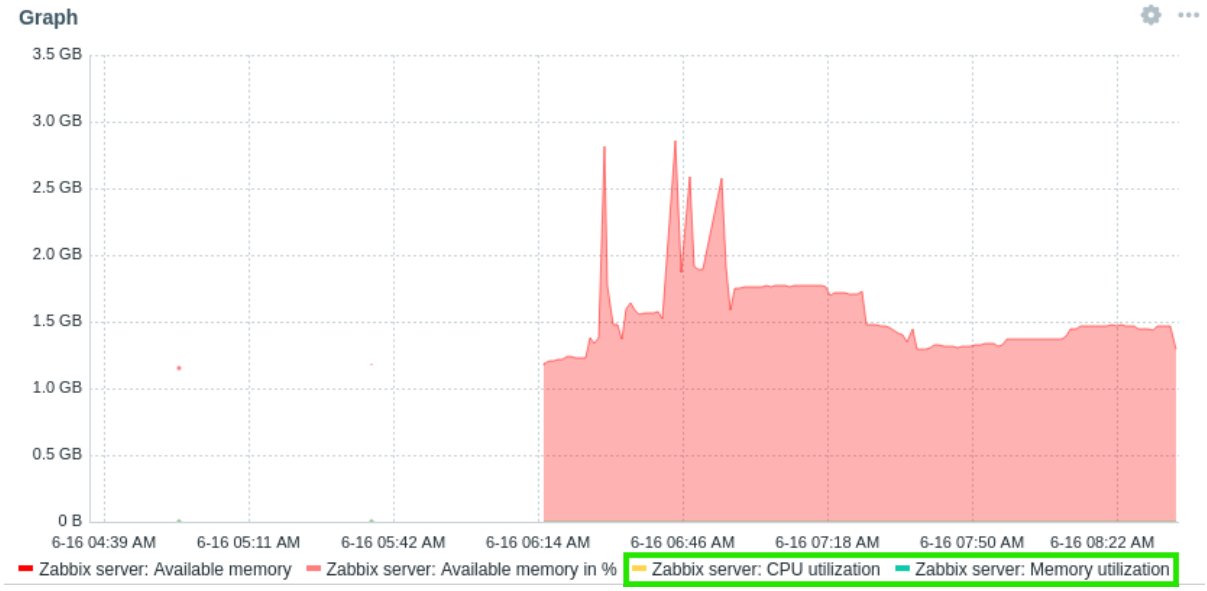
Набор данных **Список элементов данных** содержит кнопку *Добавить элемент данных*, которая позволяет добавлять элементы данных для отображения на графике. Вы также можете добавлять совместимые виджеты в качестве источника данных для элементов данных, нажав кнопку *Добавить виджет*.



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.

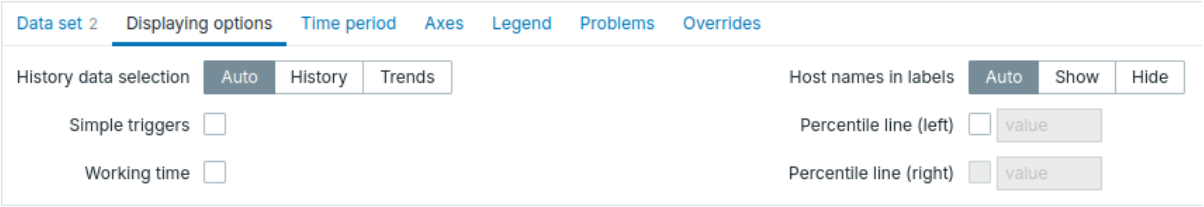


После выбора одного или нескольких элементов данных они будут отображены в списке элементов данных набора данных и на графике.



Параметры отображения

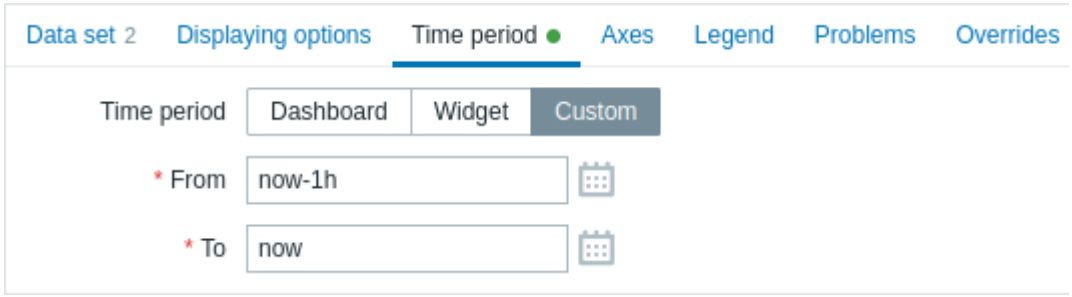
Вкладка **Параметры отображения** позволяет определить выбор данных истории:



Выбор данных истории	Укажите источник данных графика: **Auto** — данные выбираются в соответствии с классическим алгоритмом графика (по умолчанию); **History** — данные из истории; **Trends** — данные из трендов.
Простые триггеры	Установите флажок, чтобы показывать пороговые значения триггера для простых триггеров. Пороговые значения будут отображаться 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 пределах отображаемого диапазона, чтобы быть видимым.
Рабочее время	Установите флажок, чтобы показывать рабочее время на графике. Рабочее время (рабочие дни) отображается на графиках белым фоном, а нерабочее время — серым (при использовании темы веб-интерфейса по умолчанию *Original blue*).
Имена узлов сети в подписях	Выберите, отображать ли имена узлов сети в легенде графика и во всплывающей подсказке, которая появляется при наведении курсора на данные графика: **Auto** — имена узлов сети отображаются только в том случае, если в наборах данных присутствует более одного узла сети (по умолчанию). **Show** — имена узлов сети отображаются. **Hide** — имена узлов сети скрыты. Этот параметр недоступен при настройке виджета на панели шаблона.
Линия процентиля (слева)	Установите флажок и введите значение процентиля, чтобы показать указанный процентиль в виде линии на левой оси Y графика. Если, например, установлен 95% процентиль, то линия процентиля будет находиться на уровне, ниже которого лежат 95 процентов значений.
Линия процентиля (справа)	Установите флажок и введите значение процентиля, чтобы показать указанный процентиль в виде линии на правой оси Y графика. Если, например, установлен 95% процентиль, то линия процентиля будет находиться на уровне, ниже которого лежат 95 процентов значений.

Период времени

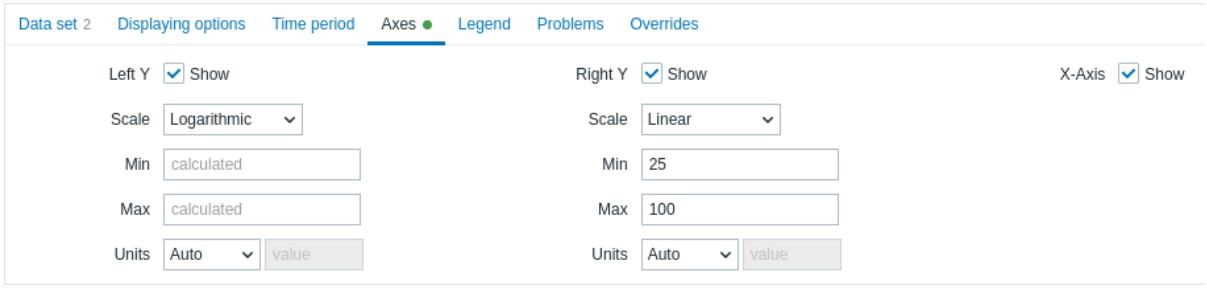
Вкладка **Период времени** позволяет задать период времени, за который будут отображаться данные на графике:



Период времени	Выберите источник данных для периода времени: **Панель** — использовать селектор периода времени панели; **Виджет** — использовать совместимый виджет (задается в параметре *Виджет*); **Пользовательский** — использовать пользовательский период времени, заданный в параметрах *С* и *По*; если он задан, в правом верхнем углу виджета будет отображаться значок часов, указывающий установленное время при наведении курсора мыши. 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".
С	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".
По	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".

Оси

Вкладка **Оси** позволяет настроить отображение осей:

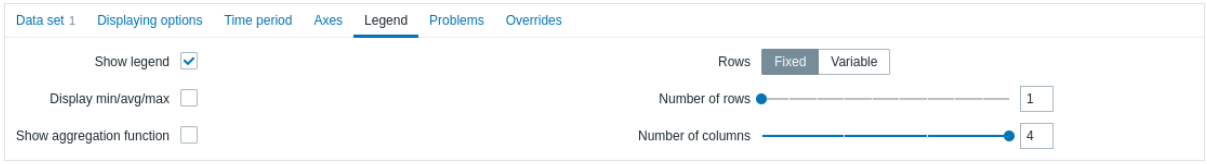


Левая Y	Установите этот флажок, чтобы сделать левую ось Y видимой. Флажок может быть недоступен, если она не выбрана ни на вкладке *Набор данных*, ни на вкладке *Переопределения*.
Правая Y	Установите этот флажок, чтобы сделать правую ось Y видимой. Флажок может быть недоступен, если она не выбрана ни на вкладке *Набор данных*, ни на вкладке *Переопределения*.
Ось X	Снимите этот флажок, чтобы скрыть ось X (по умолчанию отмечен).
Масштаб	Выберите масштаб значений оси графика из выпадающего списка: **Линейный** — значения оси увеличиваются на фиксированную величину (например, 10, 20, 30), подходит для данных, которые изменяются равномерно или охватывают небольшой либо средний диапазон; **Логарифмический** — значения оси увеличиваются экспоненциально (например, 10, 100, 1000), подходит для данных, которые быстро изменяются или охватывают большой диапазон.
Мин	Задайте минимальное значение соответствующей оси. Указывается минимальное значение видимого диапазона оси Y.

Макс	Задайте максимальное значение соответствующей оси. Указывается максимальное значение видимого диапазона оси Y.
Единицы	Выберите единицу измерения для значений оси графика из выпадающего списка: **Авто** — значения оси отображаются с использованием единицы измерения первого элемента данных в наборе данных; **Статический** — значения оси отображаются с использованием единицы измерения, указанной в поле ввода *значение*; если поле оставлено пустым, отображаются только числовые значения.

Условные обозначения

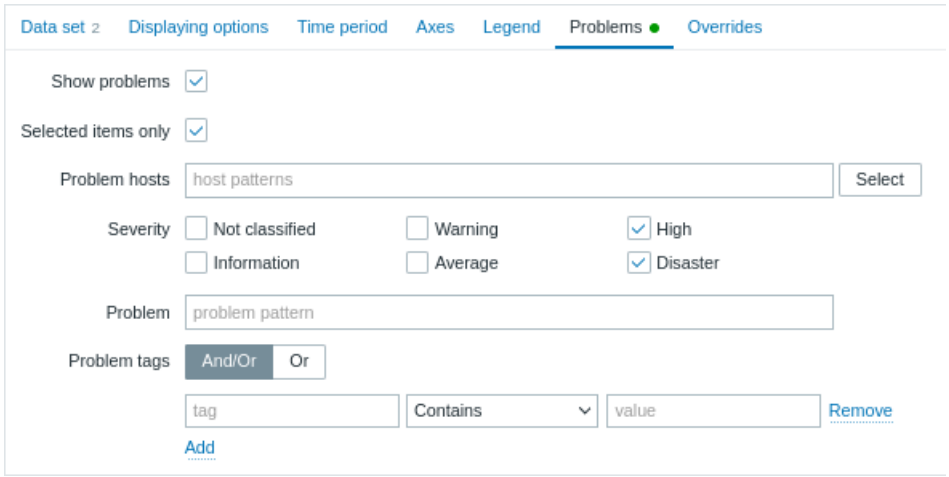
Вкладка **Условные обозначения** позволяет настроить легенду графика:



Показывать легенду	Снимите этот флажок, чтобы скрыть легенду на графике (по умолчанию установлен).
Отображать мин/ср/макс	Установите этот флажок, чтобы отображать в легенде минимальные, средние и максимальные значения элемента данных.
Показывать функцию агрегации	Установите этот флажок, чтобы показывать в легенде функцию агрегации.
Строки	Выберите режим отображения строк легенды: **Фиксированное** — количество отображаемых строк определяется значением параметра *Количество строк*; **Переменное** — количество отображаемых строк определяется числом настроенных элементов данных, но не превышает значение параметра *Максимальное количество строк*.
Количество строк/ Максимальное количество строк	Если для параметра *Строки* установлено значение ”Фиксированное”, задайте количество отображаемых строк легенды (1-10). Если для параметра *Строки* установлено значение ”Переменное”, задайте максимальное количество отображаемых строк легенды (1-10).
Количество столбцов	Задайте количество отображаемых столбцов легенды (1-4). Этот параметр доступен, если флажок *Отображать мин/ср/макс* снят.

Проблемы

Вкладка **Проблемы** позволяет настроить отображение проблем:

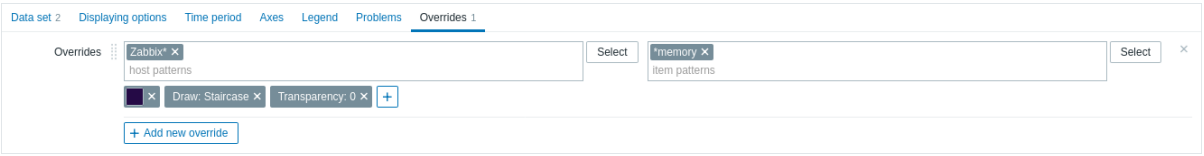


Показывать проблемы	Установите этот флажок, чтобы включить отображение проблем на графике (по умолчанию флажок снят, то есть функция отключена).
Только выбранные элементы данных	Установите этот флажок, чтобы на графике отображались только проблемы для выбранных элементов данных.

Узлы сети проблем	Выберите узлы сети с проблемами, которые должны отображаться на графике. Можно использовать шаблоны с подстановочными знаками (например, * вернет результаты, соответствующие нулю или более символам). Чтобы указать шаблон с подстановочными знаками, просто введите строку вручную и нажмите *Enter*. Во время ввода обратите внимание, что все соответствующие узлы сети отображаются в выпадающем списке. Этот параметр недоступен при настройке виджета на панели шаблона.
Важность	Отметьте уровни важности проблем, чтобы отфильтровать проблемы, отображаемые на графике. Если не отмечен ни один уровень важности, будут отображаться все проблемы.
Проблема	Укажите имя проблемы, которое должно отображаться на графике.
Теги проблем	Укажите теги проблем, чтобы ограничить количество проблем, отображаемых 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 несколько операторов: **Существует** — включить указанные имена тегов; **Равно** — включить указанные имена тегов и значения (с учетом регистра); **Содержит** — включить указанные имена тегов, где значения тегов 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, где значения тегов содержат введенную строку (поиск подстроки, без учета 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.

Переопределения

Вкладка **Переопределения** позволяет добавлять пользовательские переопределения для наборов данных:



Переопределения полезны, когда для набора данных выбрано несколько элементов данных с использованием подстановочного знака *, и вы хотите изменить способ их отображения по умолчанию (например, базовый цвет по умолчанию или любое другое свойство).

Существующие переопределения (если они есть) отображаются в виде списка. Чтобы добавить новое переопределение:

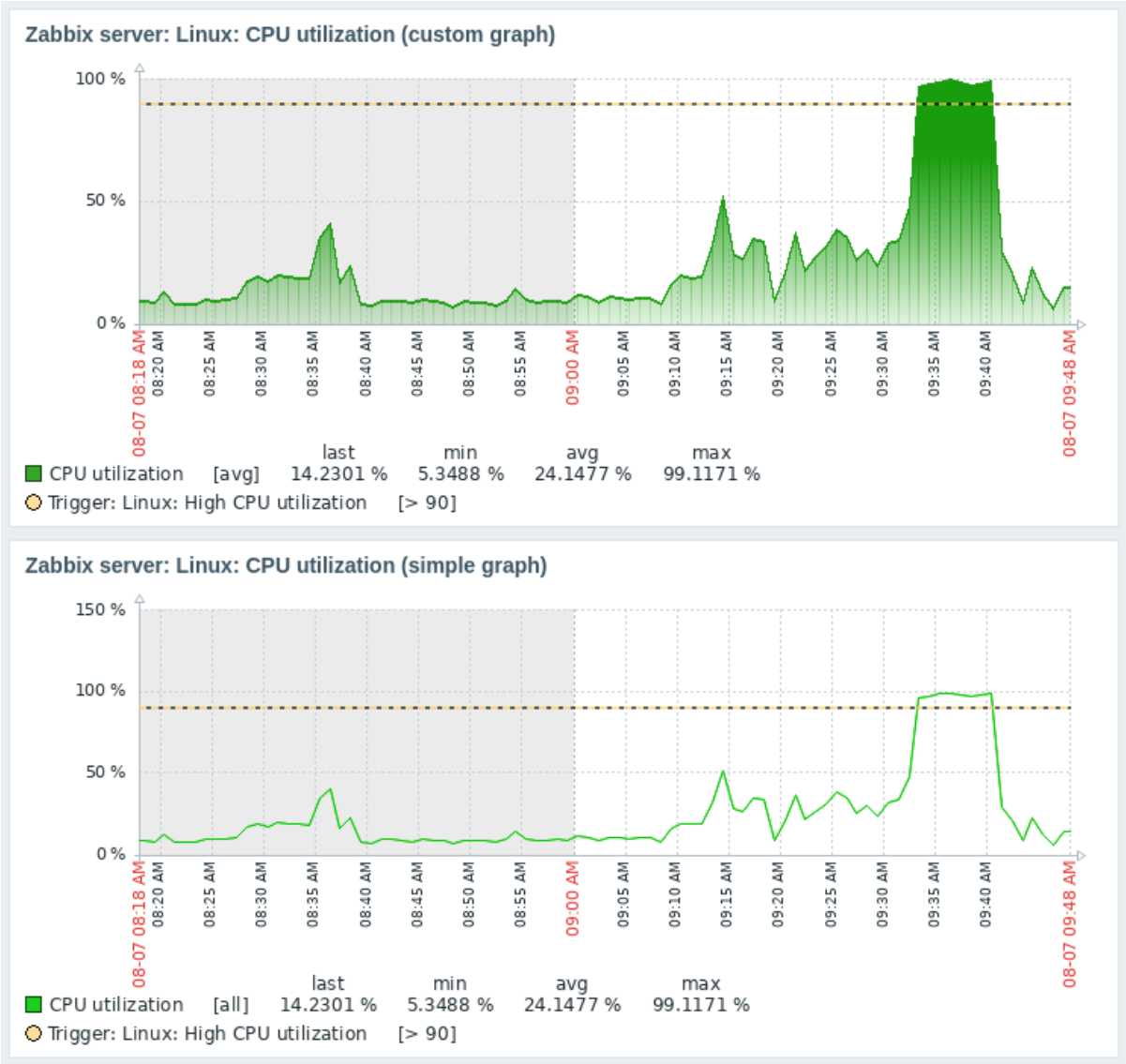
- Нажмите кнопку 

- Выберите узлы сети и элементы данных для переопределения. Либо вы можете ввести шаблоны узлов сети и элементов данных. Можно использовать шаблоны с подстановочными знаками (например, * вернет результаты, соответствующие нулю или более символам). Чтобы указать шаблон с подстановочным знаком, просто 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, например, отдельно добавить элемент данных с именем "item*", если существуют другие подходящие 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 шаблона. Параметр для указания списка элементов данных позволяет выбирать только элементы данных, настроенные в шаблоне при настройке виджета на панели шаблона.
- Нажмите  , чтобы выбрать параметры переопределения. Должен быть выбран как минимум один параметр переопределения. Описание параметров см. на вкладке *Набор данных* выше.

9 График (классический)

Обзор

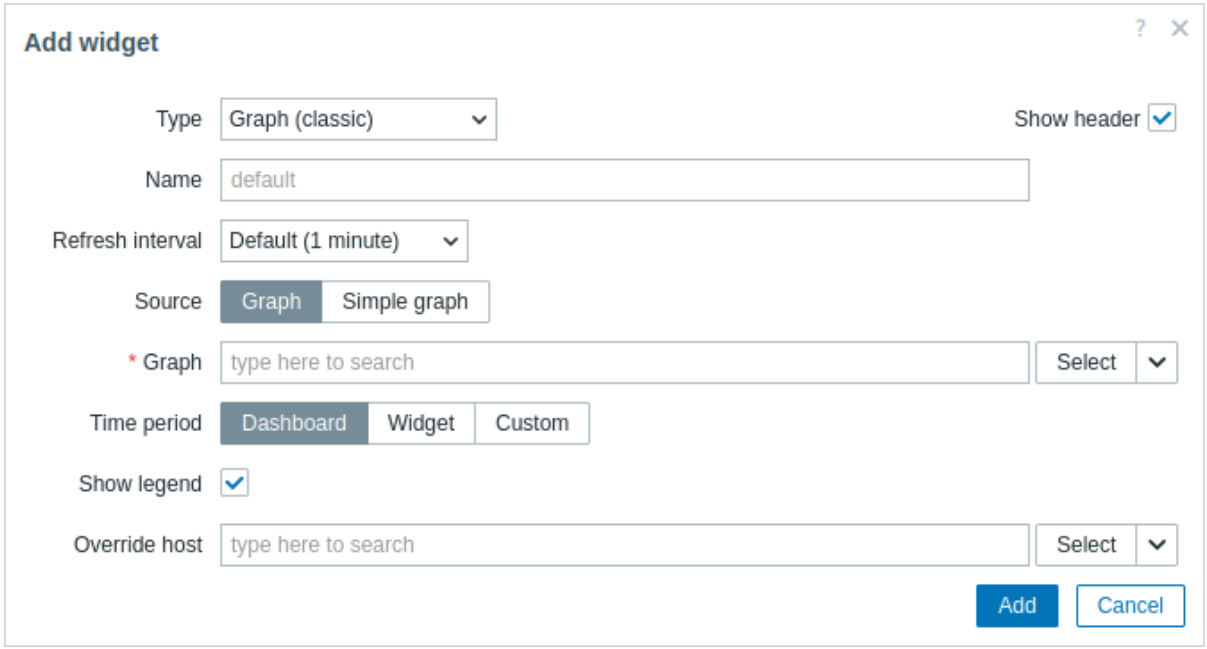
Виджет *Graph (classic)* отображает числовые данные элементов данных в виде изображения пользовательского графика или простого графика. Он поддерживает выбор периода времени, линии триггеров и необязательное отображение легенды.



Информацию, отображаемую в виджете *Graph (classic)*, можно загрузить как изображение PNG, выбрав параметр *Download image* в меню виджета.

Конфигурация

Для настройки выберите *Graph (classic)* в качестве типа:



Помимо параметров, которые являются общими для всех виджетов, вы можете задать следующие специальные параметры:

Source	Выберите тип графика: **Graph** — пользовательский график; **Simple graph** — простой график.
Graph	Выберите пользовательский график для отображения. Либо выберите совместимый виджет в качестве источника данных для графиков. Этот параметр доступен, если *Source* установлено в "Graph".
Item	Выберите элемент данных для отображения на простом графике. Либо выберите совместимый виджет в качестве источника данных для элементов данных. Этот параметр доступен, если *Source* установлено в "Simple graph".
Time period	Задайте период времени, за который будут отображаться данные на графике. Выберите источник данных для периода времени: **Dashboard** — использовать селектор периода времени панели; **Widget** — использовать совместимый виджет (задается в параметре *Widget*); **Custom** — использовать пользовательский период времени, заданный в параметрах *From* и *To*; если задан, в правом верхнем углу виджета будет отображаться значок часов, показывающий установленное время при наведении курсора. Обратите внимание, что независимо от настройки *Time period* виджета, совместимые виджеты все равно 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 *Time period* установлено в "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 *Time period* установлено в "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 *Time period* установлено в "Custom".
Show legend	Снимите этот флажок, чтобы скрыть легенду на графике (по умолчанию флажок 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 панели шаблона.

Легенда графика состоит из трех разделов:

- Элементы данных и их агрегированные значения
- Процентили (если настроены)
- Триггеры (если они связаны с отображаемыми элементами данных)

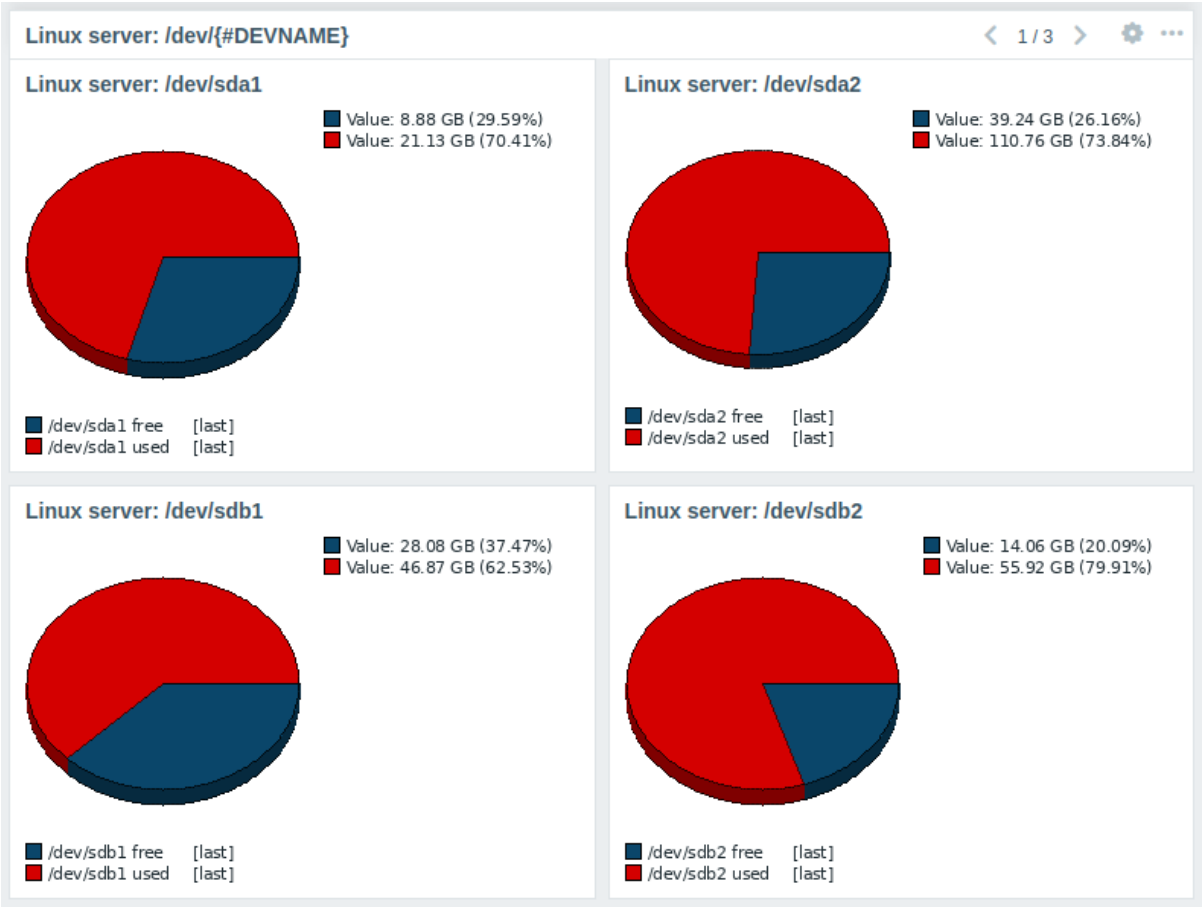
Если высота графика в виджете недостаточна, легенда может не отображаться или отображаться только частично. Сначала скрываются триггеры и процентили, затем — легенда элементов данных. Чтобы отобразить легенду полностью, увеличьте вертикальный размер виджета.

Note:
Можно отобразить не более 3 линий триггеров. Если триггеров больше, для отображения в первую очередь выбираются триггеры с более низкой важностью.

10 Прототип графика

Обзор

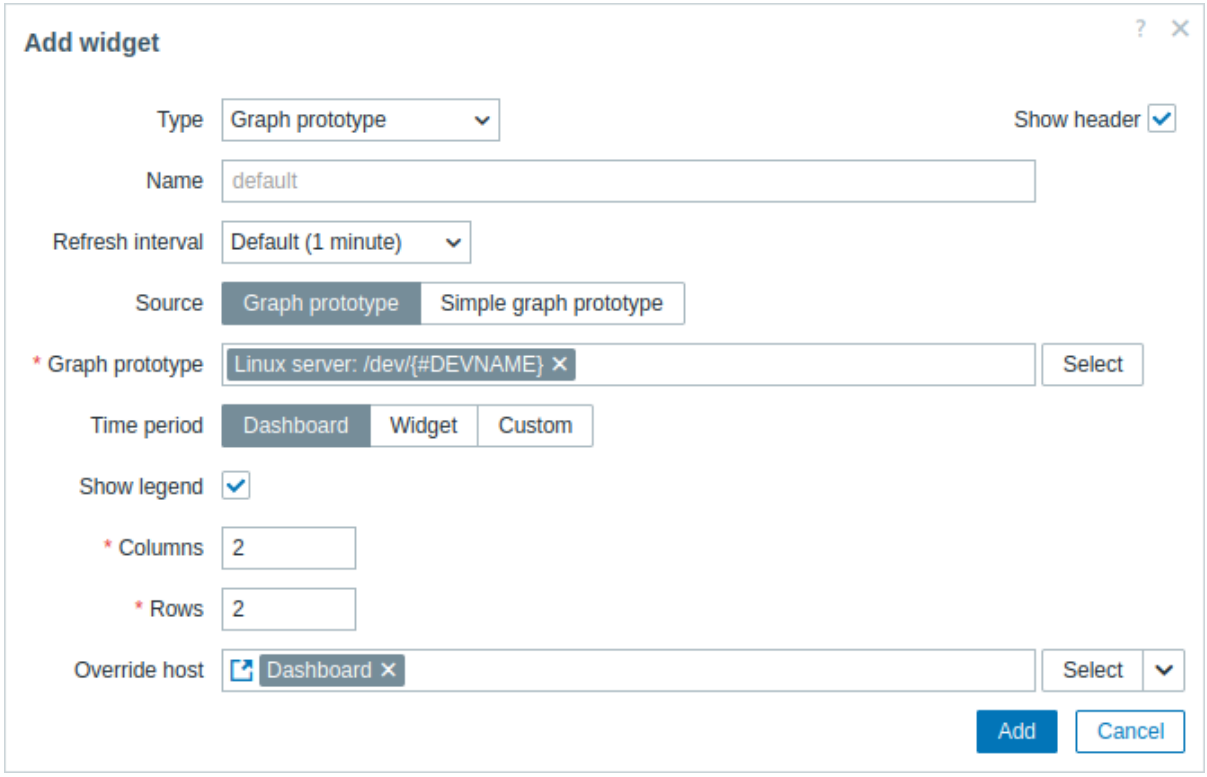
Виджет *Прототип графика* отображает сетку автоматически созданных пользовательских графиков на основе прототипов графиков или прототипов элементов данных правила низкоуровневого обнаружения.



Графики располагаются в настраиваемой сетке; если количество графиков превышает размер сетки, в виджете включается постраничная навигация.

Конфигурация

Чтобы настроить виджет, выберите *Graph prototype* в качестве типа виджета:



Помимо параметров, общих для всех виджетов, вы можете задать следующие специальные параметры:

Source	Выберите источник графиков: **Graph prototype** или **Simple graph prototype**.
Graph prototype	Выберите прототип графика для отображения графиков, обнаруженных этим прототипом графика. Этот параметр доступен, если для *Source* установлено значение "Graph prototype".
Item prototype	Выберите прототип элемента данных для отображения простых графиков для элементов данных, обнаруженных этим прототипом элемента данных. Этот параметр доступен, если для *Source* установлено значение "Simple graph prototype".
Time period	Задайте период времени, за который будут отображаться данные на графиках. Выберите источник данных для периода времени: **Dashboard** — использовать селектор периода времени панели; **Widget** — использовать совместимый виджет (задается 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 курсора. Обратите внимание, что независимо от настройки *Time period* виджета, совместимые виджеты все равно 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* 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 *Time period* 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 *Time period* установлено значение "Custom".
Show legend	Снимите этот флажок, чтобы скрыть легенду на графиках (по умолчанию флажок установлен).
Override host	Выберите совместимый виджет или селектор узла сети панели как источник данных для узлов сети. Этот параметр недоступен при настройке виджета на панели шаблона.
Columns	Введите количество столбцов графиков, отображаемых в виджете прототипа графика.
Rows	Введите количество строк графиков, отображаемых в виджете прототипа графика.

Легенда графика состоит из трех разделов:

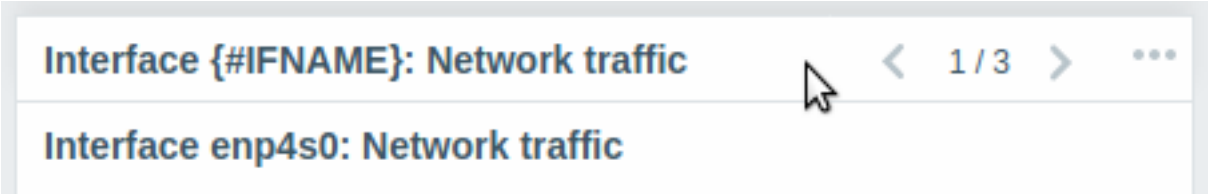
- Элементы данных и их агрегированные значения

- Процентили (если настроены)
- Триггеры (если они 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 отобразить не более 3 линий триггеров. Если триггеров больше, для отображения в приоритетном порядке выбираются триггеры с более низкой важностью.

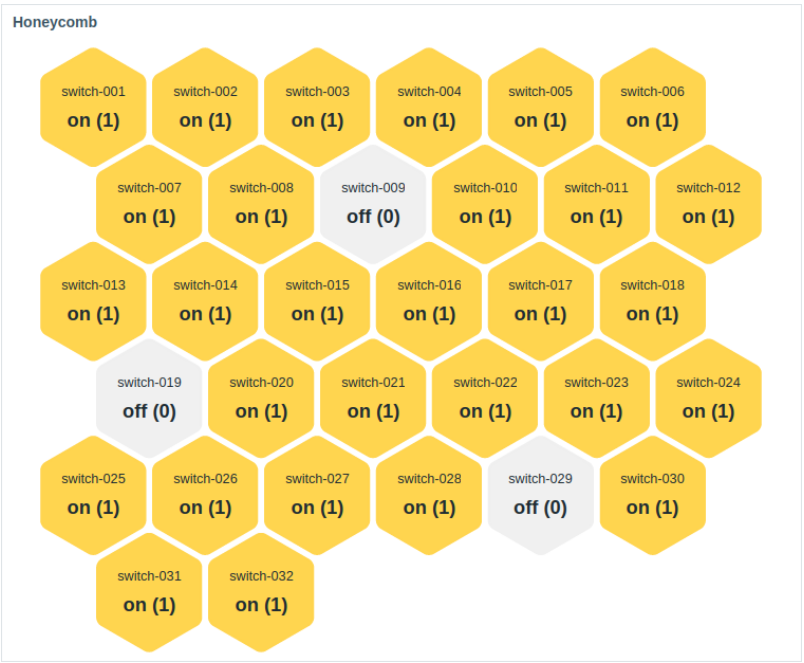
Хотя параметры *Columns* и *Rows* позволяют разместить в виджете более одного графика, обнаруженных графиков все равно может быть больше, чем столбцов/строк в виджете. В этом случае в виджете становится доступна постраничная навигация, а выдвигающийся заголовок позволяет переключаться между страницами с помощью стрелок влево и вправо:

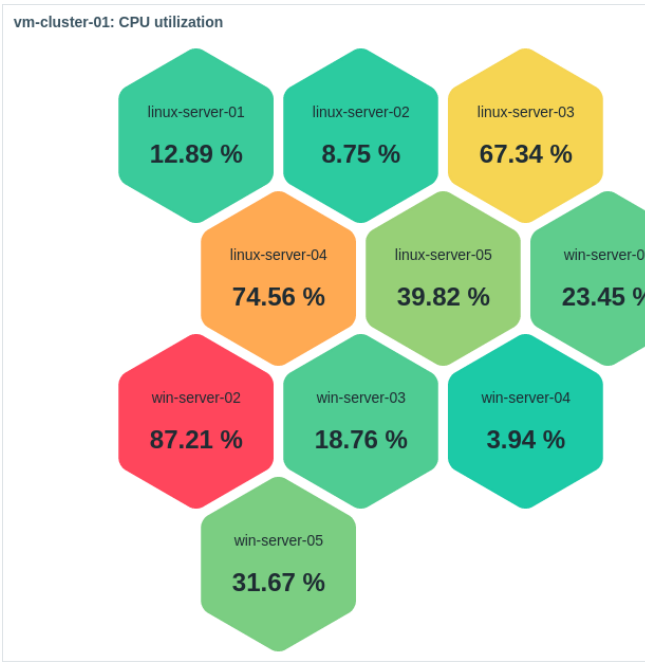


11 Honeycomb

Обзор

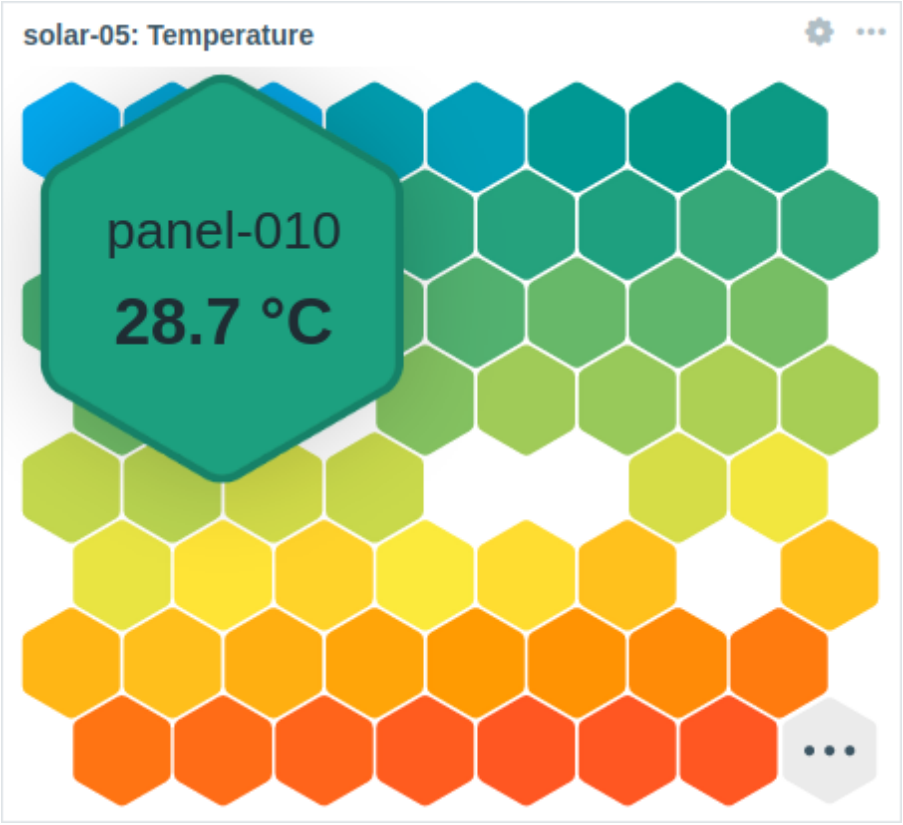
Виджет *Honeycomb* отображает узлы сети или элементы данных в виде шестиугольных ячеек в сетке. Такое расположение позволяет легко выявлять проблемные узлы сети/элементы данных, замечать группы проблем, быстро сравнивать группы и многое другое.





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 ячеек, их размер и расположение динамически подстраиваются.

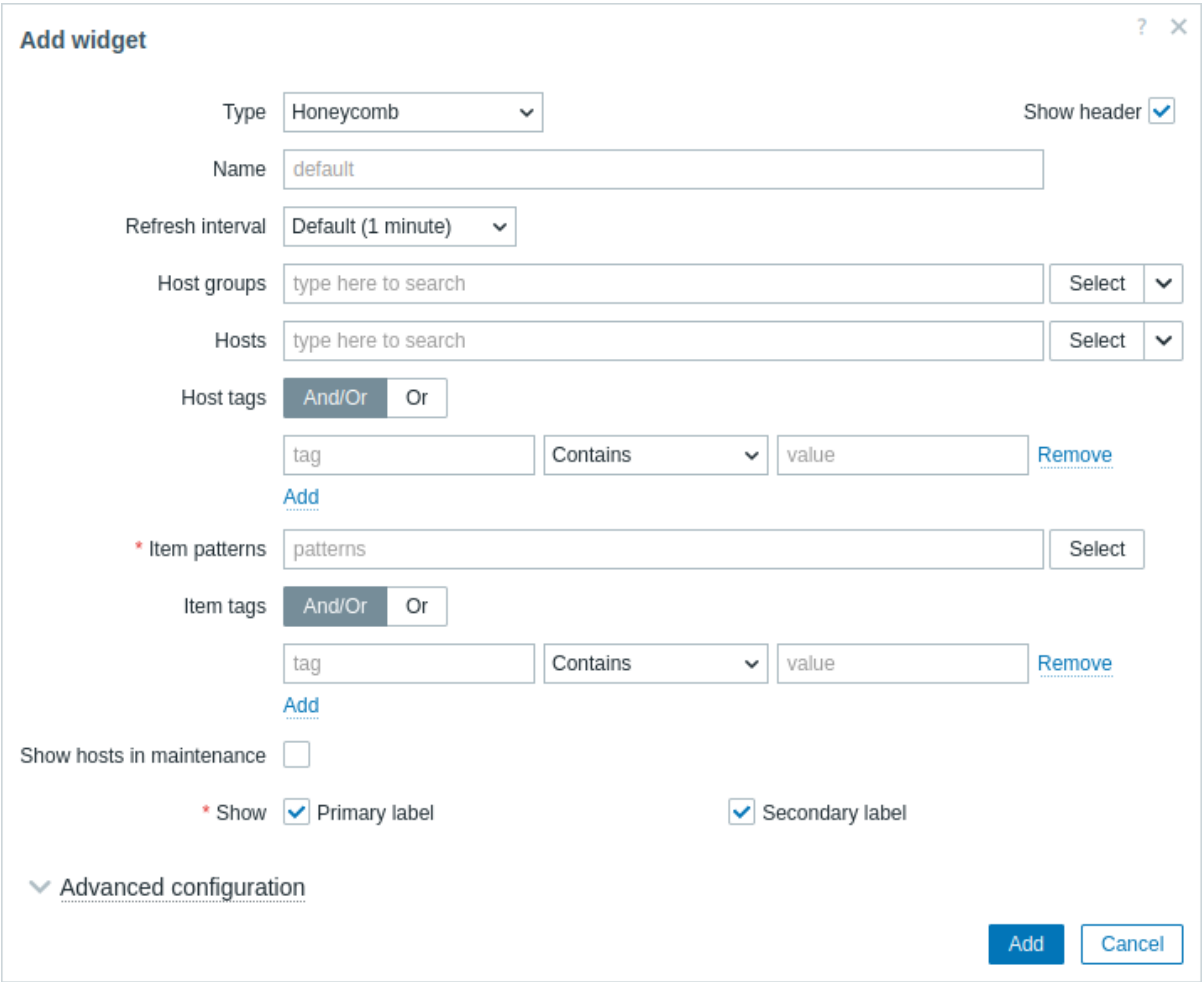
При наведении курсора выделенная ячейка увеличивается для лучшей видимости. При щелчке по ячейке ее граница подсвечивается, пока не будет выбрана другая ячейка.



Информацию, отображаемую в виджете *Honeycomb*, можно скачать в виде изображения PNG, выбрав параметр *Download image* в меню виджета.

Конфигурация

Чтобы настроить виджет, выберите тип *Honeycomb*:



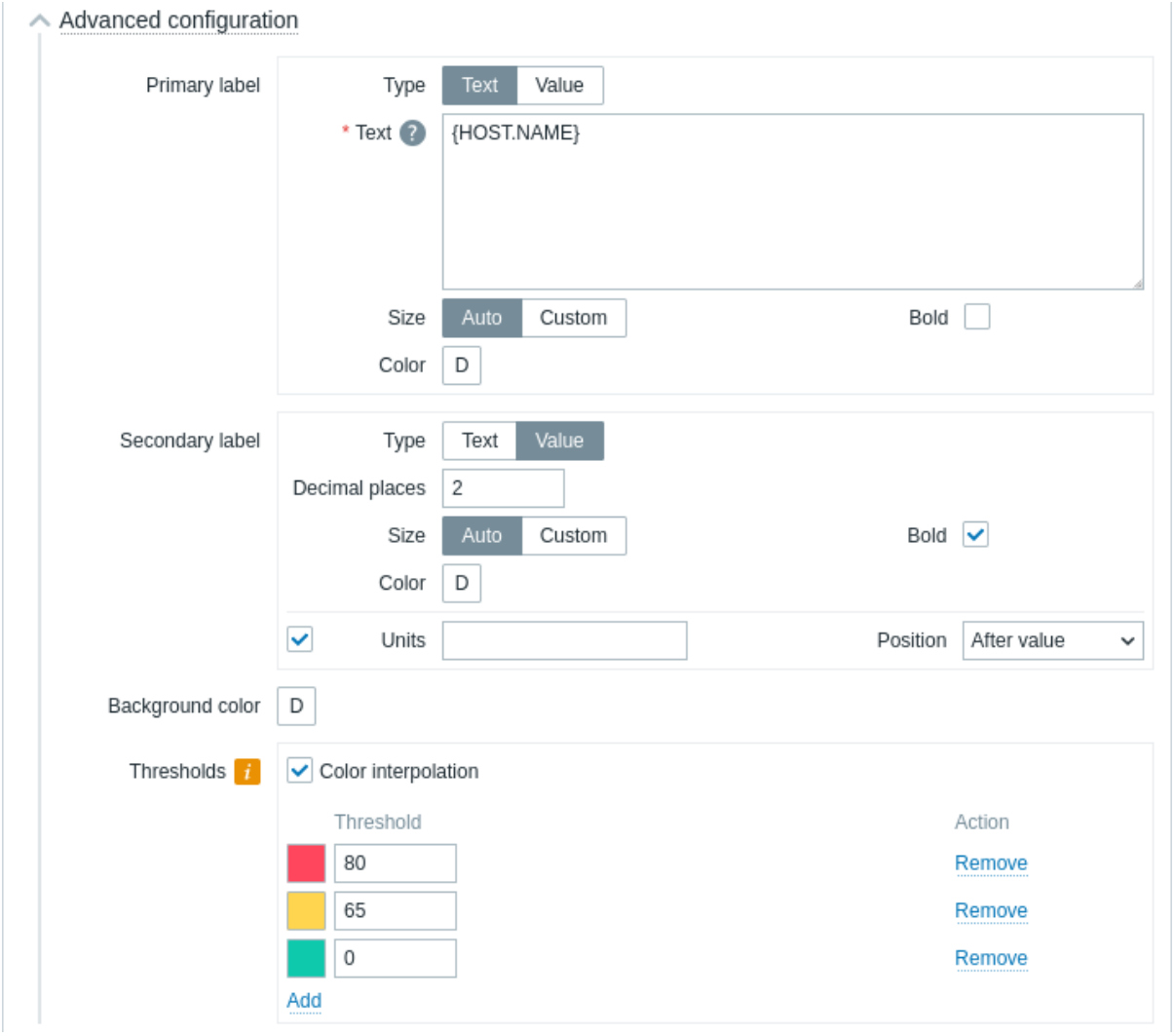
В дополнение к параметрам, общим для всех виджетов, вы можете задать следующие специальные параметры:

Группы узлов сети	Выберите группы узлов сети. В качестве источника данных для групп узлов сети также можно выбрать совместимый виджет. Это поле поддерживает автодополнение, поэтому при вводе имени группы будет предложен выпадающий список подходящих групп. При выборе родительской группы узлов сети автоматически выбираются все вложенные группы узлов сети; если не выбрана ни одна группа узлов сети, 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 панели шаблона.
Узлы сети	Выберите узлы сети. В качестве источника данных для узлов сети также можно выбрать совместимый виджет или панель. Это поле поддерживает автодополнение, поэтому при вводе имени узла сети будет предложен выпадающий список подходящих узлов сети. Если не выбран ни один узел сети, виджет отобразит все узлы сети с элементами данных, соответствующими выбранному *Шаблону элемента данных* (см. ниже). Этот параметр недоступен при настройке виджета на панели шаблона.
Теги узлов сети	Укажите теги, чтобы ограничить количество узлов сети, 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 и значения тегов (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 и значения тегов (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. Этот параметр недоступен при настройке виджета на панели шаблона.
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 введённым или выбранным шаблонам. Параметр *Шаблоны элементов данных* является обязательным. Для выбора можно использовать шаблоны с подстановочными знаками (например, `*` вернёт элементы данных, соответствующие нулю или более символам; `Zabbix*` вернёт элементы данных, начинающиеся с ”Zabbix”). Чтобы указать шаблон с подстановочным знаком, введите строку вручную и нажмите *Enter*. Когда вы начнёте ввод, в выпадающем списке будут показаны подходящие элементы данных, ограниченные теми, которые принадлежат выбранным *Узлам сети* или узлам сети в выбранных *Группах узлов сети*, если они указаны. Символ подстановки интерпретируется всегда, поэтому невозможно, например, добавить отдельно элемент данных с именем *item**, если существуют другие подходящие элементы данных (например, item2, item3). При настройке виджета на панели шаблона этот параметр позволяет выбирать только элементы данных, настроенные в шаблоне.
Теги элементов данных	Укажите теги, чтобы ограничить количество элементов данных, отображаемых в виджете. Дополнительную информацию см. в описании *Тегов узлов сети* выше.

Показывать узлы сети на обслуживании	Установите этот флажок, чтобы отображать узлы сети, находящиеся на обслуживании (в этом случае рядом с именем узла сети будет показан значок обслуживания). При настройке виджета на панели шаблона этот параметр называется *Показывать данные на обслуживании*.
Показывать	Установите этот флажок, чтобы отображать соответствующий элемент ячейки honeycomb — основную метку, вторичную метку. Должен быть выбран как минимум один элемент.
Расширенная конфигурация	Щёлкните по метке *Расширенная конфигурация*, чтобы отобразить параметры расширенной конфигурации.

Расширенная настройка

Параметры расширенной настройки доступны в сворачиваемом разделе *Расширенная настройка* и только для элементов данных, выбранных в поле *Показывать* (см. выше), а также для цвета фона или порогов для ячеек сот.



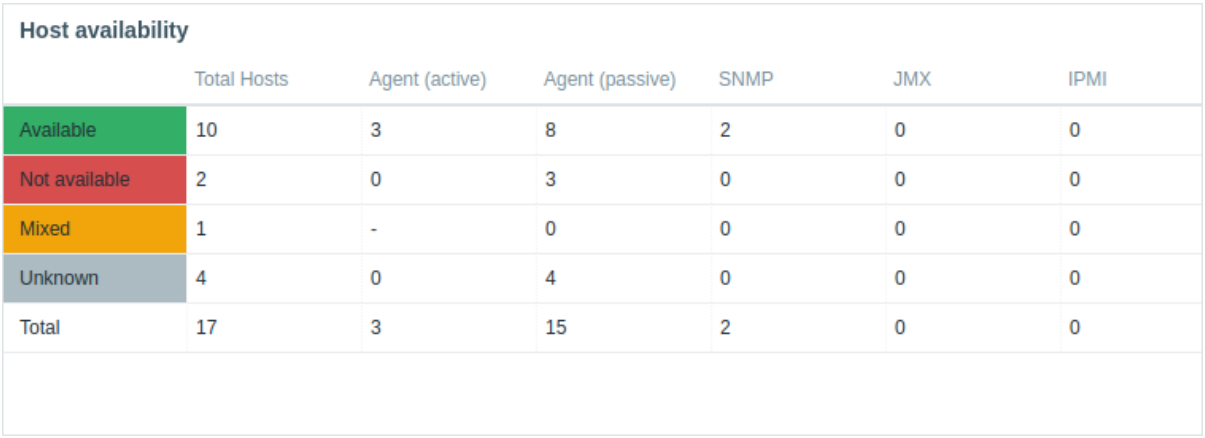
Основная/вторичная метка	
Тип	Выберите тип метки: **Текст** — метка будет отображать текст, указанный в параметре *Текст*; **Значение** — метка будет отображать значение элемента данных с количеством десятичных знаков, указанным в параметре *Десятичные знаки*.

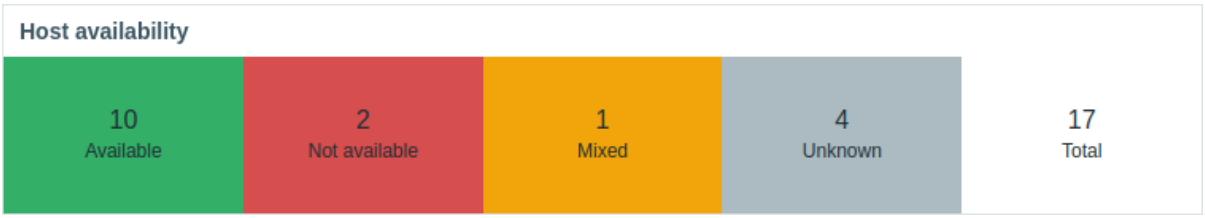
Текст	Введите текст метки. Этот текст может переопределить имя элемента данных по умолчанию. Поддерживается многострочный текст. Возможно сочетание текста и поддерживаемых макросов. Поддерживаются {HOST.*}, {ITEM.*}, {INVENTORY.*} и пользовательские макросы. Ячейки сот упорядочиваются в алфавитном порядке по имени узла сети, а внутри каждого узла сети — по имени элемента данных. Этот параметр доступен, если для *Тип* установлено значение ”Текст”.
Десятичные знаки	Введите количество десятичных знаков, отображаемых со значением. Этот параметр доступен, если для *Тип* установлено значение ”Значение”, и влияет только на элементы данных, возвращающие данные числового типа с плавающей точкой.
Размер	Выберите размер метки: **Авто** — использовать автоматически подбираемый размер метки; **Пользовательский** — введите пользовательский размер метки (в процентах относительно размера ячейки сот). Обратите внимание, что метки, не помещающиеся в размер ячейки сот, обрезаются.
Жирный	Установите флажок, чтобы отображать единицы измерения элемента данных жирным шрифтом.
Цвет	Выберите цвет единиц измерения элемента данных в палитре цветов. ”D” обозначает цвет по умолчанию, который зависит от темы веб-интерфейса. Чтобы вернуть цвет по умолчанию, нажмите кнопку *Использовать по умолчанию* 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 переопределит единицы измерения, заданные в настройке элемента данных. Этот параметр доступен, если для *Тип* установлено значение ”Текст”.
Положение	Выберите положение единиц измерения элемента данных (перед или после значения элемента данных). Этот параметр игнорируется для следующих единиц измерения, связанных со временем: unixtime, uptime, s. Этот параметр доступен, если для *Тип* установлено значение ”Текст”.
Цвет фона	
Цвет фона	Выберите цвет фона ячеек сот в палитре цветов. ”D” обозначает цвет по умолчанию, который зависит от темы веб-интерфейса. Чтобы вернуть цвет по умолчанию, нажмите кнопку *Использовать по умолчанию* в палитре цветов.
Пороги	
Интерполяция цвета	Установите флажок, чтобы включить плавный переход между цветами порогов для ячеек сот. Этот параметр доступен, если заданы два или более порога.
Порог	Нажмите *Добавить*, чтобы добавить порог, выберите цвет порога в палитре цветов и укажите числовое значение. При сохранении список порогов будет отсортирован по возрастанию. Обратите внимание, что цвета, настроенные как пороги, будут корректно отображаться только для числовых элементов данных. Поддерживаются суффиксы (например, ”1d”, ”2w”, ”4K”, ”8G”). Поддерживаются преобразования значений.

12 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.



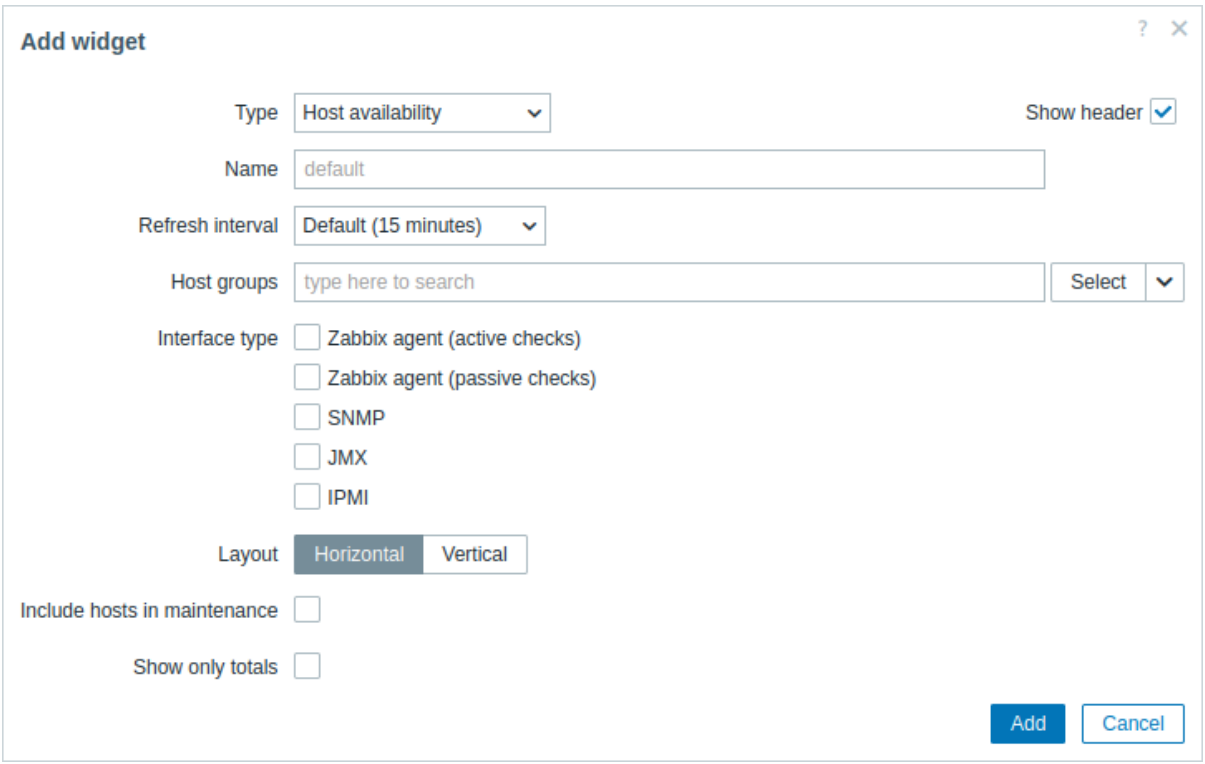


Доступность узлов сети подсчитывается следующим образом:

- *Доступен* — узлы сети, у которых все интерфейсы доступны
- *Недоступен* — узлы сети, у которых все интерфейсы недоступны
- *Смешанный* — узлы сети, у которых как минимум один интерфейс недоступен и как минимум один интерфейс доступен или имеет неизвестный статус
- *Неизвестно* — узлы сети, у которых как минимум один интерфейс имеет неизвестный статус, но нет ни одного недоступного интерфейса
- *Всего* — общее количество всех узлов сети

Настройка

Для настройки выберите *Доступность узла сети* в качестве типа:



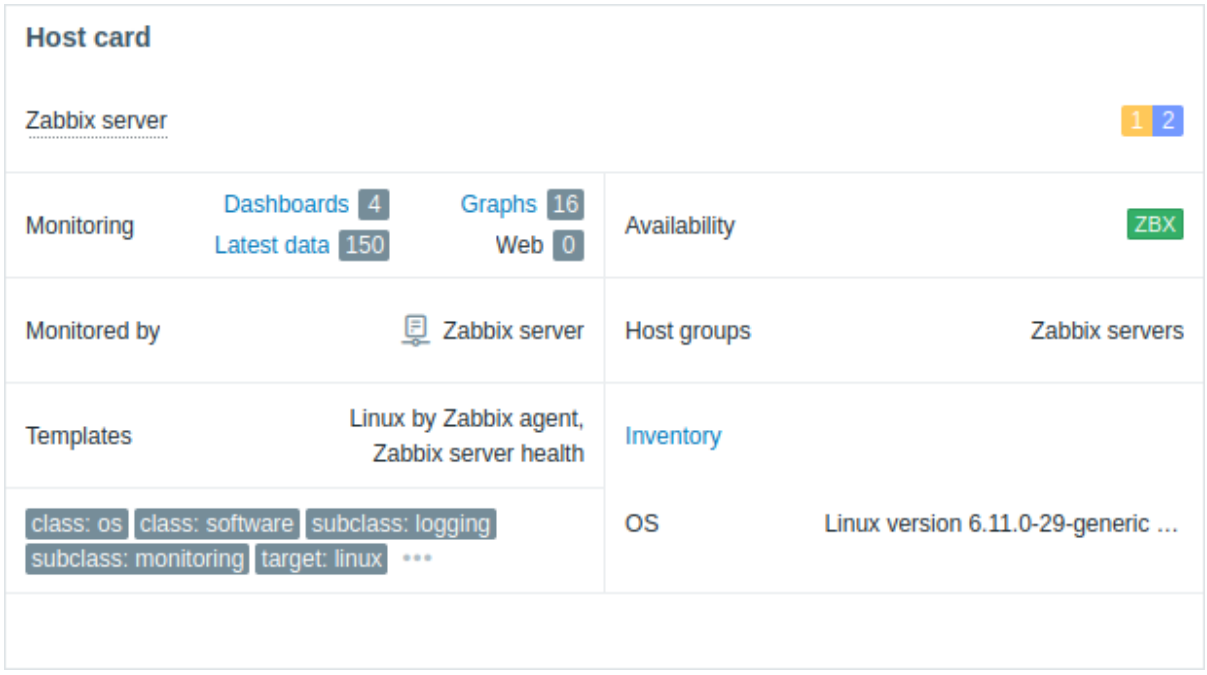
В дополнение к параметрам, которые являются общими для всех виджетов, вы можете задать следующие специальные параметры:

Группы узлов сети	Выберите группы узлов сети. 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 панели шаблона.
Тип интерфейса	Выберите, для каких интерфейсов узла сети вы хотите видеть данные о доступности: *Zabbix агент (активные проверки)*, *Zabbix агент (пассивные проверки)*, *SNMP*, *JMX*, *IPMI*. Для *Zabbix агент (активные проверки)* ячейка *Смешано* всегда будет пустой, так как элементы данных этого типа не могут иметь несколько интерфейсов. По умолчанию, если ничего не выбрано, отображается доступность всех интерфейсов.
Макет	Выберите горизонтальное отображение (столбцы) или вертикальное отображение (строки).
Включать узлы сети в обслуживании	Включать в статистику узлы сети, находящиеся на обслуживании. При настройке виджета на панели шаблона этот параметр называется *Показывать данные в обслуживании*.
Показывать только итоги	Если флажок установлен, отображается общее количество узлов сети без разбивки по интерфейсам. Эта опция отключена, если выбран только один интерфейс.

13 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.).

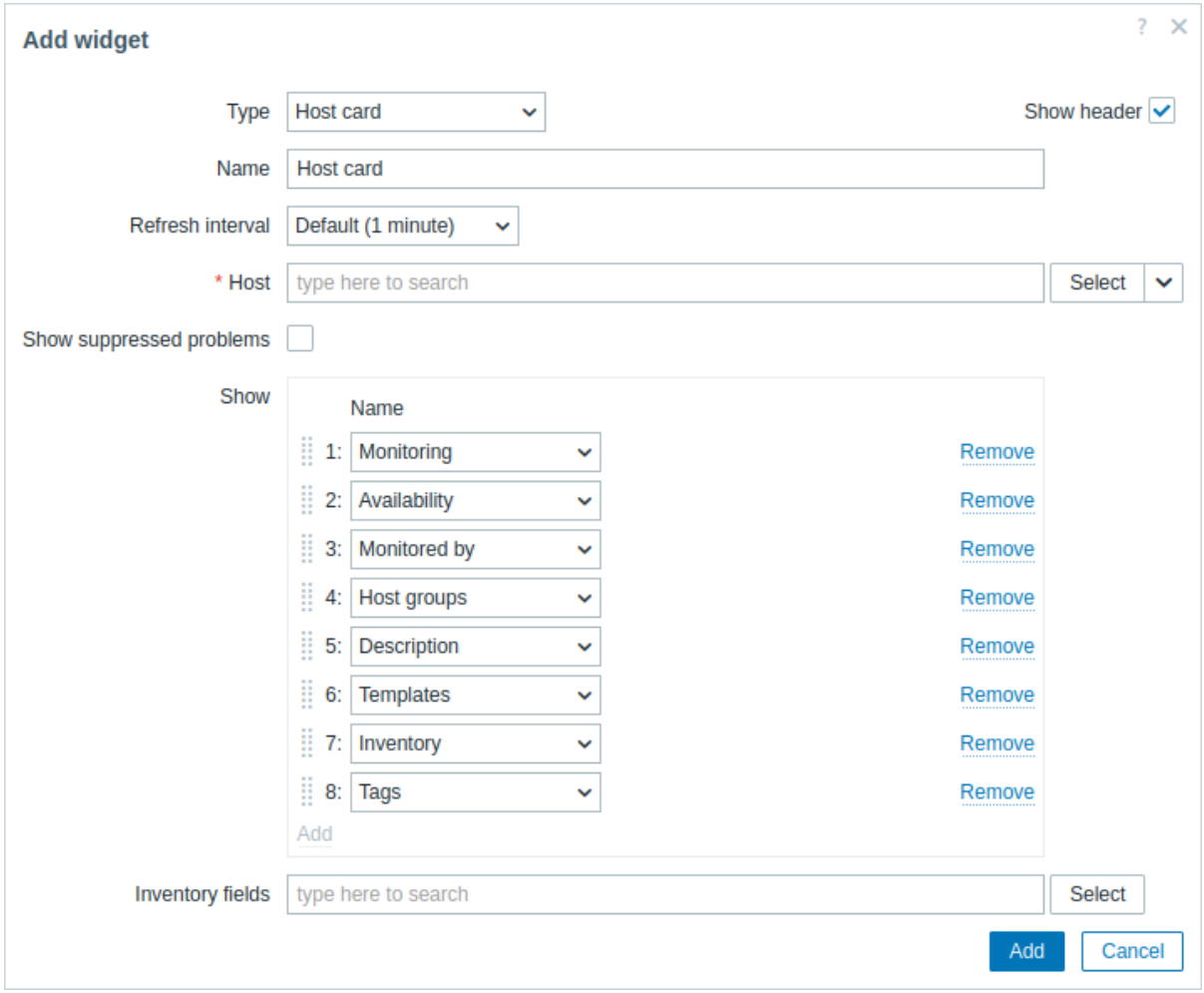


Узел сети можно задать непосредственно в настройках виджета или выбрать либо из совместимого виджета, либо с помощью селектора узла сети на панели мониторинга.

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:

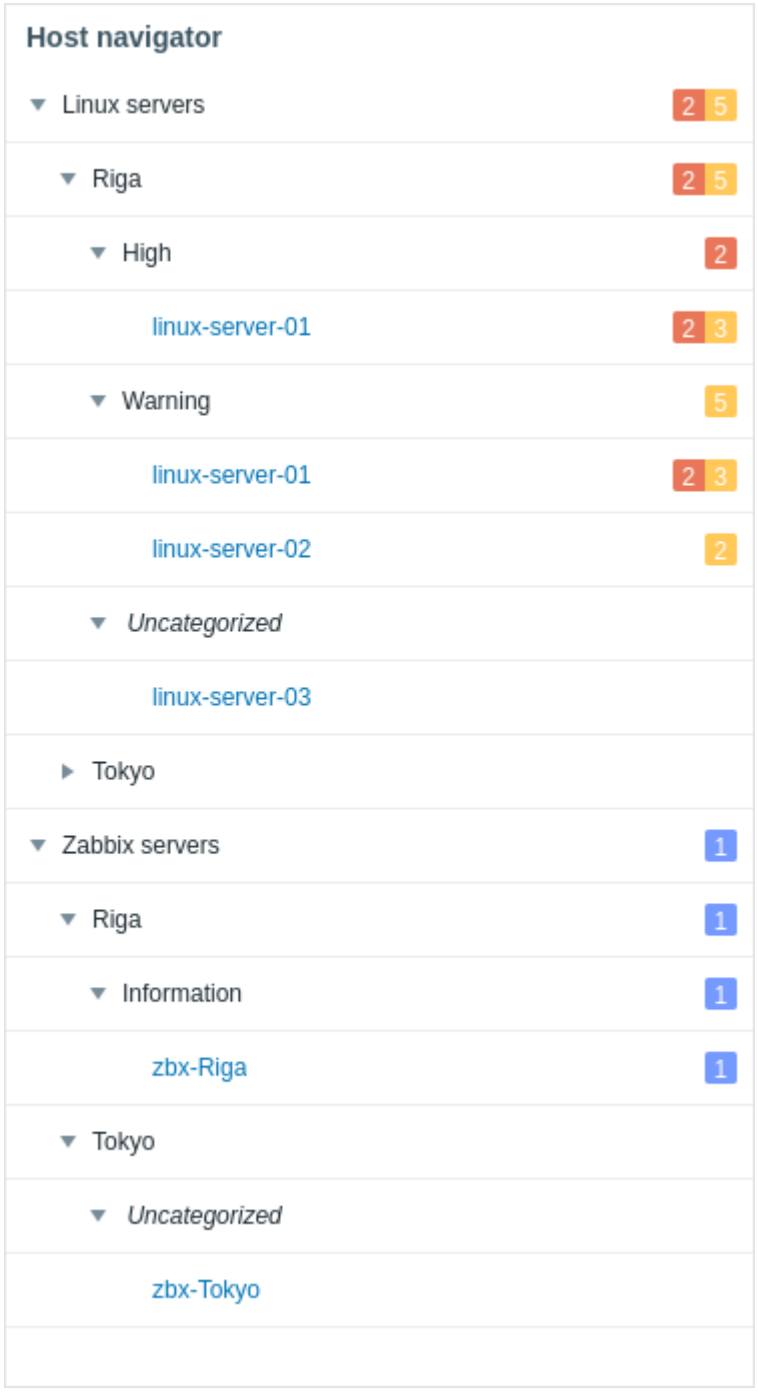
Узел сети	Выберите узел сети. Либо выберите совместимый виджет или панель как источник данных для узлов сети. По мере ввода будут появляться подходящие варианты. Верхняя часть виджета всегда отображает **имя узла сети** (нажатие на него открывает меню узла сети): - Если узел сети отключен, рядом с именем 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*.

14 Навигатор узлов сети

Обзор

Виджет *Навигатор узлов сети* отображает список узлов сети на основе различных параметров фильтрации и группировки.



Группы, по которым организованы узлы сети, можно разворачивать и сворачивать.

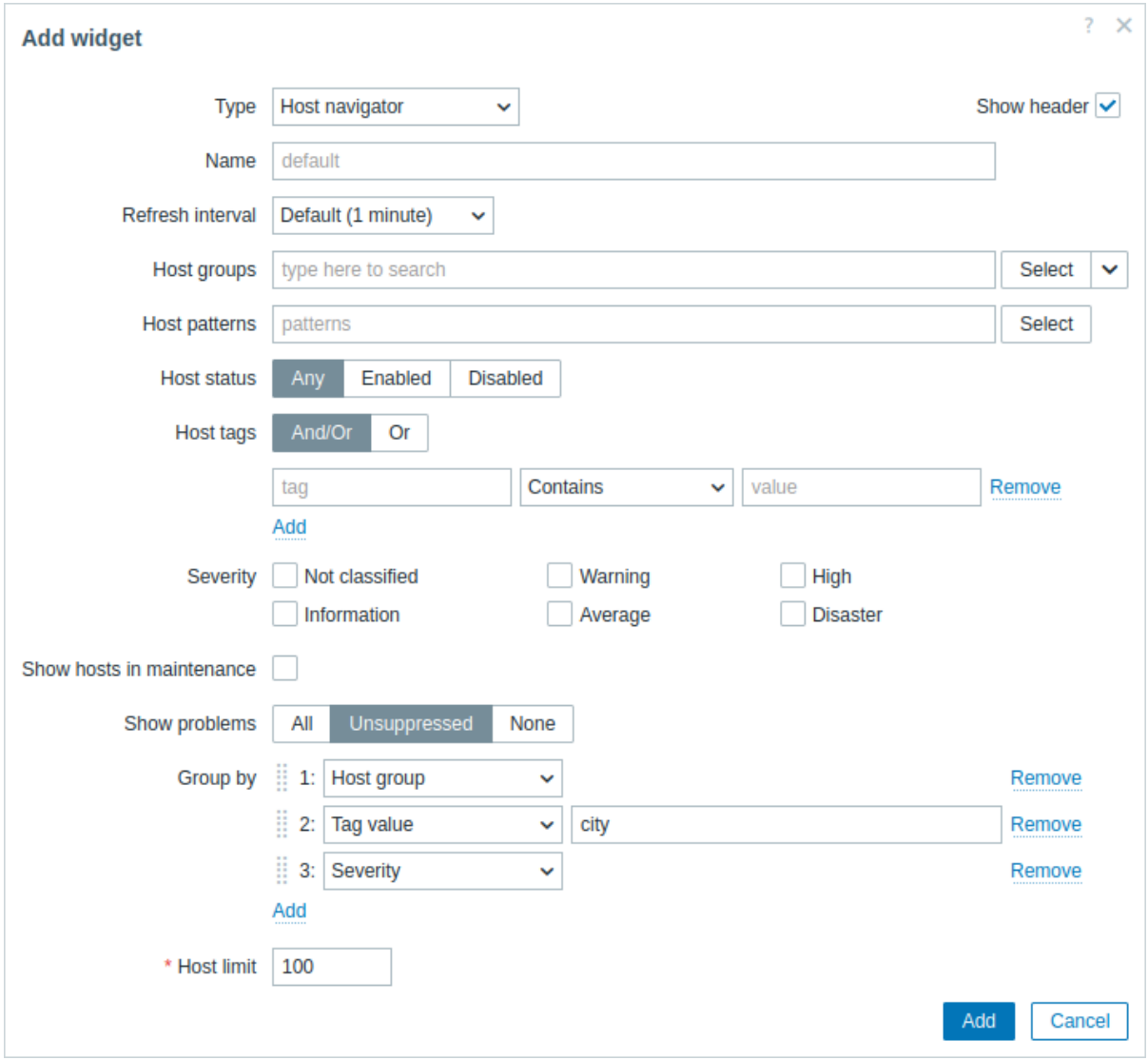
Для групп, проблем и узлов сети, находящихся на обслуживании, дополнительная информация доступна во всплывающих подсказках при наведении курсора мыши.

Этот виджет наиболее полезен для управления тем, что отображают другие виджеты, в зависимости от выбранного узла сети.



Конфигурация

Для настройки выберите *Host navigator* в качестве типа:



Помимо параметров, которые являются общими 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 предложен выпадающий список подходящих групп. При выборе родительской группы узлов сети автоматически выбираются все вложенные группы узлов сети; если не выбрана ни одна группа узлов сети, 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 узлов сети. Узлы сети, соответствующие указанным шаблонам, будут отображаться в навигаторе узлов сети. Это поле поддерживает автодополнение, поэтому при вводе имени узла сети будет предложен выпадающий список подходящих узлов сети. Если не выбран ни один узел сети, 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, имена которых начинаются с "Zabbix"). Чтобы указать шаблон с подстановочным знаком, введите строку вручную и нажмите *Enter*. Когда вы начинаете ввод, выпадающий список покажет подходящие узлы сети, ограниченные теми, которые принадлежат узлам сети в выбранных *Группах узлов сети*, если они указаны. Этот параметр недоступен при настройке виджета на панели шаблона.

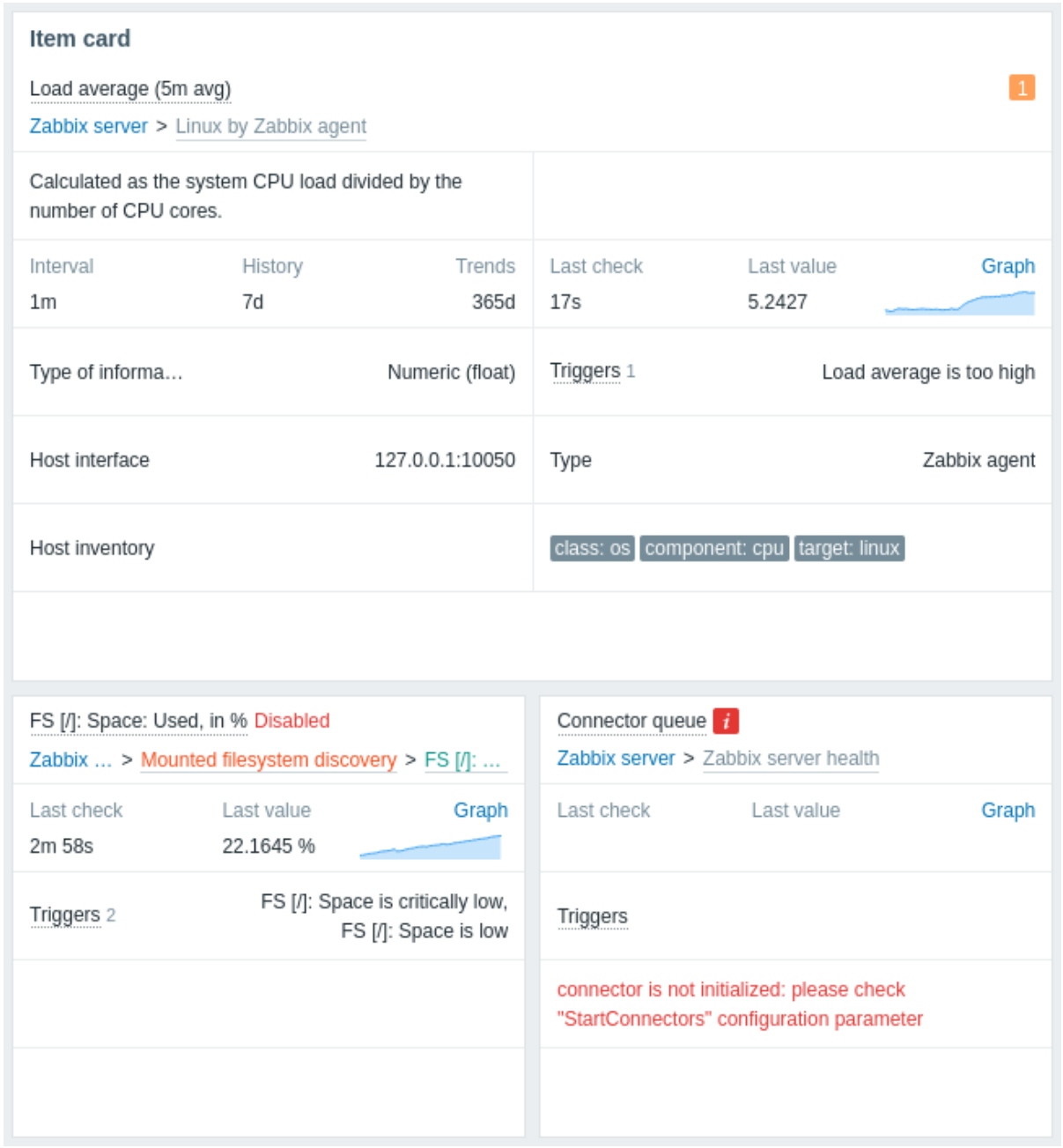
Состояние узла сети	Отфильтруйте, какие узлы сети отображать, по их состоянию (любое, включен, отключен). Этот параметр недоступен при настройке виджета на панели шаблона.
Теги узла сети	Укажите теги для фильтрации узлов сети, отображаемых в виджете. Можно как включать, так и исключать определенные теги и значения тегов. Можно задать несколько условий. Сопоставление имен тегов всегда чувствительно к регистру. Для каждого условия доступны несколько операторов: **Существует** — включить указанные имена тегов; **Равно** — включить указанные имена тегов и значения (с учетом регистра); **Содержит** — включить указанные имена тегов, где значения тегов 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, где значения тегов содержат введенную строку (поиск подстроки, без учета регистра). Для условий доступны два типа вычисления: **И/Или** — должны выполняться все условия, при этом условия с одинаковым именем тега будут сгруппированы по условию 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, которые будут отображаться в виджете. Если не отмечена ни одна важность, будут отображаться все узлы сети со всеми проблемами.
Показывать узлы сети на обслуживании	Установите этот флажок, чтобы отображать узлы сети на обслуживании (в этом случае рядом с именем узла сети будет показан значок обслуживания). Этот параметр имеет метку *Показывать данные на обслуживании* при настройке виджета на панели шаблона.
Показывать проблемы	

Группировать по	Добавьте атрибут группировки, по которому будут сгруппированы выбранные узлы сети: **Группа узлов сети** — группировать узлы сети по их группе узлов сети; **Значение тега** — введите имя тега, чтобы группировать 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 отображаются все количества его проблем. Обратите внимание, что узлы сети будут сгруппированы только по тем важностям, которые отмечены в параметре *Важность*; если не отмечена ни одна важность, все узлы сети будут сгруппированы по всем важностям. Атрибуты группировки можно переупорядочивать, 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: color, 2: city) приведет к тому, что узлы сети будут сначала сгруппированы по color (red, blue и т. д.), а затем по city (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 принадлежит нескольким группам узлов сети). Щелчок по таким узлам сети выбирает и подсвечивает их во всех группах. Узлы сети, которые не соответствуют 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.
Лимит узлов сети	Введите максимальное количество отображаемых узлов сети. Возможные значения: от 1 до 9999. Если для отображения доступно больше узлов сети, чем установленный лимит, под списком отображаемых узлов сети будет показано соответствующее сообщение (например, ”Показано 100 из 100+ узлов сети”). Обратите внимание, что настроенный лимит узлов сети также влияет на отображение настроенных групп; например, если лимит узлов сети установлен в 100, а узлы сети сгруппированы по значениям тегов (более 200), в виджете будут отображены только первые 100 значений тегов с соответствующими узлами сети. На этот параметр не влияет параметр *Limit for search and filter results* в *Администрирование* → *Общие* → *GUI*. Этот параметр недоступен при настройке виджета на панели шаблона.

15 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.).

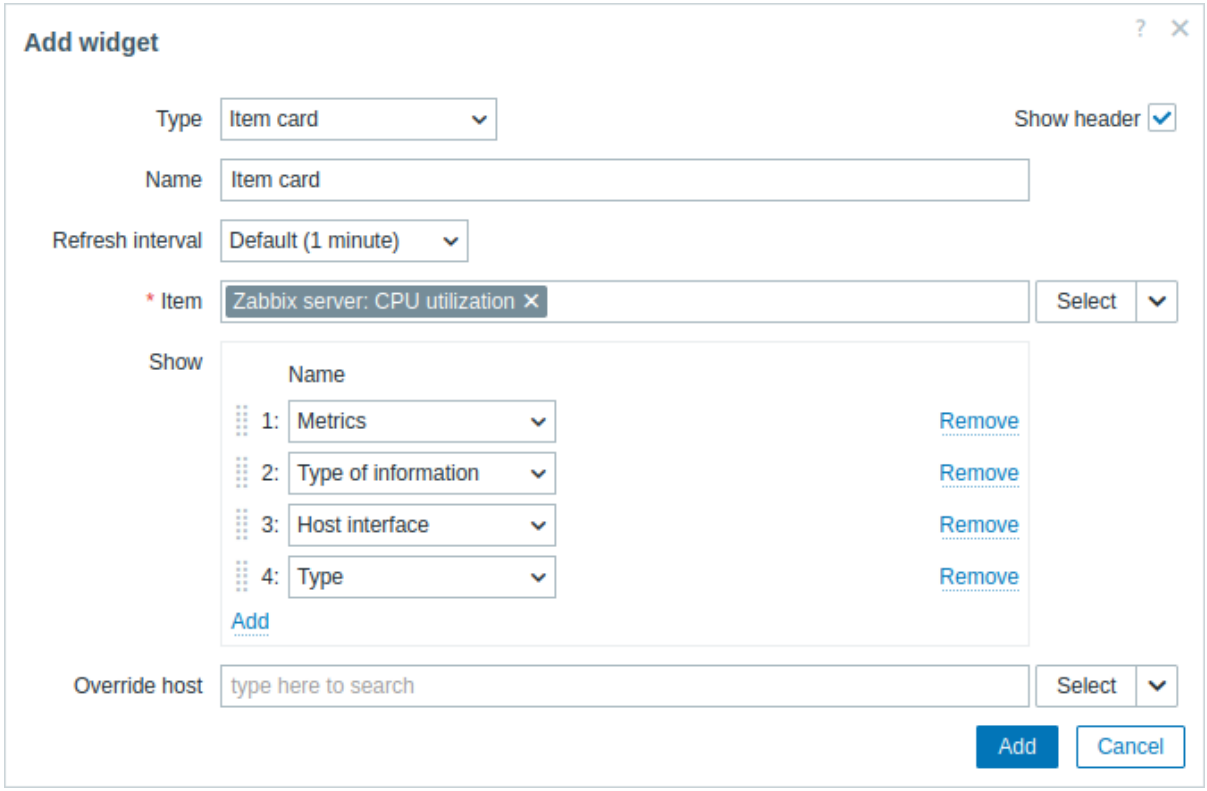


Элемент данных можно задать непосредственно в настройках виджета или выбрать из совместимого виджета на панели мониторинга.

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*:



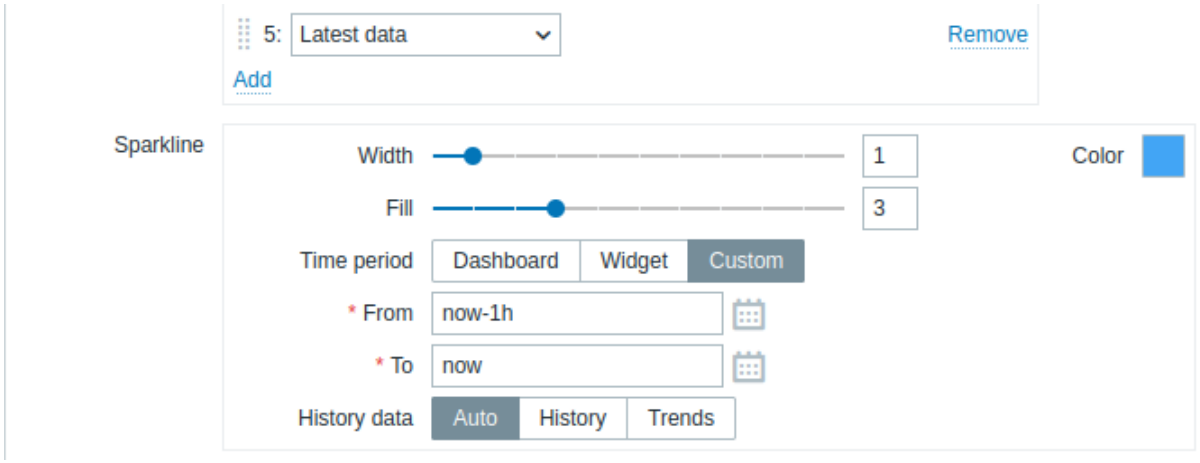
Помимо параметров, общих для всех виджетов, можно задать следующие специальные параметры:

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*.



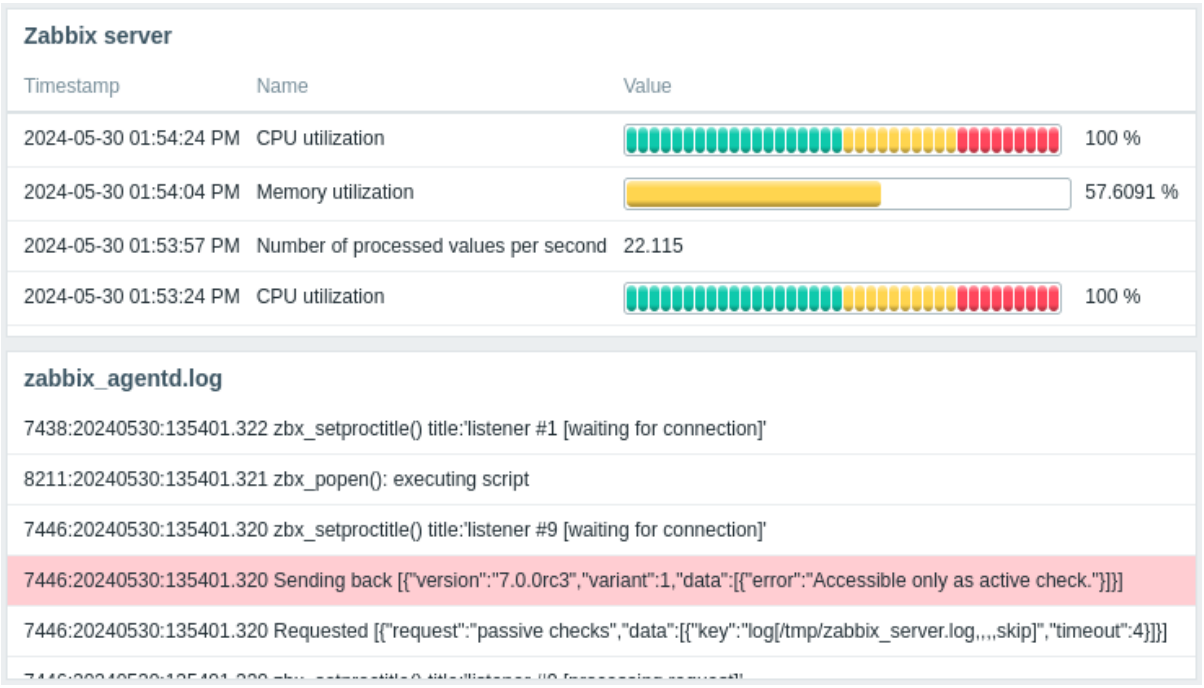
Ширина	Задайте толщину линии графика с помощью ползунка или вручную введя значение в диапазоне от 0 до 10.
Цвет	Выберите цвет линии и заливки.
Заливка	Задайте уровень прозрачности цвета заливки с помощью ползунка или вручную введя значение в диапазоне от 0 до 10.

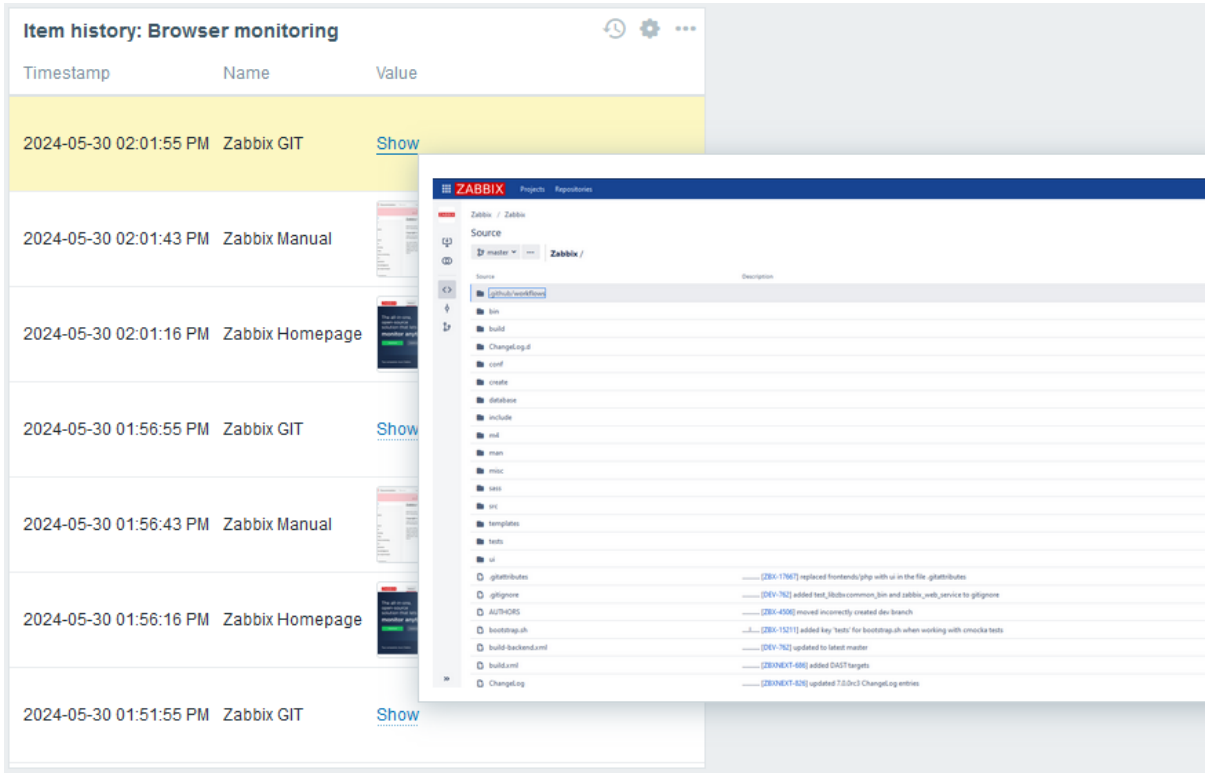
Период времени	Выберите источник данных для периода времени: **Панель** - использовать селектор периода времени панели; **Виджет** - использовать совместимый виджет (задается в параметре *Виджет*); **Пользовательский** - использовать пользовательский период времени, заданный в параметрах *От* и *До*; 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.

16 История элементов данных

Обзор

Виджет *История элементов 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 лог-файлов).

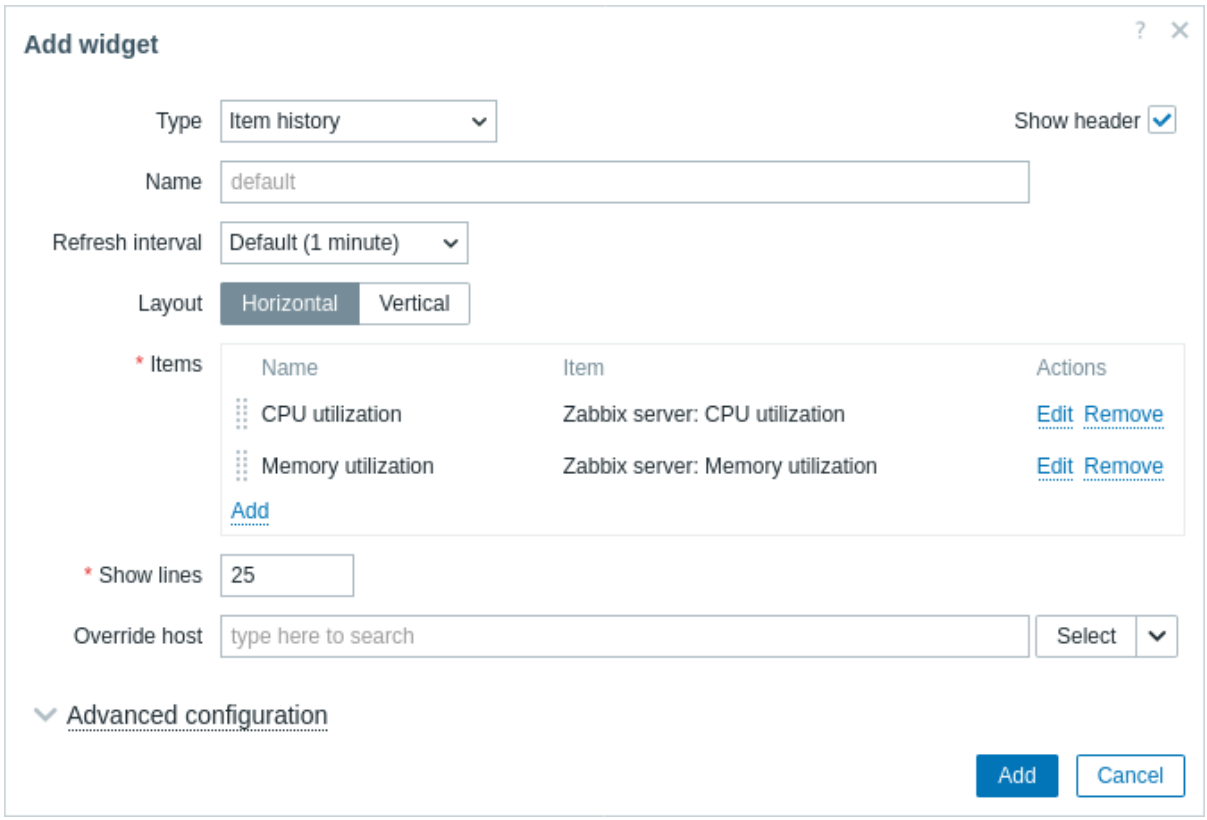




Можно отобразить до 1000 записей.

Конфигурация

Для настройки выберите *Item history* в качестве типа:

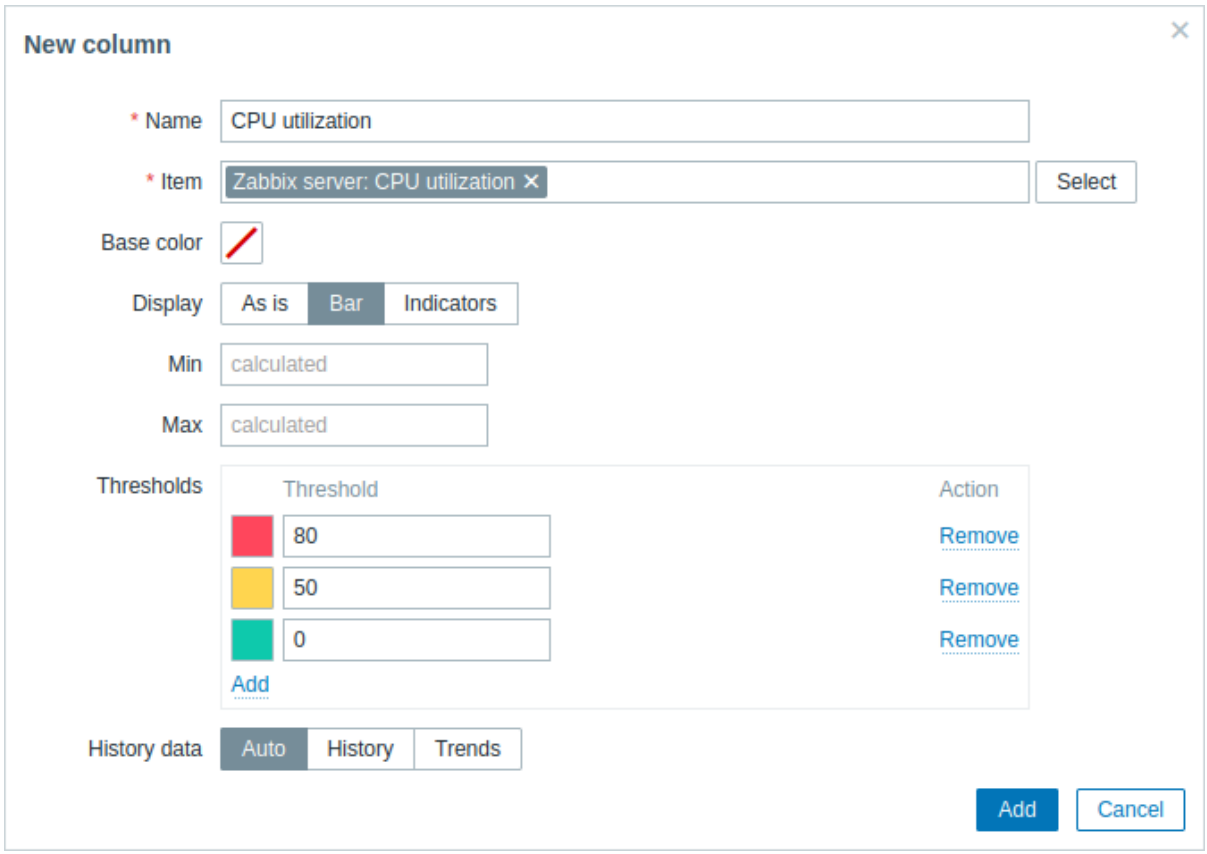


Помимо параметров, которые являются общими для всех виджетов, вы можете задать следующие специальные параметры:

Layout	Выберите вариант компоновки для столбцов элементов данных: **Horizontal** — элементы данных будут отображаться по горизонтали, значения — по вертикали; **Vertical** — элементы данных будут отображаться по вертикали, значения — по горизонтали.
Items	Добавьте столбцы элементов данных для отображения. Порядок элементов данных определяет порядок их отображения. Элементы данных можно переупорядочить, перетаскивая их вверх или вниз за маркер перед именем элемента данных.
Show lines	Укажите количество отображаемых строк значений элементов данных.
Override host	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.
Advanced configuration	Щелкните по метке *Advanced configuration*, чтобы отобразить параметры расширенной конфигурации.

Настройка столбцов

Чтобы настроить столбцы элементов данных, нажмите *Добавить* в параметре *Элементы данных*:



Общие параметры столбца:

Name	Введите имя столбца. Если оставить пустым, будет использовано имя элемента данных из параметра *Item*.
Item	Выберите элемент данных. Обратите внимание, что параметры настройки столбца различаются в зависимости от типа информации выбранного элемента данных; подробнее см. в описании отдельных параметров ниже. При настройке виджета на панели шаблона можно выбирать только элементы данных, настроенные в шаблоне.
Base color	Выберите цвет фона столбца или цвет заполнения, если для параметра *Display* установлено значение ”Bar” или ”Indicators”. Обратите внимание, что базовый цвет может быть переопределен цветами порогов или выделения.

Параметры столбца, специфичные для элементов данных числового типа:

Отображение	Выберите, как должно отображаться значение элемента данных: **Как есть** — как обычный текст; **Полоса** — как сплошная полоса, заполненная цветом; **Индикаторы** — как сегментированная полоса, заполненная цветом.
Мин	Введите минимальное значение для полосы/индикаторов. Если оставить пустым, виджет будет использовать минимальное значение элемента данных. Этот параметр доступен только если для *Отображение* установлено значение ”Полоса” или ”Индикаторы”.
Макс	Введите максимальное значение для полосы/индикаторов. Если оставить пустым, виджет будет использовать максимальное значение элемента данных. Этот параметр доступен только если для *Отображение* установлено значение ”Полоса” или ”Индикаторы”.
Пороги	Нажмите *Добавить*, чтобы добавить порог, выберите цвет порога в палитре цветов и укажите числовое значение. При сохранении список порогов будет отсортирован по возрастанию. Поддерживаются суффиксы (например, ”1d”, ”2w”, ”4K”, ”8G”). Поддерживаются преобразования значений.
Данные истории	Выберите, брать ли данные из истории или трендов: **Авто** — автоматический выбор; **История** — брать данные истории; **Тренды** — брать данные трендов.

Параметры столбца, специфичные для элементов данных типа character, text, log и JSON:

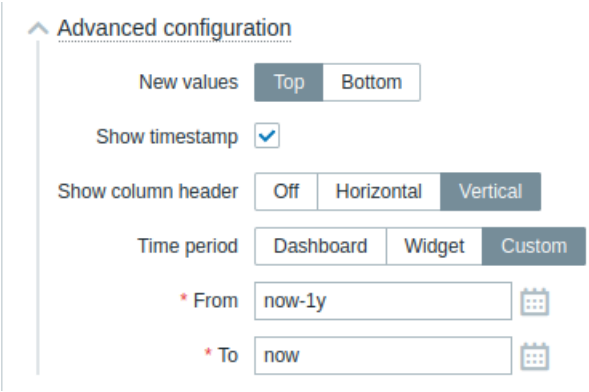
Подсветка	Нажмите *Добавить*, чтобы добавить подсветку, выберите цвет подсветки в палитре цветов и укажите регулярное выражение. Выбранный цвет будет использоваться как цвет фона для значений элемента данных, где указанное регулярное выражение соответствует тексту.
Отображение	Выберите, как должно отображаться значение элемента данных: **Как есть** — точно в том виде, в котором оно получено. Переносы строк сохраняются, если значение содержит символы перевода строки. Перенос слов применяется, если для *Макета* установлено значение ”Вертикально” и настроено более одного столбца; **HTML** — как текст, отформатированный в HTML; **Одна строка** — как одна строка, обрезанная до указанной длины (1–500 символов). При наведении курсора или щелчке по обрезанному значению открывается всплывающее окно с полным значением.
Использовать моноширинный шрифт	Установите этот флажок, чтобы отображать значение элемента данных моноширинным шрифтом (по умолчанию не установлен).
Отображать локальное время	Установите этот флажок, чтобы в столбце временной метки отображалось локальное время вместо timestamp. Обратите внимание, что флажок *Показывать временную метку* в расширенной конфигурации также должен быть установлен. Этот параметр доступен только для элементов данных типа log.

Параметры столбца, специфичные для элементов данных двоичного типа:

Показывать миниатюру	Установите этот флажок, чтобы отображать миниатюру для двоичных данных изображения или параметр ”Показать” для двоичных данных, не являющихся изображениями. Снимите этот флажок, чтобы отображать параметр ”Показать” для всех значений двоичных элементов данных. При наведении курсора или щелчке по параметру ”Показать” открывается всплывающее окно со значением элемента данных (изображение или строка Base64). Если значение элемента данных является пустой строкой, отображается параметр ”Показать”; при наведении курсора или щелчке по нему открывается всплывающее окно с текстом ”Пустая строка”.

Расширенная настройка

Параметры расширенной настройки доступны в сворачиваемом разделе *Расширенная настройка*:

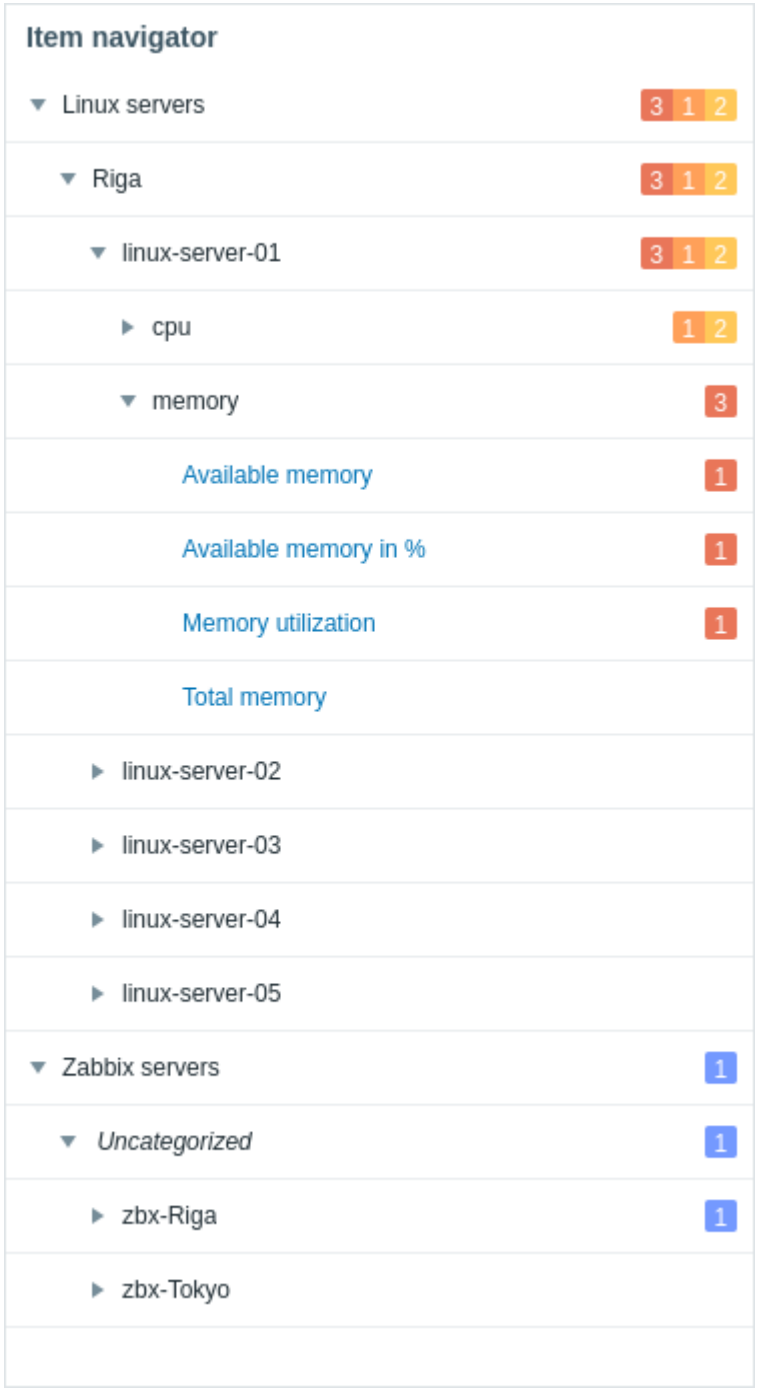


Новые значения	Выберите, где должны добавляться новые значения элементов данных: **Сверху** — вверху столбцов; **Снизу** — внизу столбцов.
Показывать метку времени	Установите этот флажок, чтобы отображать столбец метки времени (по умолчанию не установлен).
Показывать заголовок столбца	Выберите ориентацию заголовка столбца: **Выкл.** — скрыть заголовок; **Горизонтально** — отображать заголовок горизонтально; **Вертикально** — отображать заголовок вертикально.
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 мыши. 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 (*График*, *График (классический)*, *Прототип графика*) 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".

17 Навигатор элементов данных

Обзор

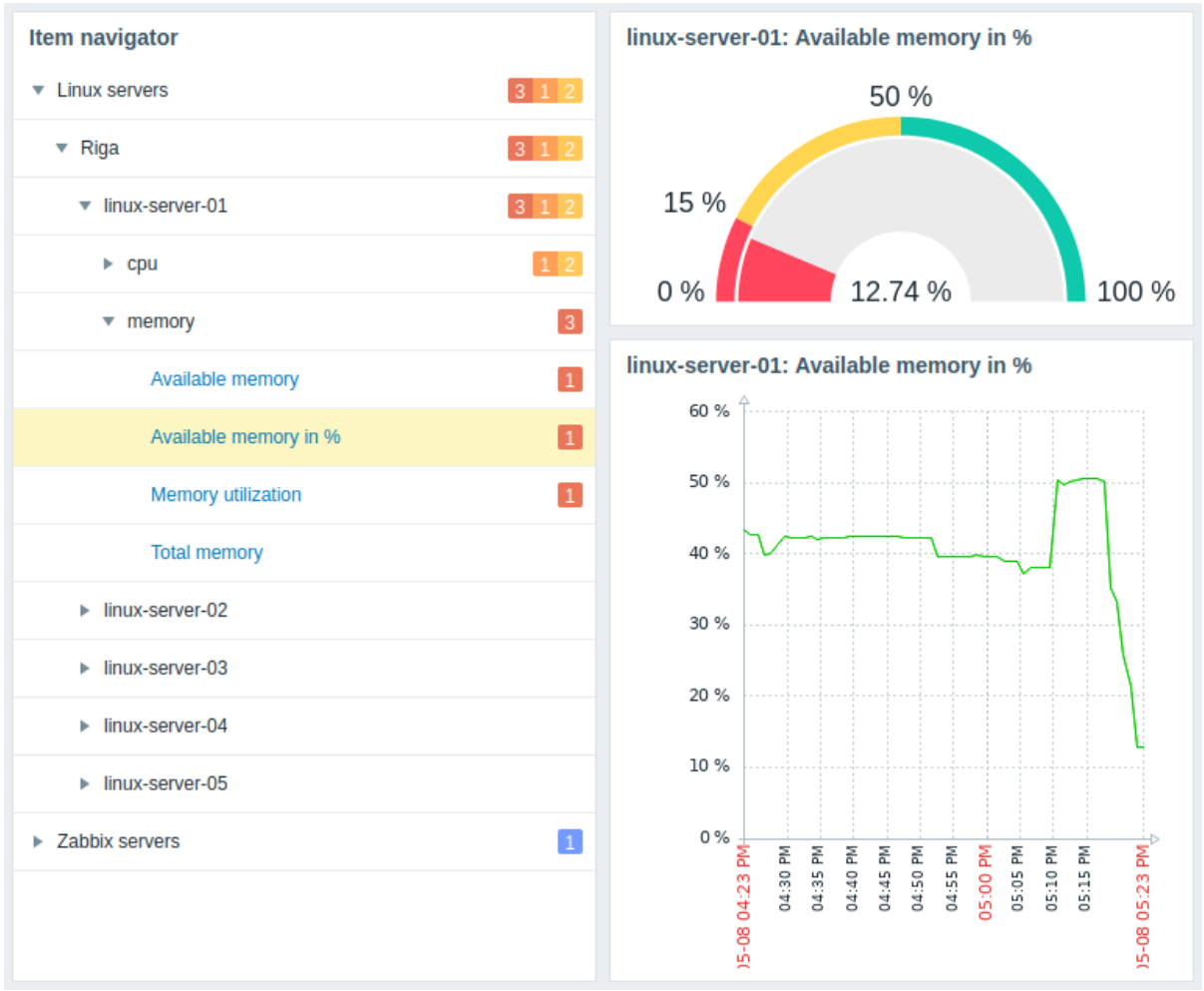
Виджет *Навигатор элементов данных* отображает список элементов данных на основе различных параметров фильтрации и группировки.



Группы, по которым организованы элементы данных, можно разворачивать и сворачивать.

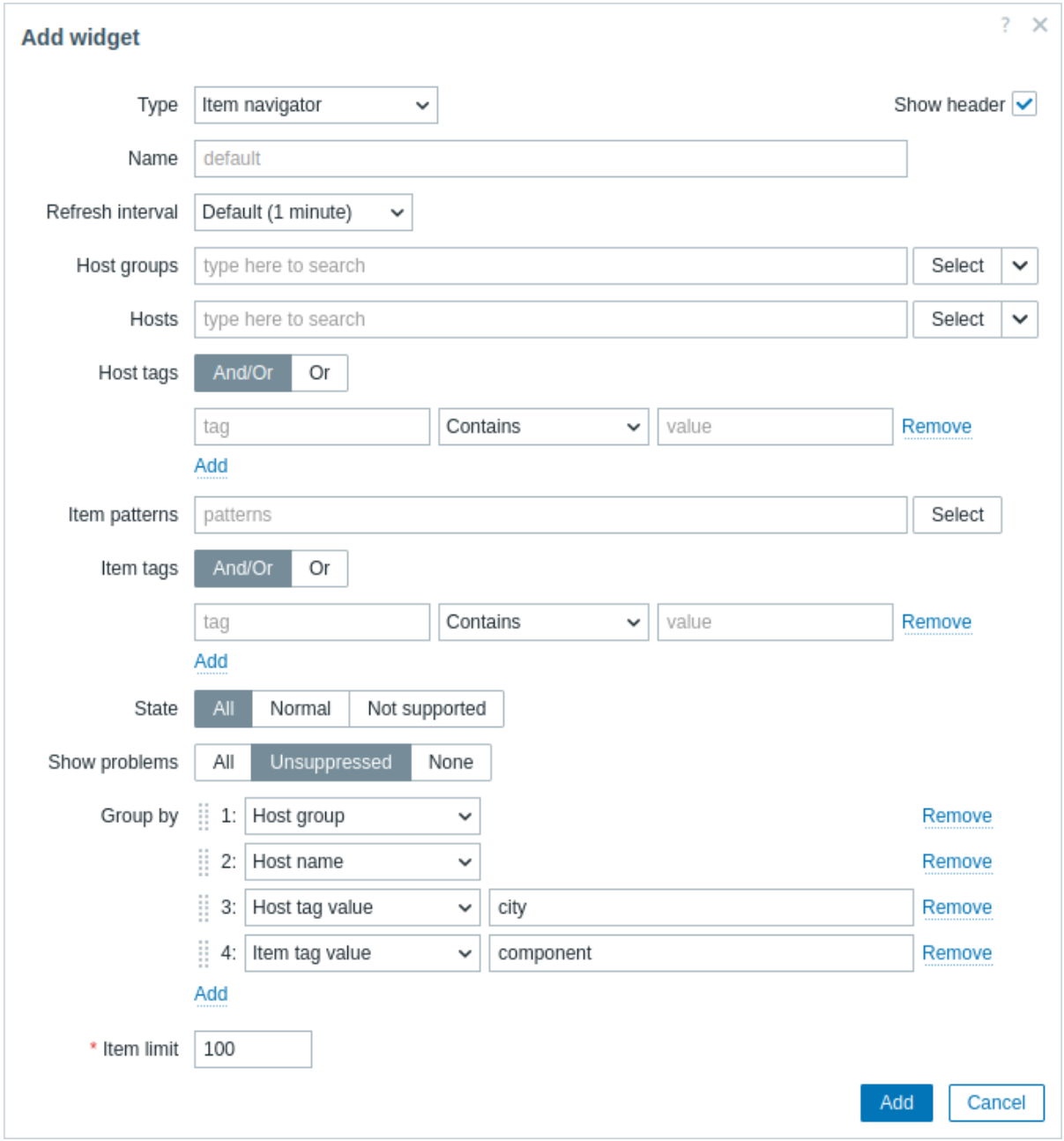
Для групп и проблем дополнительная информация доступна во всплывающих подсказках при наведении указателя мыши.

Этот виджет наиболее полезен для управления тем, что отображают другие виджеты, на основе выбранного элемента данных.



Конфигурация

Для настройки выберите *Навигатор элементов данных* в качестве типа:



В дополнение к параметрам, общим для всех виджетов, можно задать следующие специальные параметры:

Группы узлов сети	Выберите группы узлов сети. В качестве источника данных для групп узлов сети также можно выбрать совместимый виджет. Это поле поддерживает автодополнение, поэтому при вводе имени группы будет предложен выпадающий список подходящих групп. При выборе родительской группы узлов сети неявно выбираются все вложенные группы узлов сети; если не выбрана ни одна группа узлов сети, виджет будет отображать элементы данных, принадлежащие всем узлам сети из всех групп узлов сети. Этот параметр недоступен при настройке виджета на панели шаблона.
Узлы сети	Выберите узлы сети. В качестве источника данных для узлов сети также можно выбрать совместимый виджет или панель. Это поле поддерживает автодополнение, поэтому при вводе имени узла сети будет предложен выпадающий список подходящих узлов сети. Если не выбран ни один узел сети, виджет будет отображать элементы данных, принадлежащие всем узлам сети. Этот параметр недоступен при настройке виджета на панели шаблона.

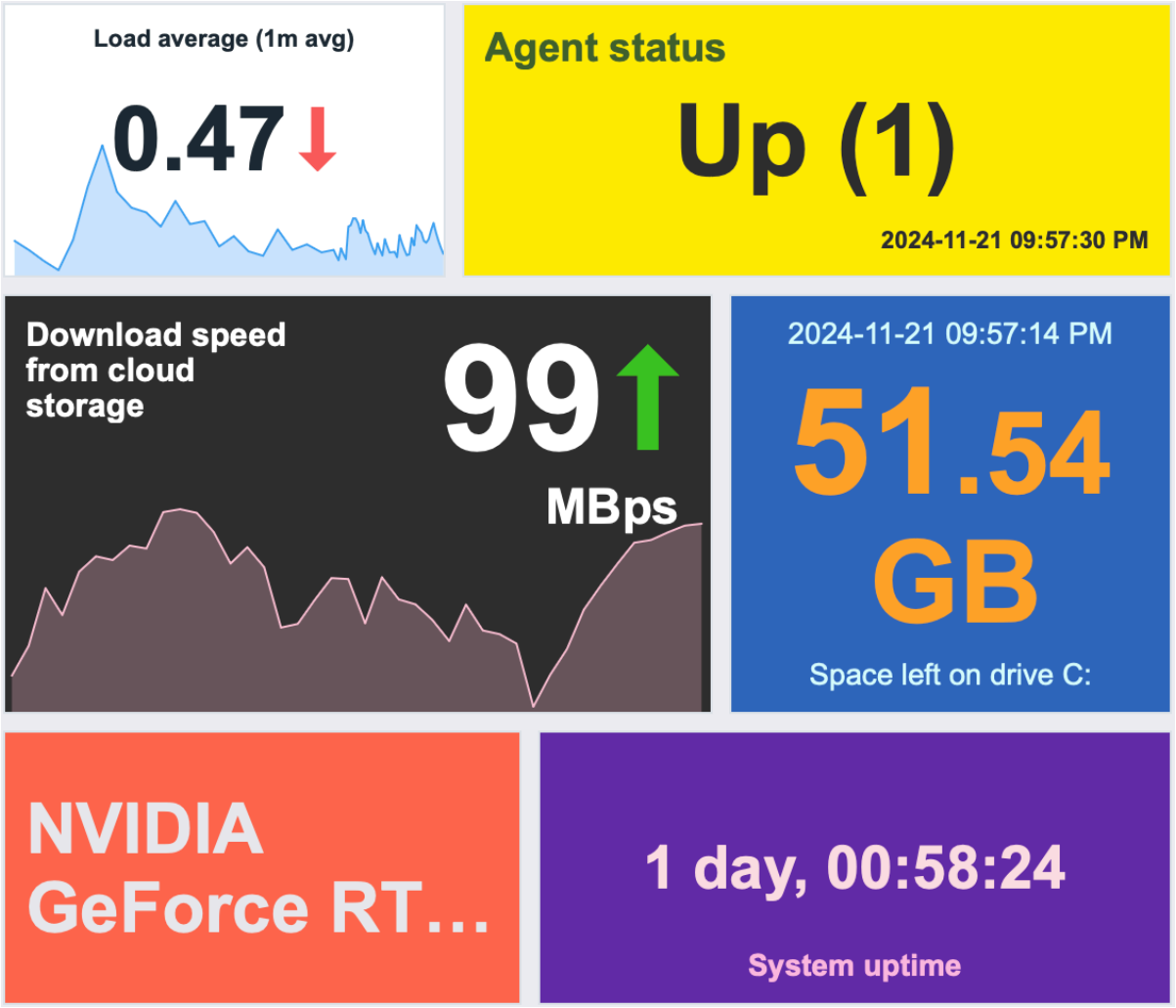
Теги узлов сети	Укажите теги для фильтрации элементов данных, отображаемых в виджете. Можно как включать, так и исключать определённые теги и значения тегов. Можно задать несколько условий. Сопоставление имён тегов всегда чувствительно к регистру. Для каждого условия доступны несколько операторов: **Существует** — 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 выполняться все условия, при этом условия с одинаковым именем тега будут сгруппированы по условию Или; **Или** — достаточно выполнения одного условия. Этот параметр недоступен при настройке виджета на панели шаблона.
Шаблоны элементов данных	Введите шаблоны элементов данных или выберите существующие элементы данных в качестве шаблонов элементов данных. В навигаторе элементов данных будут отображаться элементы данных, соответствующие указанным шаблонам. Для выбора можно использовать шаблоны с подстановочными знаками (например, `*` вернёт элементы данных, соответствующие нулю или более символам; `Zabbix*` вернёт элементы данных, начинающиеся с "Zabbix"). Чтобы указать шаблон с подстановочным знаком, введите строку вручную и нажмите *Enter*. Когда вы начинаете ввод, в выпадающем списке будут показаны подходящие элементы данных, ограниченные теми, которые принадлежат выбранным *Узлам сети* или узлам сети в выбранных *Группах узлов сети*, если они заданы. Символ подстановки интерпретируется всегда, поэтому невозможно, например, добавить отдельно элемент данных с именем *item**, если существуют другие подходящие элементы данных (например, item2, item3). При настройке виджета на панели шаблона этот параметр позволяет выбирать только элементы данных, настроенные в шаблоне.
Теги элементов данных	Укажите теги для фильтрации 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 выполняться все условия, при этом условия с одинаковым именем тега будут сгруппированы по условию Или; **Или** — достаточно выполнения одного условия.
Состояние	Отфильтруйте, какие элементы данных отображать, на основе их состояния (все, нормальные, неподдерживаемые).

Показывать проблемы	Отфильтруйте, какие проблемы отображать, на основе их статуса (все, неподтверждённые подавлением, нет). Проблемы обозначаются цветными блоками в конце строки элемента данных. Цвета зависят от важности проблемы, которую можно настроить в *Администрирование* > *Общие* > *Параметры отображения триггеров*. Важности проблем можно изменять при обновлении проблем.
Группировать по	Добавьте атрибут группировки, по которому будут группироваться элементы данных: **Группа узлов сети** — группировать элементы данных по группам узлов сети их узлов сети; **Имя узла сети** — группировать элементы данных по их 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 переупорядочивать, перетаскивая их вверх или вниз за маркер перед именем группы. Обратите внимание, что порядок атрибутов группировки определяет порядок вложенности групп. Например, указание нескольких имён тегов узла сети (1: color, 2: city) приведёт к тому, что элементы данных будут сначала сгруппированы по color (red, blue и т. д.), а затем по city (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 такие элементы данных они выбираются и подсвечиваются 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 всех подчинённых элементов 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 будут сгруппированы.
Лимит элементов данных	Введите максимальное количество отображаемых элементов данных. Возможные значения: от 1 до 9999. Если для отображения доступно больше элементов данных, чем установленный лимит, под отображаемыми элементами данных будет показано соответствующее сообщение (например, ”Показано 100 из 100+ элементов данных”). Обратите внимание, что настроенный лимит элементов данных также влияет на отображение настроенных групп; например, если лимит элементов данных установлен в 100, а элементы данных сгруппированы по их узлам сети (каждый из которых содержит 200 элементов данных), в виджете будет отображён только первый узел сети с его 100 элементами данных. На этот параметр не влияет параметр *Лимит результатов поиска и фильтрации* в *Администрирование* → *Общие* → *GUI*.

18 Значение элемента данных

Обзор

Виджет *Значение элемента данных* отображает значение одного числового или строкового элемента данных. Он полезен для отслеживания важных метрик, визуализации пороговых значений и обнаружения резких изменений в данных.



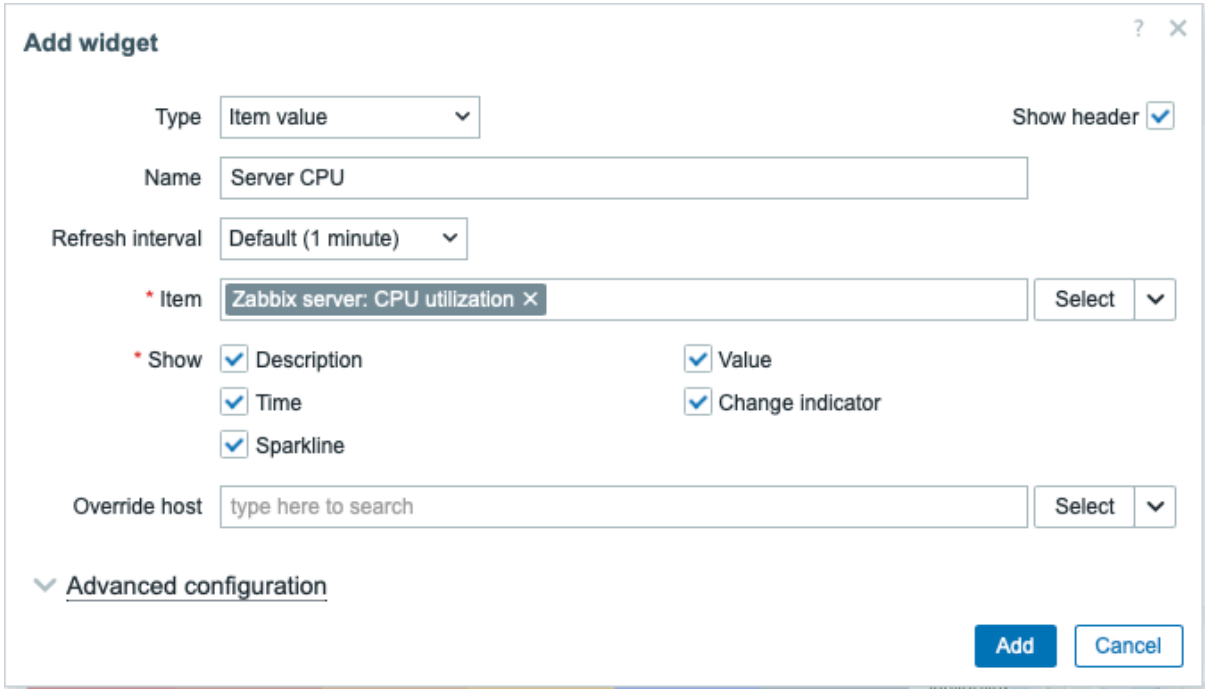
Вы можете настроить виджет для отображения следующего:

- Временная метка 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
- Спарклайн для значений за указанный период времени

Щелчок по виджету открывает ad-hoc graph для числовых элементов данных или последние данные для строковых элементов данных.

Настройка

Для настройки выберите *Значение элемента данных* в качестве типа виджета:



В дополнение к параметрам, общим для всех виджетов, вы можете задать следующие специальные параметры:

Элемент данных	Выберите элемент данных. В качестве источника данных для элементов данных также можно выбрать совместимый виджет. Элементы данных, возвращающие двоичные данные, не поддерживаются.
Показывать	Установите флажок, чтобы отображать соответствующий элемент: описание, значение, время, индикатор изменения, спарклайн. Снимите флажок, чтобы скрыть. Должен быть выбран как минимум один элемент.
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.
Расширенная настройка	Нажмите метку *Расширенная настройка*, чтобы отобразить параметры расширенной настройки.

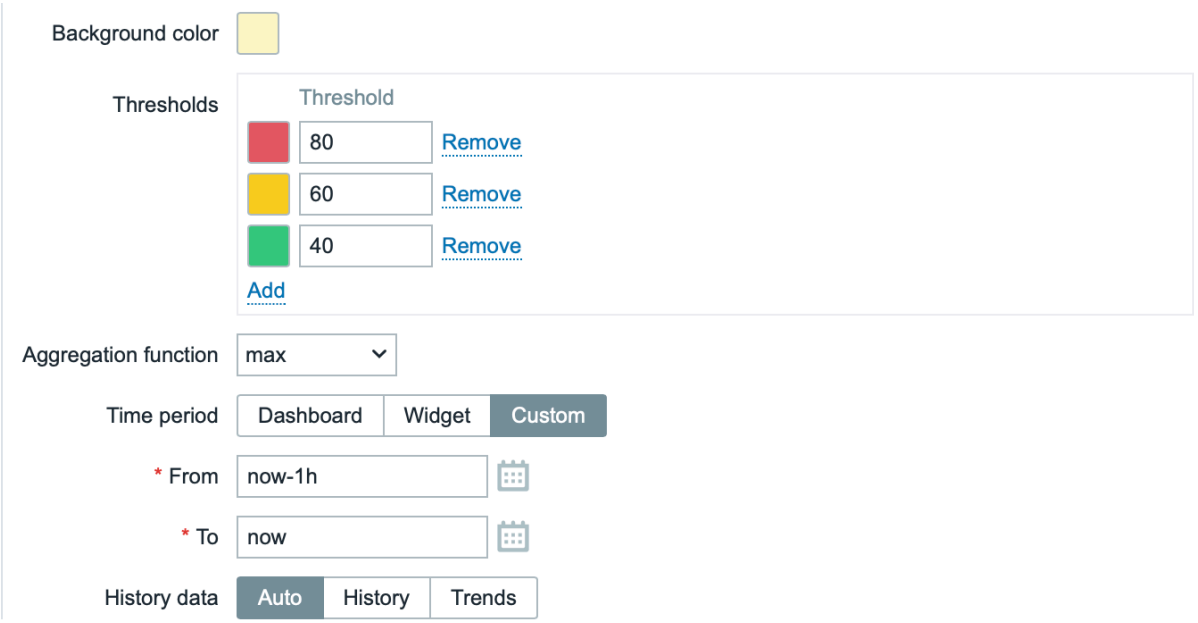
Расширенная настройка

Параметры расширенной настройки доступны в сворачиваемом разделе *Расширенная настройка*, и только для тех элементов данных, которые выбраны в поле *Показать* (см. выше).

Обратите внимание, что несколько элементов данных не могут занимать одно и то же место; если они размещены в одном и том же месте, будет показано сообщение об ошибке.

Общие параметры виджета

Эти параметры определяют цвет фона (статический или динамический) для всего виджета и функцию агрегации для отображения значений.

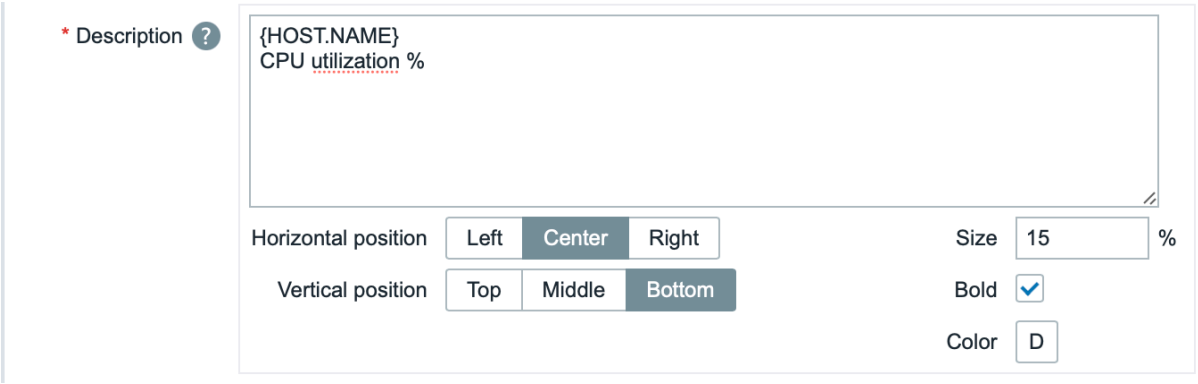


Цвет фона	Выберите цвет фона для всего виджета в палитре цветов. D обозначает цвет по умолчанию (зависит от темы веб-интерфейса). Чтобы вернуть значение по умолчанию, нажмите кнопку *Use default* в палитре цветов.
Пороги	Настройте динамический цвет фона для всего виджета. Нажмите *Add*, чтобы добавить порог, выберите цвет фона в палитре цветов и укажите числовое значение. Когда значение элемента данных станет равным или превысит пороговое значение, цвет фона изменится. При сохранении список будет отсортирован по возрастанию. Обратите внимание, что динамический цвет фона будет корректно отображаться только для числовых элементов данных.
Функция агрегации	Укажите, какую функцию агрегации использовать: **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; **not used** - отображать самое последнее значение (без агрегации). Агрегация позволяет отображать агрегированное значение за выбранный интервал (5 минут, час, день) вместо самого последнего значения. Для *min*, *max*, *avg* и *sum* могут отображаться только числовые данные. Для *count* нечисловые данные будут преобразованы в числовые.
Период времени	Период времени, используемый для агрегации значений. Выберите источник данных для периода времени: **Dashboard** - использовать селектор периода времени панели; **Widget** - использовать совместимый виджет (задается 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 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. Этот параметр недоступен, если для *Функция агрегации* установлено значение "not used".
Widget	Введите или выберите совместимый виджет 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".
Данные истории	Брать данные из истории или трендов: **Auto** - автоматический выбор; **History** - брать данные истории; **Trends** - брать данные трендов. Этот параметр применяется только к числовым данным. Нечисловые данные всегда будут браться из истории.

Описание

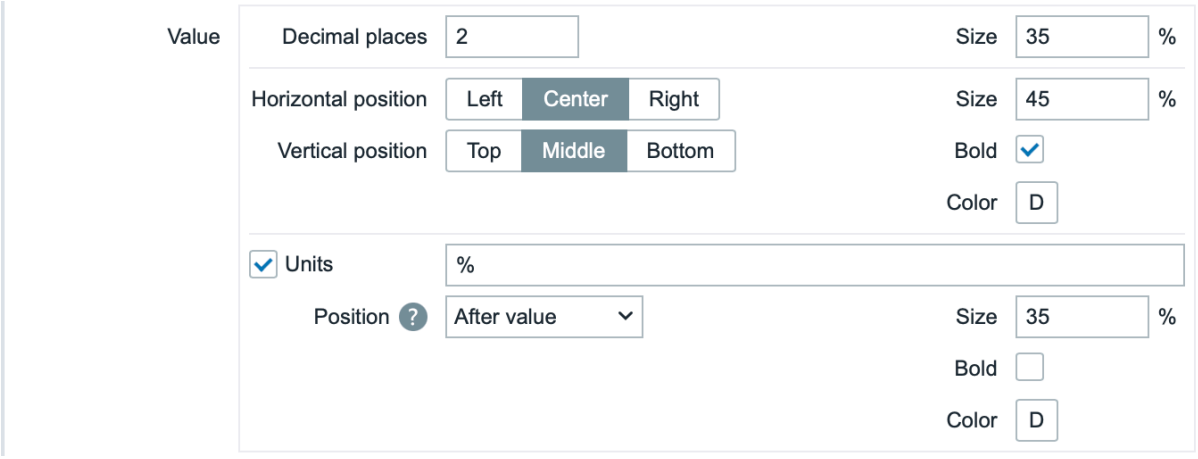
Эти параметры определяют, как должно отображаться описание элемента данных.



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

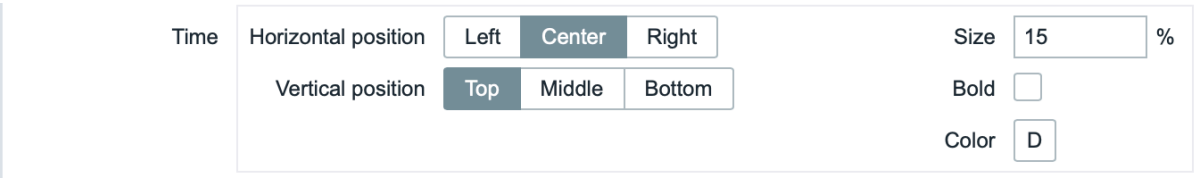
Эти параметры определяют, как должно отображаться значение элемента данных.



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.



Индикаторы изменения следующие:

- ↑ — 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:

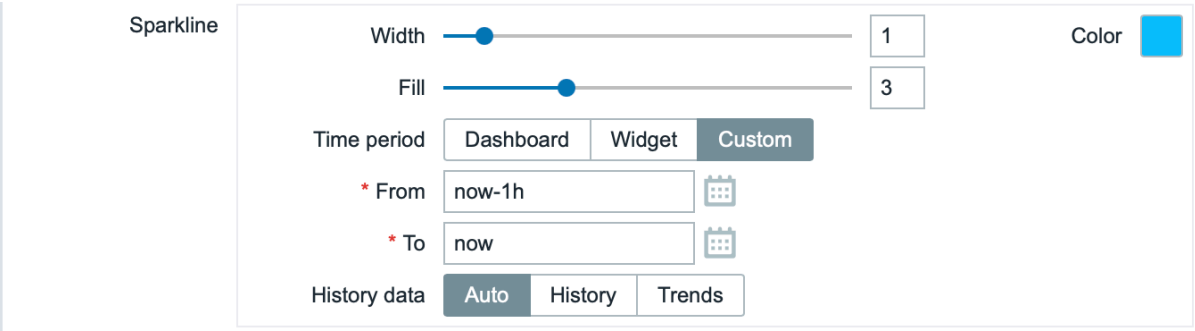
- Индикаторы увеличения (↑) и уменьшения (↓) не отображаются, если имеется только одно значение.
- Значения всегда сравниваются в пределах одной и той же области; например:
  - Последнее значение сравнивается с предыдущим значением.
  - Значение за месяц сравнивается с предыдущим месяцем.
  - Для агрегаций предыдущий период имеет ту же длительность, что и выбранный период, и заканчивается непосредственно перед началом выбранного периода.

D обозначает цвет по умолчанию (зависит от темы веб-интерфейса). Чтобы вернуться к значению по умолчанию, нажмите кнопку *Use default* 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.



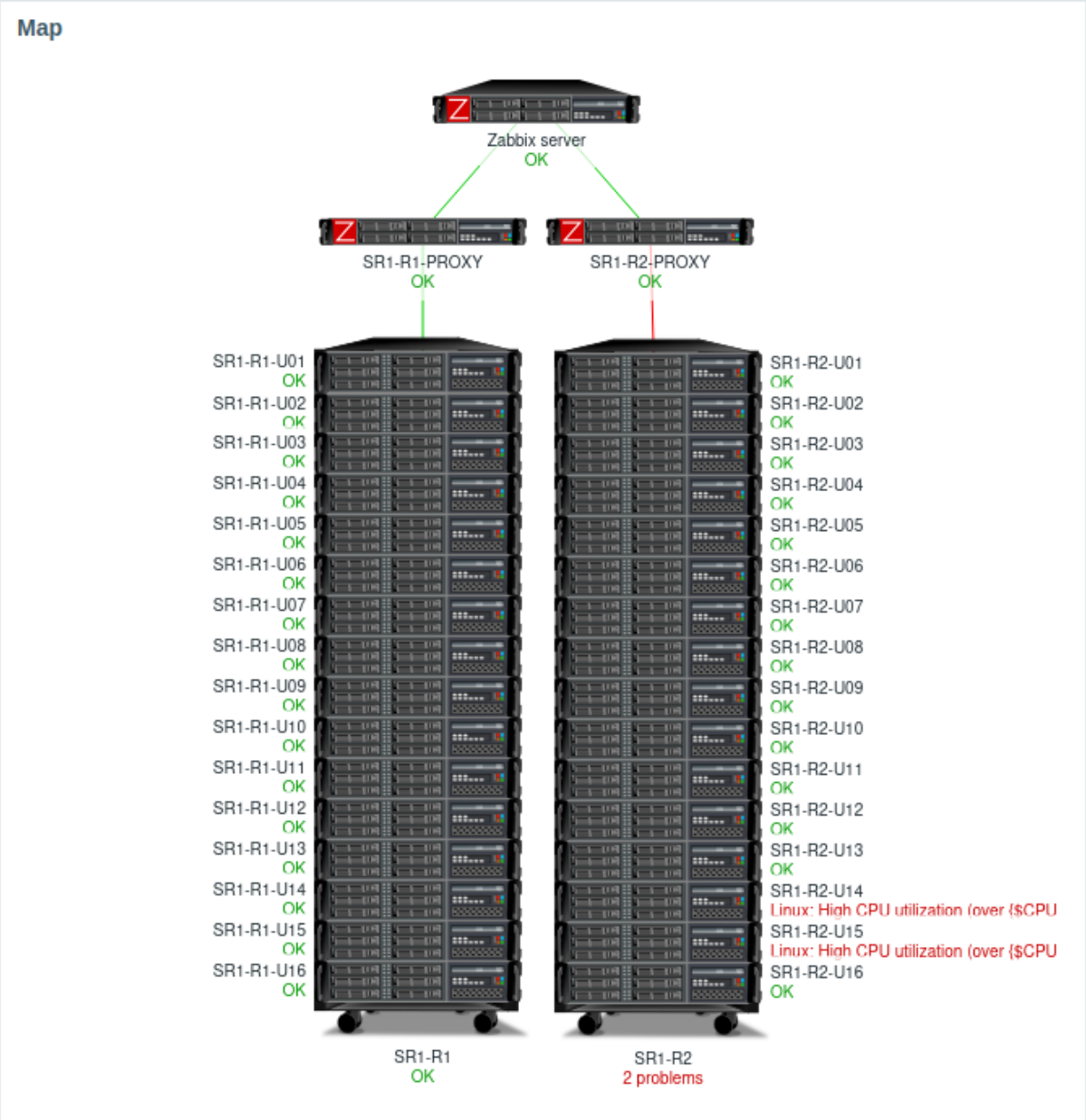
Ширина	Задайте толщину линии графика с помощью ползунка или вручную введя значение в диапазоне от 0 до 10.
Цвет	Выберите цвет линии и заливки.
Заливка	Задайте уровень прозрачности цвета заливки с помощью ползунка или вручную введя значение в диапазоне от 0 до 10.
Период времени	Укажите период времени, 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.

19 Карта

Обзор

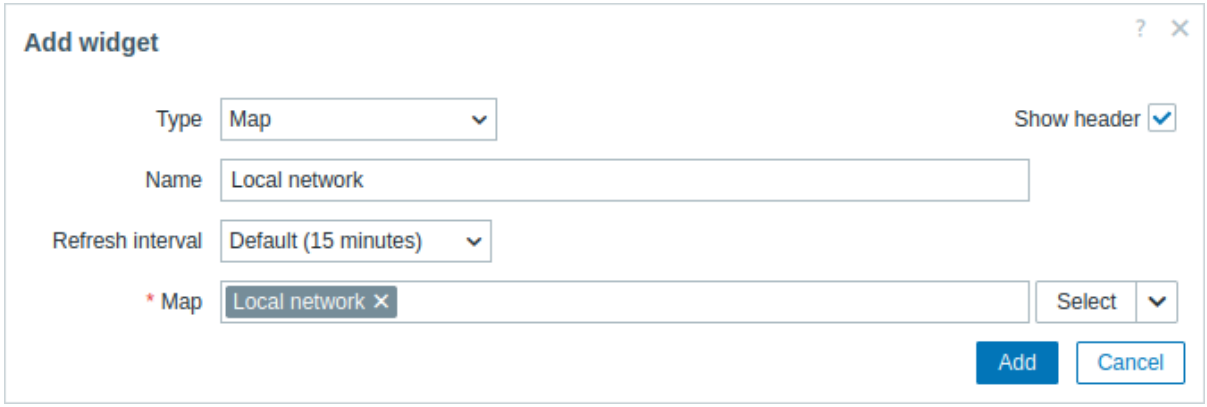
Виджет *Карта* может отображать карту сети или аналогичную визуализацию, предоставляя вам динамический обзор вашей сети.



Вы можете использовать его вместе с виджетом Дерево навигации по картам, чтобы динамически отображать карту, выбранную в дереве.

Настройка

Для настройки выберите *Карта* в качестве типа:



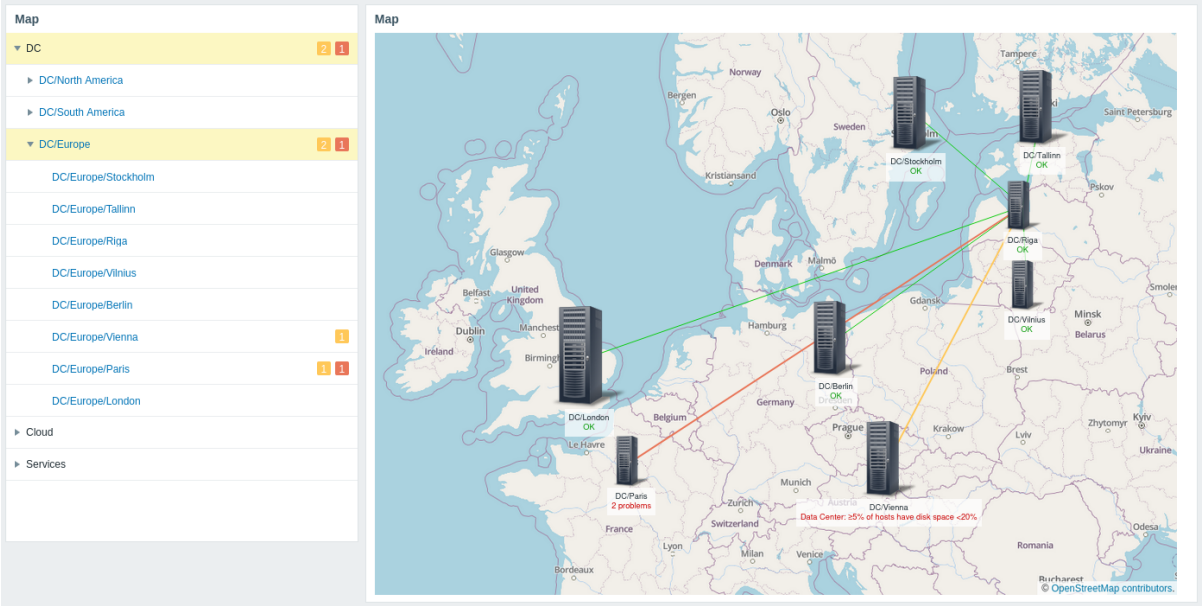
В дополнение к параметрам, которые являются общими для всех виджетов, вы можете задать следующие специальные параметры:

Карта	Укажите карту для отображения. Либо выберите совместимый виджет в качестве источника данных для отображаемой карты. Это поле поддерживает автодополнение, поэтому при вводе имени карты или виджета будет предложен выпадающий список подходящих карт или виджетов.

20 Дерево навигации по картам

Обзор

Виджет *Дерево навигации по картам* отображает иерархию существующих карт и количество проблем для каждой карты и группы карт.

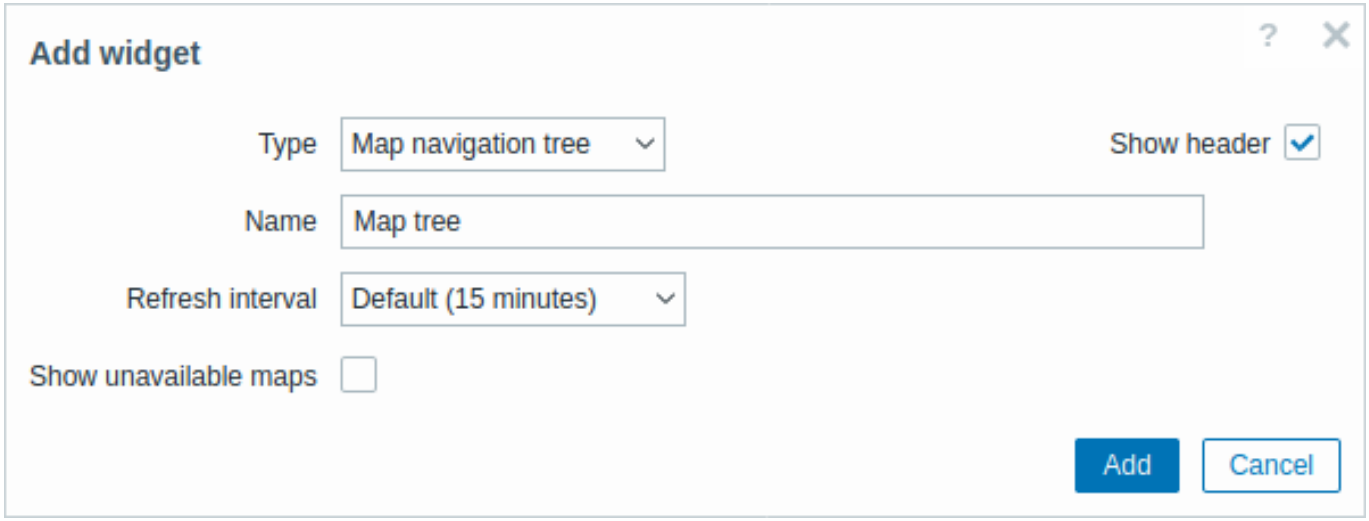


Его можно использовать вместе с виджетом Карта, чтобы динамически отображать карту, выбранную в дереве.

Количество проблем для карты верхнего уровня в иерархии включает все проблемы подчинённых карт, а также её собственные.

Настройка

Для настройки выберите тип *Дерево навигации карт*:

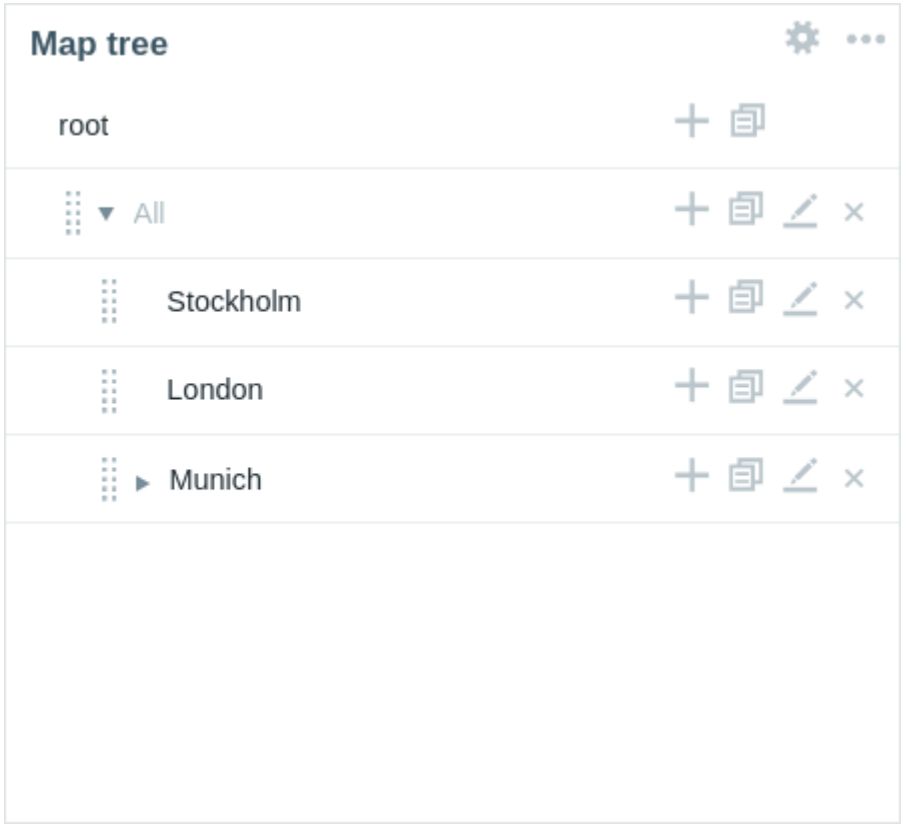


В дополнение к параметрам, которые являются общими для всех виджетов, вы можете задать следующие специальные параметры:

Показывать недоступные карты	Установите этот флажок, чтобы отображать карты, на чтение которых у пользователя нет прав. Недоступные карты в дереве навигации будут отображаться с серым значком. Обратите внимание, что если этот флажок установлен, доступные подкарты отображаются, даже если карта родительского уровня недоступна. Если флажок снят, доступные подкарты недоступной родительской карты не будут отображаться вовсе. Количество проблем рассчитывается на основе доступных карт и доступных элементов карты.

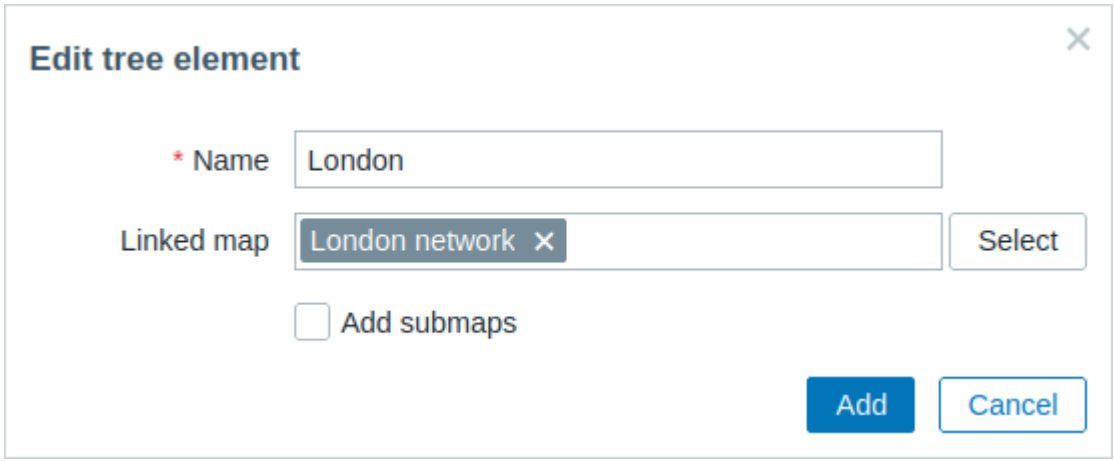
Элементы дерева навигации отображаются в виде списка. Вы можете:

- перетащить элемент (включая его дочерние элементы) на новое место в списке;
- развернуть или свернуть элемент, чтобы отобразить или скрыть его дочерние элементы;
- добавить дочерний элемент (с привязанной картой или без неё) к элементу;
- добавить несколько дочерних элементов (с привязанными картами) к элементу;
- изменить элемент;
- удалить элемент (включая его дочерние элементы).



Настройка элемента данных

Чтобы настроить элемент дерева навигации, либо добавьте новый элемент, либо отредактируйте существующий элемент.



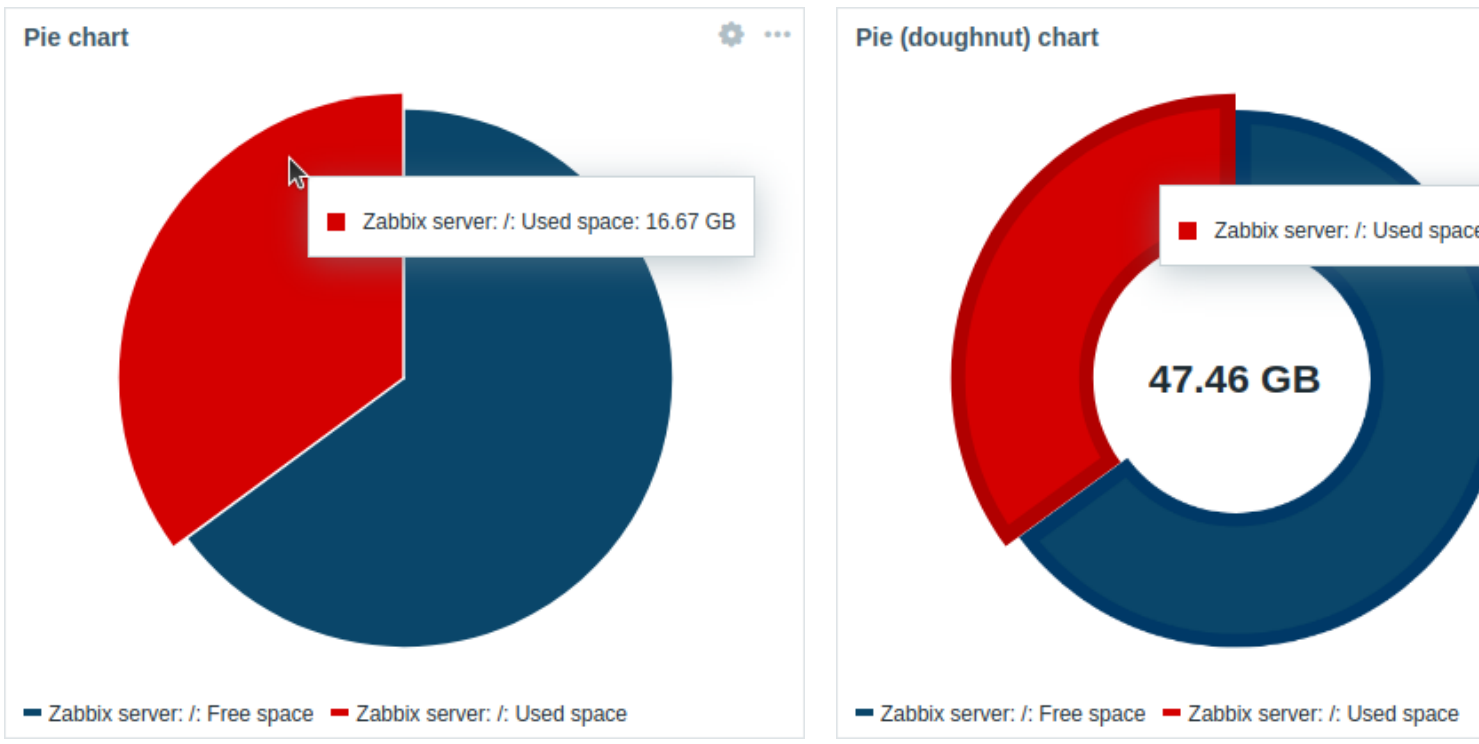
Доступны следующие параметры настройки элемента дерева навигации:

Name	Введите имя элемента дерева навигации.
Linked map	Выберите карту, которую нужно связать с элементом дерева навигации. Это поле поддерживает автодополнение, поэтому при вводе имени карты будет предложен выпадающий список подходящих карт.
Add submaps	Установите этот флажок, чтобы добавить подкарты связанной карты как дочерние элементы элемента дерева навигации.

21 Круговая диаграмма

Обзор

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.

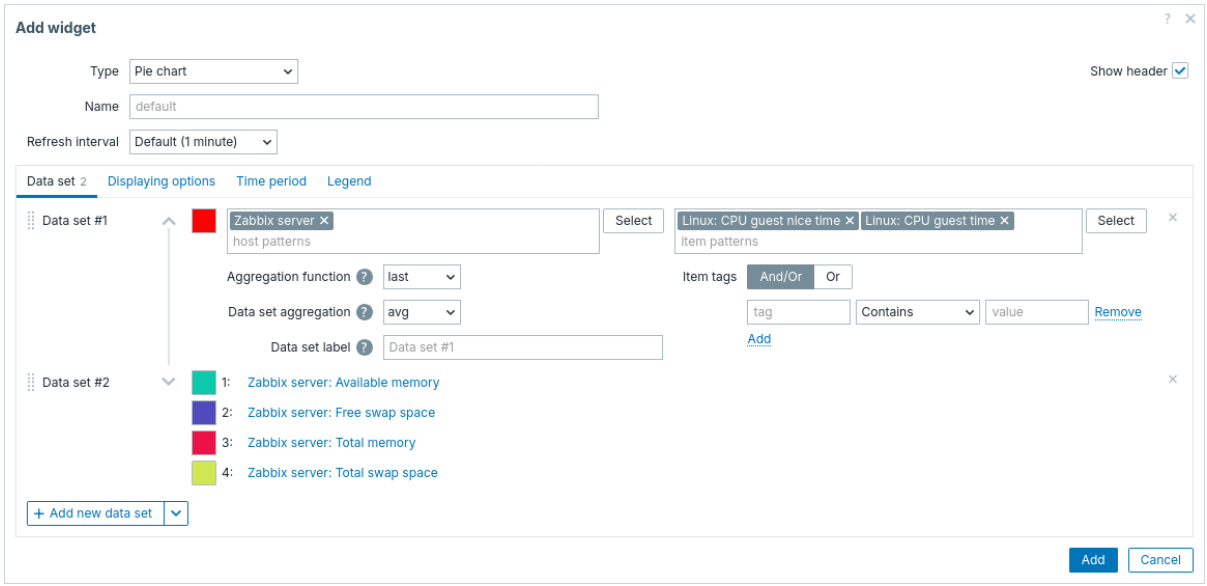


При наведении курсора выделенный сектор расширяется и показывает свою легенду; при щелчке по сектору он остается расширенным, а его легенда остается видимой до закрытия.

Информацию, отображаемую в виджете *Круговая диаграмма*, можно скачать как PNG-изображение, выбрав опцию *Download image* в меню виджета.

Настройка

Чтобы настроить, выберите *Круговая диаграмма* в качестве типа:



Набор данных

Вкладка **Набор данных** позволяет выбирать данные для круговой диаграммы, добавляя наборы данных. Можно добавить два типа наборов данных:

- *Шаблоны элементов данных* — отображаются данные из элементов данных, соответствующих шаблонам. Можно выбрать один базовый цвет или строку палитры, чтобы назначить разные цвета каждому найденному элементу данных.
- *Список элементов данных* — отображаются данные из выбранных элементов данных. Можно выбрать цвет каждого элемента данных отдельно с помощью палитры.

По умолчанию добавляется набор данных *Шаблоны элементов данных*.

Набор данных	Для набора данных **Шаблоны элементов данных**: Выберите или введите шаблоны узла сети и элемента данных; данные элементов данных, соответствующих введённым шаблонам, будут отображаться на круговой диаграмме; может быть отображено до 50 элементов данных. Для выбора можно использовать шаблоны с подстановочными знаками (например, * вернёт результаты, соответствующие нулю или более символам). Чтобы указать шаблон с подстановочным знаком, введите строку вручную и нажмите *Enter*. Символ подстановки всегда интерпретируется как шаблон, поэтому невозможно, например, добавить отдельно элемент данных с именем *item** при наличии других подходящих элементов данных (например, *item2*, *item3*). Указание шаблонов узла сети и элемента данных обязательно для наборов данных "Шаблоны элементов данных". См. также: Подробности настройки набора данных. Для набора данных **Список элементов данных**: Выберите элементы данных для круговой диаграммы, нажав кнопку *Добавить элемент данных*. Выпадающий список *Тип* после имени элемента данных позволяет выбрать тип отображения для каждого элемента данных: **Обычный** — значение элемента данных отображается на круговой диаграмме пропорционально (по умолчанию); **Итог** — значение элемента данных занимает всю круговую диаграмму. Обратите внимание, что для одной круговой диаграммы может существовать только один элемент данных типа "Итог", и он будет размещён первым в легенде круговой диаграммы. Если для элемента данных задан тип "Итог", параметр *Агрегация набора данных* (см. ниже) будет отключён и установлен в значение "не используется". Также можно выбрать совместимые виджеты в качестве источника данных для элементов данных, нажав кнопку *Добавить виджет*. К виджетам применяются те же параметры, что и к отдельным элементам данных. Указание элементов данных или виджетов обязательно для наборов данных "Список элементов данных". См. также: Подробности настройки набора данных. Обратите внимание, что допускаются только числовые типы элементов данных. При настройке виджета на панели шаблона параметр для указания шаблонов узла сети недоступен, а параметр для указания списка элементов данных позволяет выбирать только элементы данных, настроенные в шаблоне.
Функция агрегации	Укажите, какую функцию агрегации использовать для каждого элемента данных в наборе данных: **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). Агрегация позволяет отображать агрегированное значение за интервал (5 минут, час, день), выбранный на вкладке *Период времени* или используемый для всей панели.
Агрегация набора данных	Укажите, какую функцию агрегации использовать для всего набора данных: **not used** — без агрегации, 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. Агрегация позволяет отображать агрегированное значение за 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 настройке набора данных и в легенде круговой диаграммы (для агрегированных наборов данных). Все наборы данных нумеруются, включая те, для которых указана *Метка набора данных*. Если метка не указана, набор данных будет автоматически обозначен в соответствии со своим номером (например, ”Набор данных #2”, ”Набор данных #3” и т. д.). Нумерация наборов данных пересчитывается после изменения порядка/перетаскивания наборов данных. Слишком длинные метки наборов данных будут сокращены по месту отображения (например, ”Количество proc...”).
Теги элементов данных	Укажите теги для фильтрации 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: **Exists** — включить указанные имена тегов; **Equals** — включить указанные имена тегов и значения (с учётом регистра); **Contains** — включить указанные имена тегов, у которых значения тегов содержат введённую строку (поиск подстроки, без учёта регистра); **Does not exist** — исключить указанные имена тегов; **Does not equal** — исключить указанные имена тегов и значения (с учётом регистра); **Does not contain** — исключить указанные имена тегов, у которых значения тегов содержат введённую строку (поиск подстроки, без учёта 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.

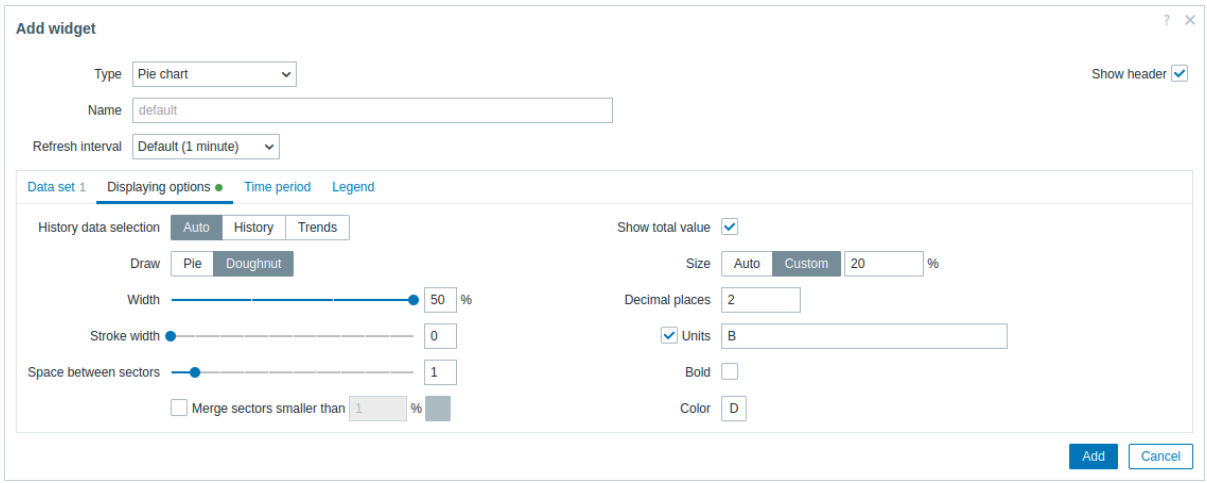
Сведения о настройке набора данных

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 настройке набора данных в виджете *Graph*. Эти сведения также применимы к виджету *Pie chart*.

Параметры отображения

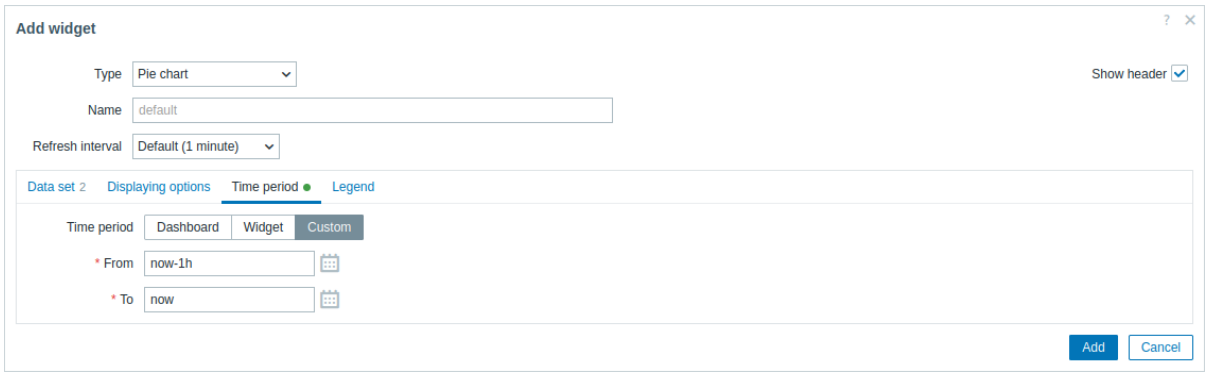
Вкладка **Параметры отображения** позволяет определить параметры выбора исторических данных и визуализации для круговой диаграммы:



Выбор исторических данных	Выберите источник данных: **Авто** — данные выбираются в соответствии с классическим алгоритмом (по умолчанию); **История** — данные из истории; **Тренды** — данные из трендов.
Отрисовка	Выберите стиль визуализации круговой диаграммы: **Круговая** — полная круговая диаграмма (секторы занимают 100% радиуса); **Кольцевая** — круговая диаграмма с пустым кругом посередине (секторы используют до 50% радиуса).
Пространство между секторами	Выберите размер пространства между секторами (в единицах от 0 до 10) (по умолчанию — ”1”).
Объединять сектора меньше N%	Установите флажок, чтобы объединять сектора меньше N%. Если параметр включен, выберите цвет для объединенных секторов и процентный порог (N) для объединения малых секторов.
Стиль отрисовки:	
Кольцевая	
Ширина	Выберите ширину кольцевой диаграммы: 20, 30, 40 или 50% (по умолчанию) от радиуса.
Ширина границы	Выберите ширину границы сектора кольцевой диаграммы (0-10).
Показывать общее значение	Установите флажок, чтобы показывать общее значение в центре кольцевой диаграммы.
Размер	Выберите параметр размера для общего значения: **Авто** — размер текста подбирается автоматически, чтобы он читабельно помещался в центре кольцевой диаграммы; **Пользовательский** — укажите размер текста как процент высоты от общей высоты виджета.
Десятичные знаки	Укажите количество знаков после запятой для общего значения (0-6).
Единицы измерения	Укажите единицы измерения для общего значения.
Жирный	Установите флажок, чтобы отображать общее значение жирным шрифтом.
Цвет	Выберите цвет для общего значения.

Период времени

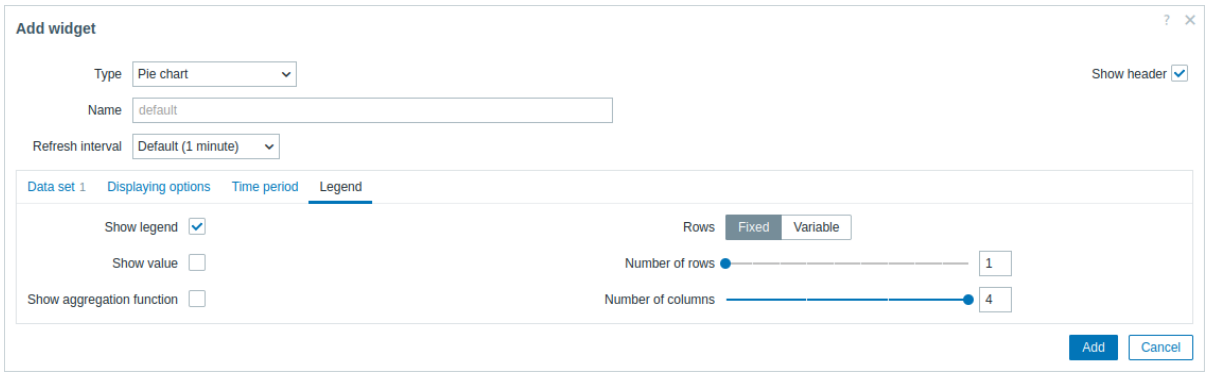
Вкладка **Период времени** позволяет задать пользовательский период времени для настроек агрегации круговой диаграммы:



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, указанный в параметрах *От* и *До*; если задан, в правом верхнем углу виджета будет отображаться значок часов, показывающий установленное время при наведении курсора мыши. Обратите внимание, что независимо от конфигурации *Период времени* виджета, совместимые виджеты все равно 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 *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".

Легенда

Вкладка **Легенда** позволяет настроить легенду круговой диаграммы:

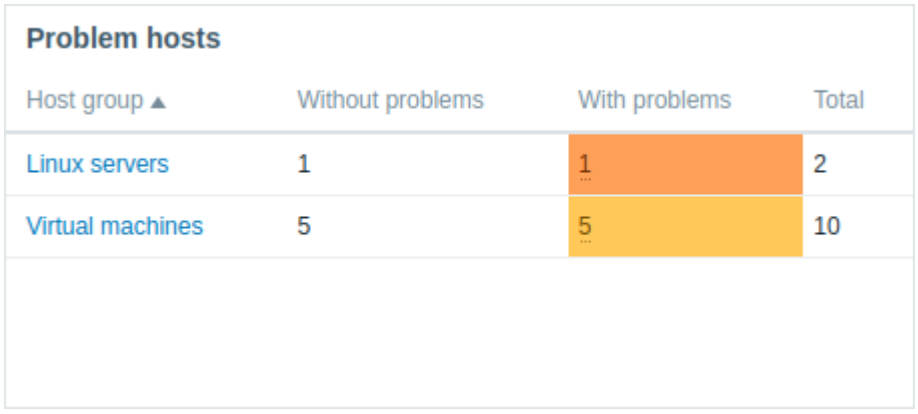


Показывать легенду	Снимите этот флажок, чтобы скрыть легенду на круговой диаграмме (по умолчанию установлен).
Показывать значение	Установите этот флажок, чтобы показывать значение элемента данных в легенде.
Показывать функцию агрегации	Установите этот флажок, чтобы показывать функцию агрегации в легенде.
Строки	Выберите режим отображения строк легенды: **Фиксированное** — количество отображаемых строк определяется значением параметра *Количество строк*; **Переменное** — количество отображаемых строк определяется числом настроенных элементов данных, но не превышает значение параметра *Максимальное количество строк*.
Количество строк/ Максимальное количество строк	Если для параметра *Строки* установлено значение "Фиксированное", задайте количество отображаемых строк легенды (1-10). Если для параметра *Строки* установлено значение "Переменное", задайте максимальное количество отображаемых строк легенды (1-10).
Количество столбцов	Задайте количество отображаемых столбцов легенды (1-4). Этот параметр доступен, если флажок *Показывать значение* снят.

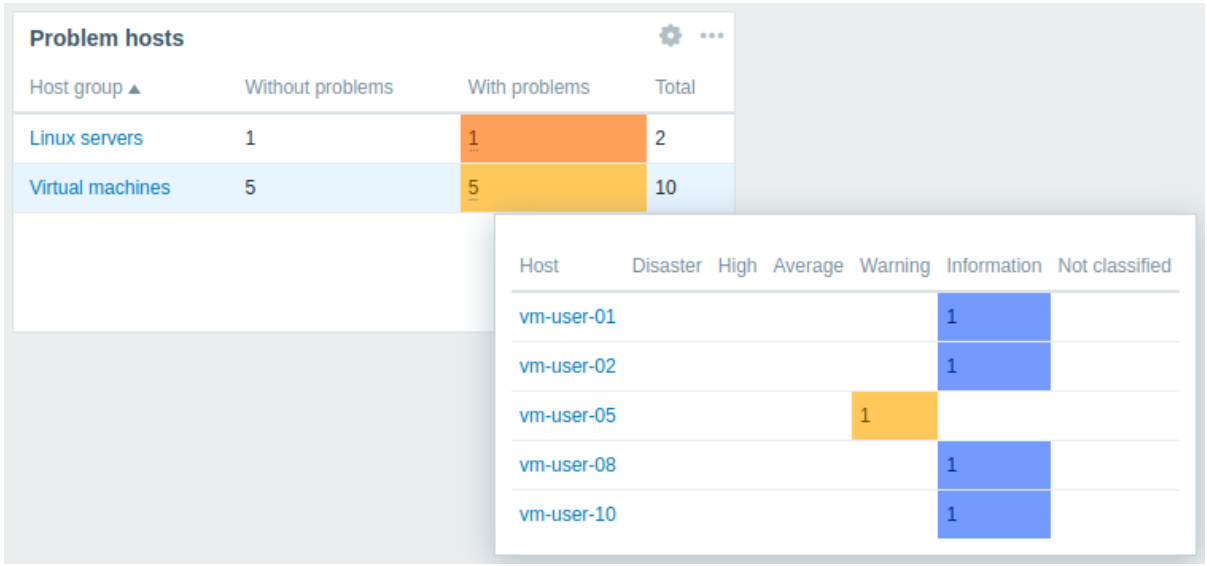
22 Проблемные узлы сети

Обзор

Виджет *Problem hosts* отображает количество первопричин проблем по группам узлов сети и показывает наивысший уровень важности проблемы в каждой группе. Он позволяет фильтровать проблемы по различным параметрам (группы узлов сети, узлы сети, имена проблем и т. д.).

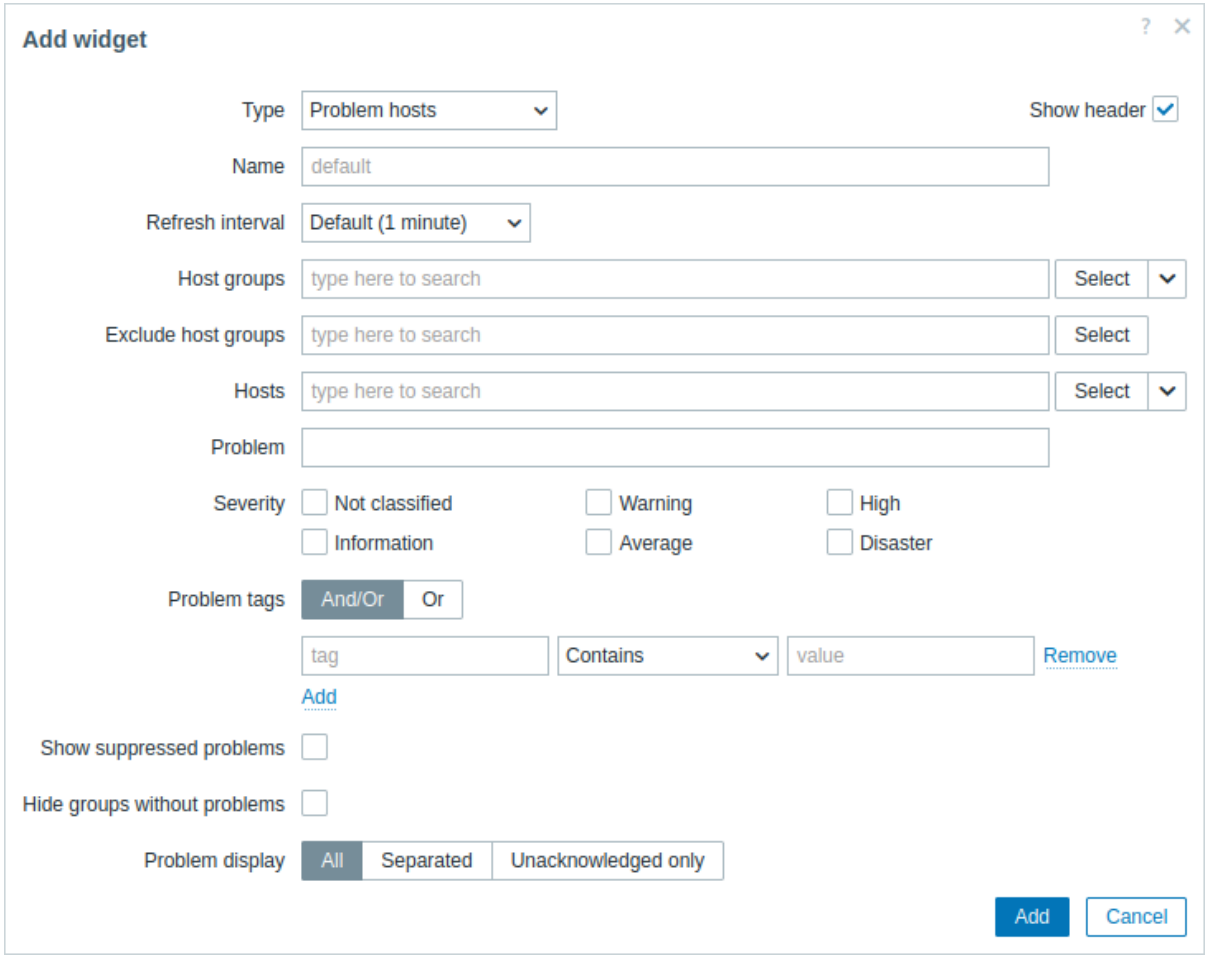


При наведении курсора на количество проблем отображается список узлов сети с проблемами:



Настройка

Для настройки выберите тип *Узлы сети с проблемами*:



В дополнение к параметрам, которые являются общими для всех виджетов, вы можете задать следующие специальные параметры:

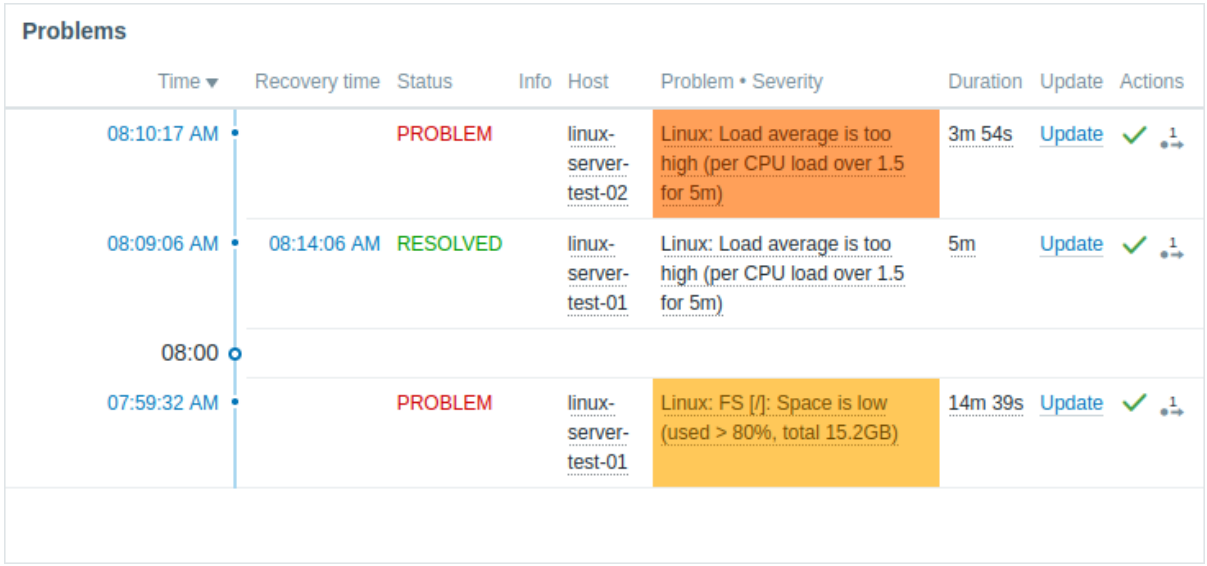
Группы узлов сети	Выберите группы узлов сети для отображения в виджете. В качестве источника данных для групп узлов сети также можно выбрать совместимый виджет. Это поле поддерживает автодополнение, поэтому при вводе имени группы будет предложен выпадающий список подходящих групп. Указание родительской группы узлов сети неявно выбирает все вложенные группы узлов сети. В виджете будут отображаться данные узлов сети из этих групп узлов сети; если группы узлов сети не указаны, будут отображаться все группы узлов сети. Этот параметр недоступен при настройке виджета на панели шаблона.
Исключить группы узлов сети	Выберите группы узлов сети, которые нужно скрыть в виджете. Это поле поддерживает автодополнение, поэтому при вводе имени группы будет предложен выпадающий список подходящих групп. Указание родительской группы узлов сети неявно выбирает все вложенные группы узлов сети. Данные узлов сети из этих групп узлов сети не будут отображаться в виджете. Например, узлы сети 001, 002, 003 могут входить в Группу A, а узлы сети 002, 003 также в Группу B. Если одновременно выбрать *показывать* Группу A и *исключить* Группу B, на панели будет отображаться только данные узла сети 001. Этот параметр недоступен при настройке виджета на панели шаблона.
Узлы сети	Выберите узлы сети для отображения в виджете. В качестве источника данных для узлов сети также можно выбрать совместимый виджет или панель. Это поле поддерживает автодополнение, поэтому при вводе имени узла сети будет предложен выпадающий список подходящих узлов сети. Если узлы сети не указаны, будут отображаться все узлы сети. Этот параметр недоступен при настройке виджета на панели шаблона.

Проблема	Вы можете ограничить количество отображаемых узлов сети с проблемами по имени проблемы. Если вы введете здесь строку, будут отображаться только те узлы сети, у которых есть проблемы, имя которых содержит введенную строку. Макросы не раскрываются.
Важность	Отметьте уровни важности проблем, чтобы отфильтровать проблемы, отображаемые в виджете. Если не отмечен ни один уровень важности, будут отображаться все проблемы.
Теги проблем	Укажите теги проблем, чтобы ограничить количество проблем, отображаемых 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: **Существует** - включить указанные имена тегов; **Равно** - включить указанные имена тегов и значения (с учетом регистра); **Содержит** - включить указанные имена тегов, у которых значения 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, у которых значения содержат введенную строку (поиск подстроки, без учета 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.
Показывать подавленные проблемы	Установите этот флажок, чтобы отображать проблемы, которые в противном случае были бы подавлены (не показаны) из-за обслуживания узла сети.
Скрывать группы без проблем	Отметьте параметр *Скрывать группы без проблем*, чтобы скрыть в виджете данные групп узлов сети без проблем. Этот параметр недоступен при настройке виджета на панели шаблона.
Отображение проблем	Отображать количество проблем как: **Все** - будет отображаться полное количество проблем; **Раздельно** - количество неподтвержденных проблем будет отображаться отдельно как число от общего количества проблем; **Только неподтвержденные** - будет отображаться только количество неподтвержденных проблем.

23 Проблемы

Обзор

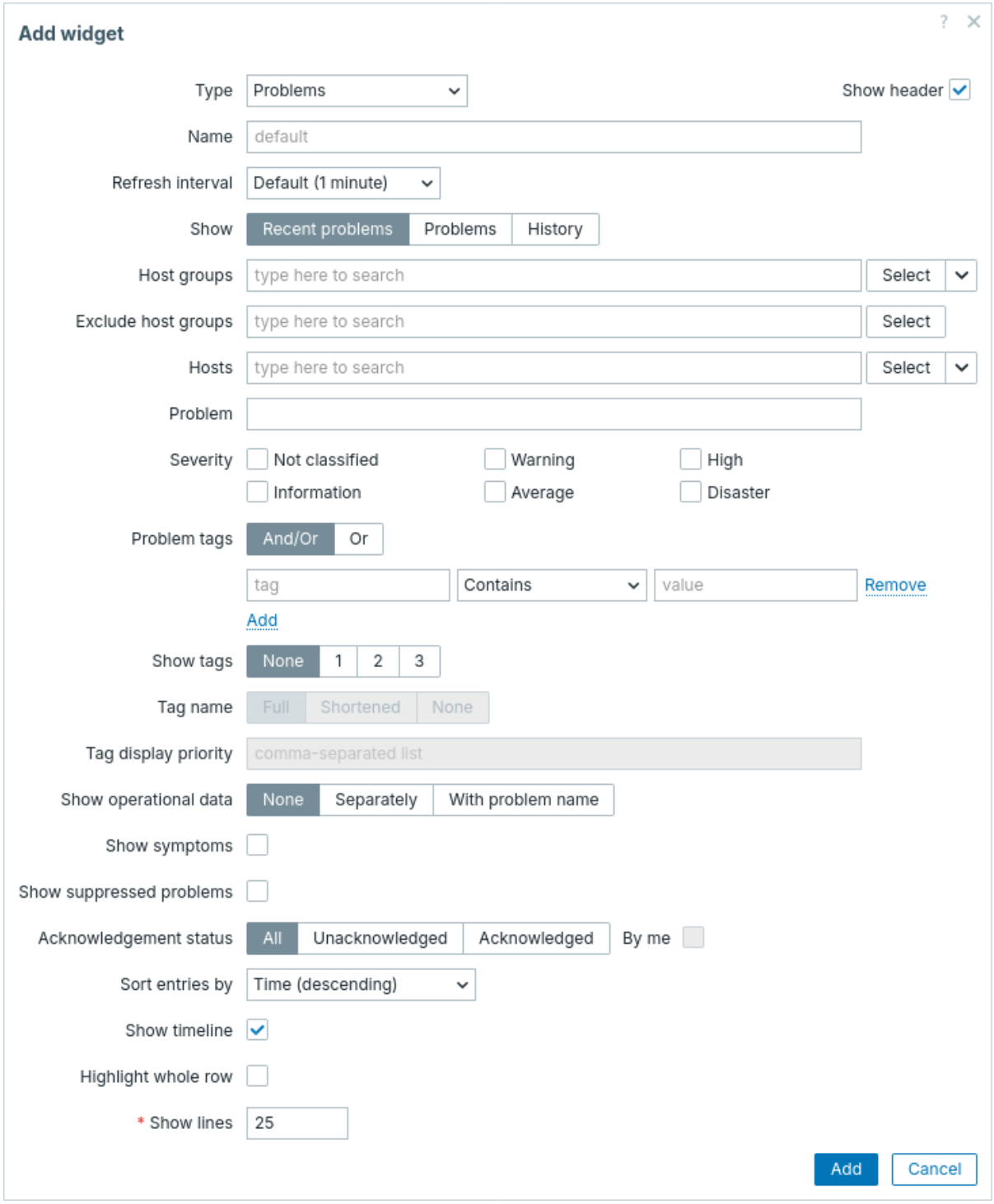
Виджет *Проблемы* отображает текущие проблемы, отфильтрованные по различным параметрам (группы узлов сети, узлы сети, имена проблем и т. д.), предоставляя вам наглядное представление о том, что требует внимания.



Он отображает те же данные, что и *Мониторинг* > *Проблемы*, и может показывать до 1000 записей.

Настройка

Для настройки выберите тип *Проблемы*:



Вы можете ограничить количество проблем, отображаемых в виджете, различными способами — по статусу проблемы, имени проблемы, важности, группе узлов сети, узлу сети, тегу события, статусу подтверждения и т. д.

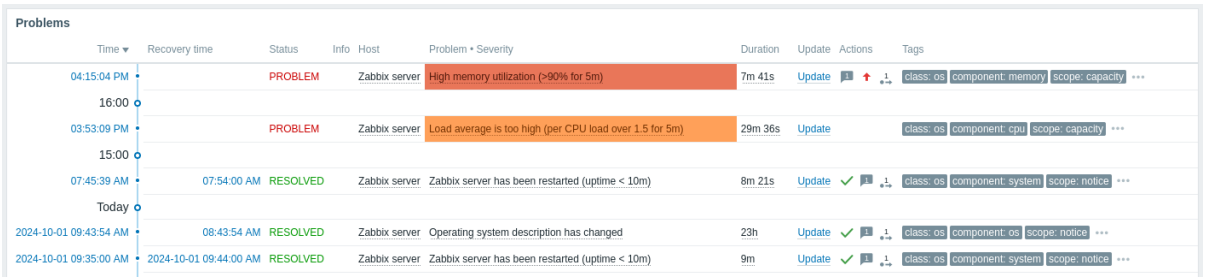
В дополнение к параметрам, которые являются общими для всех виджетов, можно задать следующие специальные параметры:

Показывать	Фильтр по статусу проблемы: **Недавние проблемы** — отображаются неустраненные и недавно устраненные проблемы (по умолчанию); **Проблемы** — отображаются неустраненные проблемы; **История** — отображается история всех событий.

Группы узлов сети	Выберите группы узлов сети, проблемы которых будут отображаться в виджете. В качестве источника данных для групп узлов сети также можно выбрать совместимый виджет. Это поле поддерживает автодополнение, поэтому при вводе имени группы будет предложен выпадающий список подходящих групп. Указание родительской группы узлов сети неявно выбирает все вложенные группы узлов сети. В виджете будут отображаться проблемы из этих групп узлов сети; если ни одна группа узлов сети не указана, будут отображаться проблемы из всех групп узлов сети. Этот параметр недоступен при настройке виджета на панели шаблона.
Исключить группы узлов сети	Выберите группы узлов сети, проблемы которых нужно скрыть в виджете. Это поле поддерживает автодополнение, поэтому при вводе имени группы будет предложен 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 группу A, а узлы сети 002, 003 — также в группу B. Если одновременно выбрать *показывать* группу A и *исключить* группу B, в виджете будут отображаться только проблемы узла сети 001. Этот параметр недоступен при настройке виджета на панели шаблона.
Узлы сети	Выберите узлы сети, проблемы которых будут отображаться в виджете. В качестве источника данных для узлов сети также можно выбрать совместимый виджет или панель. Это поле поддерживает автодополнение, поэтому при вводе имени узла сети будет предложен выпадающий список подходящих узлов сети. Если узлы сети не указаны, будут отображаться проблемы всех узлов сети. Этот параметр недоступен при настройке виджета на панели шаблона.
Проблема	Вы можете ограничить количество отображаемых проблем по их имени. Если ввести здесь строку, будут отображаться только те проблемы, имя которых содержит введенную строку. Макросы не раскрываются.
Важность	Отметьте уровни важности проблем для фильтрации проблем, отображаемых в виджете. Если не отмечен ни один уровень важности, будут отображаться все проблемы.
Теги проблем	Укажите теги проблем, чтобы ограничить количество проблем, отображаемых 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 несколько операторов: **Существует** — включить указанные имена тегов; **Равно** — включить указанные имена тегов и значения (с учетом регистра); **Содержит** — включить указанные имена тегов, у которых значения 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, у которых значения содержат введенную строку (поиск подстроки, без учета 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. При фильтрации указанные здесь теги будут отображаться у проблемы первыми, если это не переопределено списком *Приоритет отображения тегов* (см. ниже).
Показывать теги	Выберите количество отображаемых тегов: **Нет** — столбец *Теги* отсутствует; **1** — столбец *Теги* содержит один тег; **2** — столбец *Теги* содержит два тега; **3** — столбец *Теги* содержит три тега. Чтобы увидеть все теги проблемы, наведите указатель мыши на значок с тремя точками.

Имя тега	Выберите режим отображения имени тега: **Полное** — имена тегов и значения отображаются полностью; **Сокращенное** — имена тегов сокращаются до 3 символов, а значения отображаются полностью; **Нет** — отображаются только значения тегов, без имен.
Приоритет отображения тегов	Введите приоритет отображения тегов для проблемы в виде списка тегов, разделенных запятыми. Следует использовать только имена тегов, без значений. Пример: `Services,Applications,Application` Теги из этого списка всегда будут отображаться первыми, переопределяя естественный алфавитный порядок.
Показывать оперативные данные	Выберите режим отображения оперативных данных: **Нет** — оперативные данные не отображаются; **Отдельно** — оперативные данные отображаются в отдельном столбце; **С именем проблемы** — оперативные данные добавляются к имени проблемы в скобках.
Показывать симптомы	Установите этот флажок, чтобы отображать на отдельной строке проблемы, классифицированные как симптомы.
Показывать подавленные проблемы	Установите этот флажок, чтобы отображать проблемы, которые в противном случае были бы подавлены (не показаны) из-за обслуживания узла сети или отдельного подавления проблем.
Статус подтверждения	Фильтр для отображения всех проблем, только неподтвержденных проблем или только подтвержденных проблем. Установите дополнительный флажок, чтобы исключить те проблемы, которые когда-либо были подтверждены вами.
Сортировать записи по	Сортировка записей по: **Время** (по убыванию или по возрастанию); **Важность** (по убыванию или по возрастанию); **Имя проблемы** (по убыванию или по возрастанию); **Узел сети** (по убыванию или по возрастанию). Сортировка записей по **Узлу сети** (по убыванию или по возрастанию) недоступна при настройке виджета на панели шаблона.
Показывать временную шкалу	Установите этот флажок, чтобы отображать визуальную временную шкалу.
Подсвечивать всю строку	Установите этот флажок, чтобы подсвечивать всю строку для неустраненных проблем. Для подсветки используется цвет важности проблемы. *Подсвечивать всю строку* недоступно в темах с высокой контрастностью.
Показывать строки	Укажите количество строк с проблемами для отображения.

Использование виджета

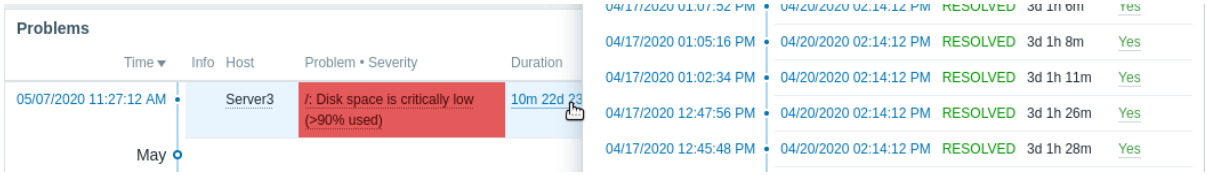


Виджет проблем предоставляет быстрый доступ к дополнительной информации:

- Нажмите на дату проблемы (в столбце Time) или дату восстановления (в столбце Recovery Time), чтобы просмотреть сведения о событии.
- Если поле Info не пустое, наведите указатель мыши на отображаемый значок, чтобы просмотреть дополнительные сведения.
- Нажмите на имя узла сети, чтобы открыть меню узла сети.
- Нажмите на имя проблемы, чтобы открыть меню события.
- Наведите указатель мыши на длительность проблемы или нажмите на неё, чтобы просмотреть всплывающее окно события проблемы.
- Нажатие кнопки Update открывает окно Update problem.
- Наведите указатель мыши на серый значок стрелки в столбце Actions или нажмите на него, чтобы просмотреть список выполненных действий.

Всплывающее окно события проблемы

Всплывающее окно события проблемы содержит список событий проблемы для этого триггера, а также, если они определены, описание триггера и URL-адрес, по которому можно перейти.



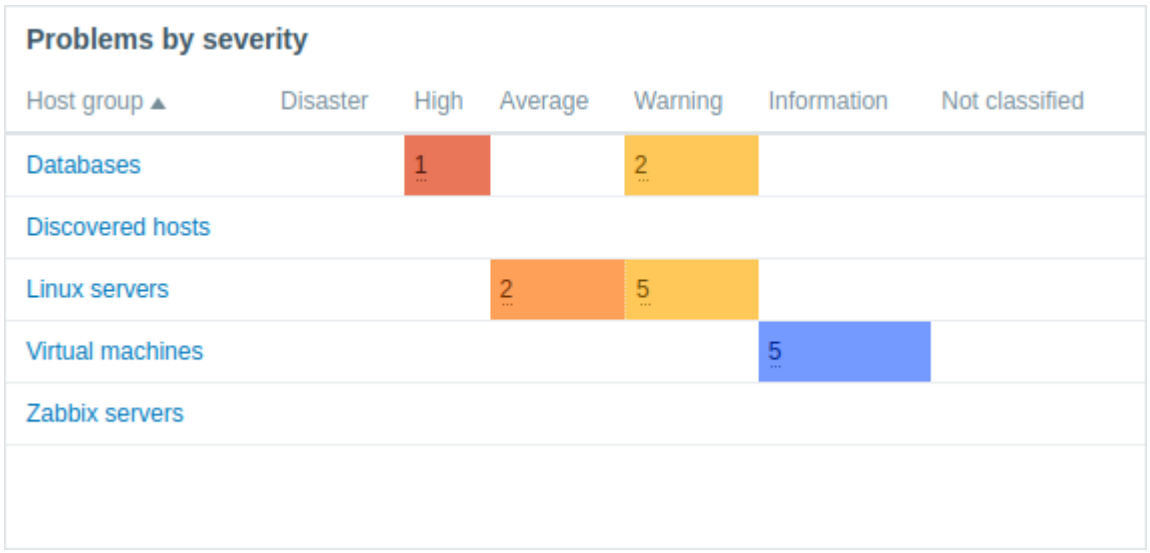
Чтобы открыть всплывающее окно события проблемы:

- Наведите указатель мыши на длительность проблемы в столбце *Duration* виджета *Problems*. Всплывающее окно исчезнет, как только вы уберёте указатель мыши с длительности.
- Нажмите на длительность в столбце *Duration* виджета *Problems*. Всплывающее окно исчезнет только после повторного нажатия на длительность.

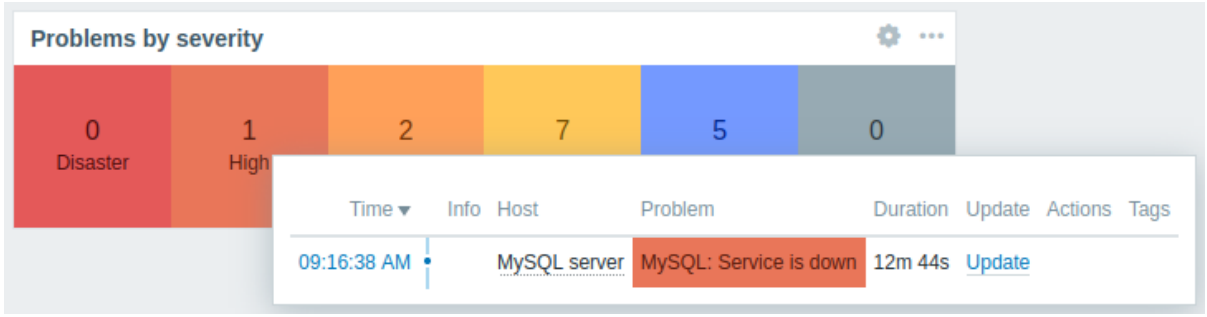
24 Проблемы по серьезности

Обзор

Виджет *Проблемы по важности* отображает количество первопричин проблем, сгруппированных по важности. Он позволяет фильтровать проблемы по различным параметрам (группы узлов сети, узлы сети, имена проблем и т. д.).

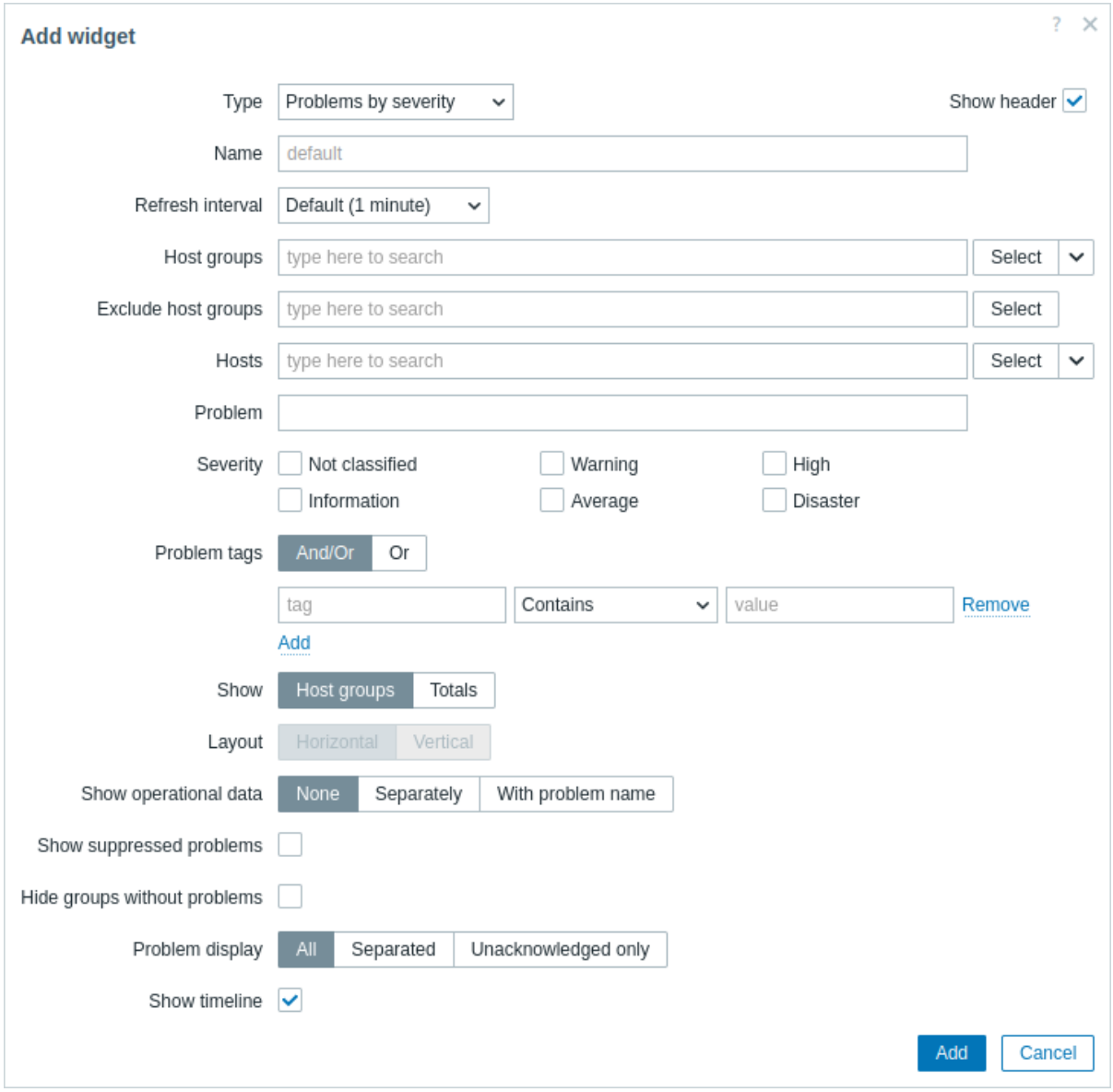


При наведении курсора на количество проблем отображается список связанных проблем:



Конфигурация

Для настройки выберите *Проблемы по важности* в качестве типа:



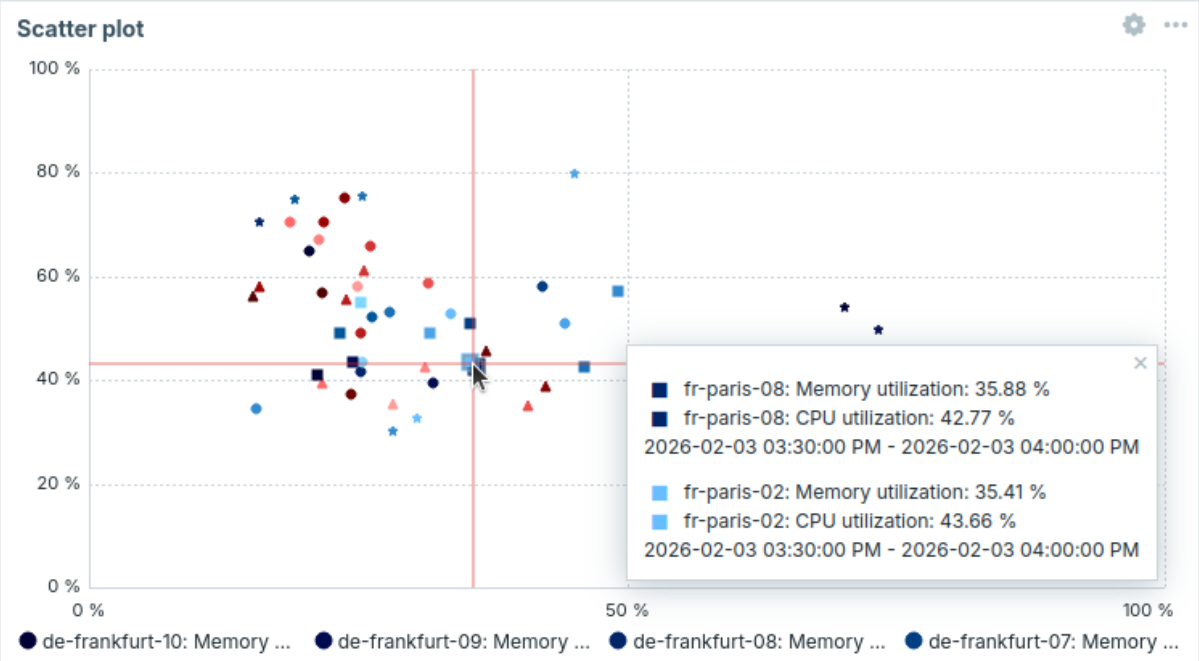
В дополнение к параметрам, которые являются общими для всех виджетов, вы можете задать следующие специальные параметры:

Группы узлов сети	Выберите группы узлов сети для отображения в виджете. В качестве источника данных для групп узлов сети также можно выбрать совместимый виджет. Это поле поддерживает автодополнение, поэтому при вводе имени группы будет предложен выпадающий список подходящих групп. Указание родительской группы узлов сети неявно выбирает все вложенные группы узлов сети. В виджете будут отображаться данные узлов сети из этих групп узлов сети; если группы узлов сети не указаны, будут отображаться все группы узлов сети. Этот параметр недоступен при настройке виджета на дашборде шаблона.
Исключить группы узлов сети	Выберите группы узлов сети, которые нужно скрыть в виджете. Это поле поддерживает автодополнение, поэтому при вводе имени группы будет предложен выпадающий список подходящих групп. Указание родительской группы узлов сети неявно выбирает все вложенные группы узлов сети. Данные узлов сети из этих групп узлов сети не будут отображаться в виджете. Например, узлы сети 001, 002, 003 могут входить в Группу A, а узлы сети 002, 003 также в Группу B. Если одновременно выбрать *показ* Группы A и *исключение* Группы B, на дашборде будут отображаться только данные узла сети 001. Этот параметр недоступен при настройке виджета на дашборде шаблона.

Узлы сети	Выберите узлы сети для отображения в виджете. В качестве источника данных для узлов сети также можно выбрать совместимый виджет или дашборд. Это поле поддерживает автодополнение, поэтому при вводе имени узла сети будет предложен выпадающий список подходящих узлов сети. Если узлы сети не указаны, будут отображаться все узлы сети. Этот параметр недоступен при настройке виджета на дашборде шаблона.
Проблема	Вы можете ограничить количество проблемных узлов сети, отображаемых по имени проблемы. Если вы введете здесь строку, будут отображаться только те узлы сети, у которых есть проблемы, имя которых содержит введенную строку. Макросы не раскрываются.
Важность	Отметьте важности проблем, чтобы отфильтровать проблемы, отображаемые в виджете. Если не отмечена ни одна важность, будут отображаться все проблемы.
Теги проблем	Укажите теги проблем, чтобы ограничить количество проблем, отображаемых 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 несколько операторов: **Существует** - включить указанные имена тегов; **Равно** - включить указанные имена тегов и значения (с учетом регистра); **Содержит** - включить указанные имена тегов, у которых значения 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, у которых значения содержат введенную строку (поиск подстроки, без учета регистра). Для условий доступны два типа вычисления: **И/Или** - должны выполняться все условия, при этом условия с одинаковым именем тега будут сгруппированы по условию *Или*; **Или** - достаточно выполнения одного условия.
Показывать	Выберите вариант отображения: **Группы узлов сети** - отображать проблемы по группам узлов сети; **Итоги** - отображать общее количество проблем для всех выбранных групп узлов сети в цветных блоках, соответствующих важности проблемы. Этот параметр недоступен при настройке виджета на дашборде шаблона, и будет отображаться только общее количество проблем.
Макет	Выберите вариант макета: **Горизонтально** - цветные блоки с итогами будут отображаться горизонтально; **Вертикально** - цветные блоки с итогами будут отображаться вертикально. Этот параметр доступен, если для *Показывать* установлено значение ”Итоги”.
Показывать оперативные данные	Отображать оперативные данные (см. описание *Оперативные данные* в разделе *Мониторинг* > *Проблемы*) как: **Нет** - оперативные данные не будут отображаться **Отдельно** - оперативные данные будут отображаться отдельной строкой **С именем проблемы** - оперативные данные будут отображаться вместе с именем проблемы.
Показывать подавленные проблемы	Установите этот флажок, чтобы отображать проблемы, которые в противном случае были бы подавлены (не показаны) из-за обслуживания узла сети.
Скрывать группы без проблем	Отметьте параметр *Скрывать группы без проблем*, чтобы скрыть в виджете данные групп узлов сети без проблем. Этот параметр недоступен при настройке виджета на дашборде шаблона.
Отображение проблем	Отображать количество проблем как: **Все** - будет отображаться полное количество проблем; **Раздельно** - количество неподтвержденных проблем будет отображаться отдельно как число от общего количества проблем; **Только неподтвержденные** - будет отображаться только количество неподтвержденных проблем.
Показывать временную шкалу	Установите этот флажок, чтобы отображать визуальную временную шкалу.

Обзор

Виджет *Точечная диаграмма* 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.



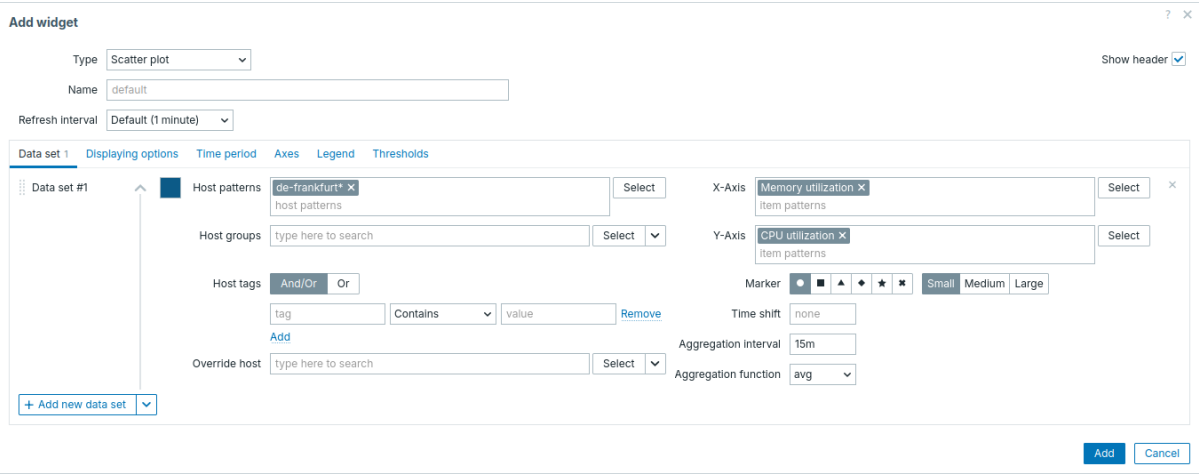
При наведении курсора на точку данных отображается всплывающая подсказка со списком элементов данных из этой точки и других точек в радиусе 6px; сортировка выполняется сначала по значениям оси X точечной диаграммы в порядке возрастания, а затем по значениям оси Y в порядке убывания.

При выборе элемента данных во всплывающей подсказке его данные передаются другим виджетам.

Информацию, отображаемую в виджете *Точечная диаграмма*, можно скачать как изображение PNG, выбрав опцию *Download image* в меню виджета.

Настройка

Чтобы настроить, выберите *Scatter plot* в качестве типа:



Набор данных

Вкладка **Набор данных** позволяет выбирать данные для диаграммы рассеяния, добавляя наборы данных. Можно добавить два типа наборов данных:

- *Шаблоны элементов данных* — узлы сети с совпадающими элементами данных отображаются как точки данных. Можно выбрать один базовый цвет или строку палитры, чтобы назначить разные цвета каждому узлу сети с совпавшими элементами данных.

- *Список элементов данных* — выбранные элементы данных отображаются как точки данных, где каждая точка представляет пару элементов данных (X и Y). Для каждого набора данных можно отдельно выбрать цвет с помощью палитры.

Точки данных создаются только для тех пар узел сети/элемент данных, у которых есть хотя бы одно значение на обеих осях.

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 диаграмме рассеяния; может быть отображено до 100 элементов данных (50 на каждой оси для 50 узлов сети суммарно). 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 интерпретируется всегда, поэтому невозможно, например, добавить отдельно элемент данных с именем *item** при наличии других совпадающих элементов данных (например, *item2*, *item3*). Указание шаблонов узлов сети и элементов данных обязательно для наборов данных ”Шаблоны элементов данных”. См. также: Подробности настройки набора данных. Для набора данных **Список элементов данных**: Выберите элементы данных для диаграммы рассеяния, нажав кнопку *Добавить элемент данных*. Также можно выбрать совместимые виджеты как источник данных для элементов данных, нажав кнопку *Добавить виджет*. Указание элементов данных или виджетов обязательно для наборов данных ”Список элементов данных”. См. также: Подробности настройки набора данных. Обратите внимание, что допускаются только числовые типы элементов данных. При настройке виджета на дашборде 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. Либо выберите совместимый виджет как источник данных для групп узлов сети. Это поле поддерживает автодополнение, поэтому при вводе имени группы будет предложен выпадающий список совпадающих 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 доступен только для наборов данных *Шаблоны элементов данных* и недоступен при настройке виджета на дашборде шаблона.

Теги узлов сети	Укажите теги для фильтрации узлов сети, отображаемых 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 несколько операторов: **Существует** — включить указанные имена тегов; **Равно** — включить указанные имена и значения тегов (с учетом регистра); **Содержит** — включить указанные имена тегов, у которых значения тегов содержат введенную строку (поиск подстроки, без учета регистра); **Не существует** — исключить указанные имена тегов; **Не равно** — исключить указанные имена и значения тегов (с учетом регистра); **Не содержит** — исключить указанные имена тегов, у которых значения тегов содержат введенную строку (поиск подстроки, без учета регистра). Для условий доступны два типа вычисления: **И/Или** — должны выполняться все условия, при этом условия с одинаковым именем тега будут сгруппированы по условию Или; **Или** — достаточно выполнения одного условия. Этот параметр доступен только для наборов данных *Шаблоны элементов данных* и недоступен при настройке виджета на дашборде шаблона.
Переопределить узел сети	Выберите совместимый виджет или селектор узла сети дашборда как источник данных для узлов сети. Этот параметр недоступен при настройке виджета на дашборде шаблона.
Маркер	Выберите тип и размер маркера для точек данных узлов сети в наборе данных.
Сдвиг времени	При необходимости укажите сдвиг времени. В этом поле можно использовать суффиксы времени. Допускаются отрицательные значения.
Функция агрегации	Укажите, какую функцию агрегации использовать: **min** — отобразить наименьшее значение; **max** — отобразить наибольшее значение; **avg** — отобразить среднее значение; **sum** — отобразить сумму значений; **count** — отобразить количество значений (если хотя бы одно значение существует на обеих осях); **first** — отобразить первое значение; **last** — отобразить последнее значение; **none** — отобразить все значения (без агрегации). Агрегация позволяет отображать агрегированное значение за выбранный интервал (5 минут, час, день) вместо всех значений. См. также: Агрегация в графиках.
Интервал агрегации	Укажите интервал для агрегации значений. В этом поле можно использовать суффиксы времени. Числовое значение без суффикса будет считаться секундами. Обратите внимание, что если виджет настроен на отображение исторических данных на основе трендов (*Выбор исторических данных* установлен в *Тренды* или *Авто*), рекомендуется использовать интервал агрегации, кратный 1 часу (например, 3600, 60m, 1h, 3h и т. д.). Тренды хранят почасовые агрегированные значения, поэтому использование интервала агрегации, не кратного 1 часу (например, 100s, 7min, 15min, 90min и т. д.), может привести к результатам, которые трудно интерпретировать. Указание интервала агрегации обязательно. Интервал агрегации определяет временной интервал и координаты точки данных; например, если элементы данных оси X имеют 4 значения, а элементы данных оси Y — 1 значение, координаты отражают агрегацию 4 значений по X и 1 значения по Y в указанном интервале.

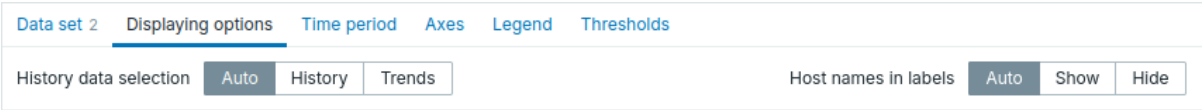
Подробности настройки набора данных

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

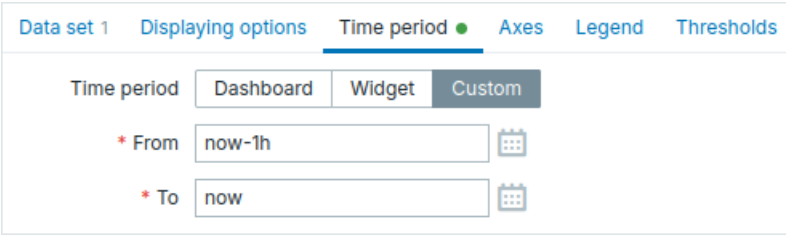
Вкладка **Параметры отображения** позволяет определить выборку исторических данных:



Выбор исторических данных	Укажите источник данных для точечной диаграммы: **Авто** — данные выбираются в соответствии с классическим алгоритмом графика (по умолчанию). **История** — данные из истории. **Тренды** — данные из трендов.
Имена узлов сети в метках	Выберите, отображать ли имена узлов сети в легенде графика и во всплывающей подсказке, которая появляется при наведении курсора 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 дашборде шаблона.

Период времени

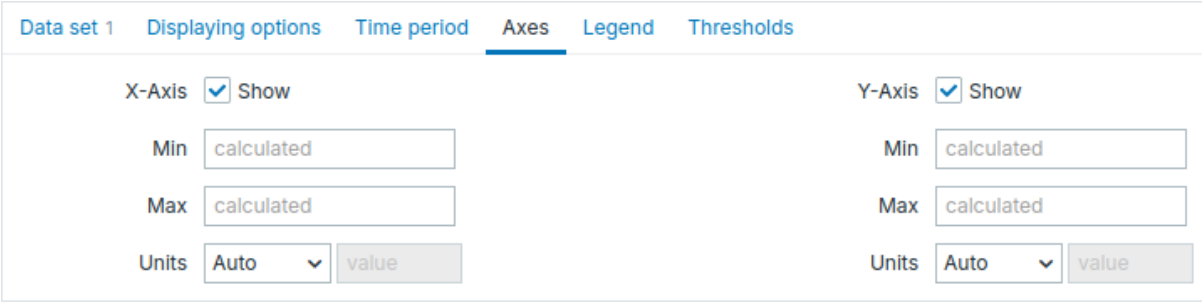
Вкладка **Период времени** позволяет задать период времени, за который будут отображаться данные на диаграмме рассеяния:



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 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 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".

Оси

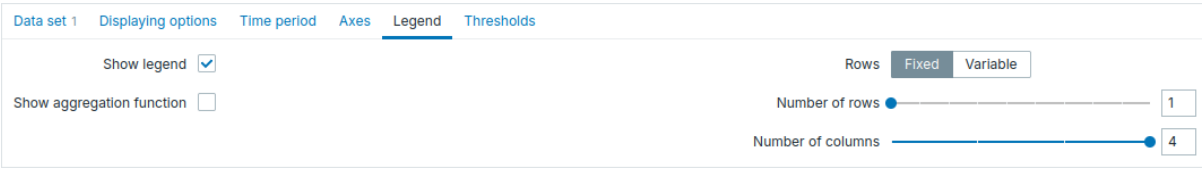
Вкладка **Оси** позволяет настроить отображение осей:



Ось X/Ось Y	Снимите этот флажок, чтобы скрыть ось X/ось Y (по умолчанию установлен).
Мин	Задайте минимальное значение соответствующей оси.
Макс	Задайте максимальное значение соответствующей оси.
Единицы измерения	Выберите единицу измерения для значений осей диаграммы рассеяния из выпадающего списка: **Авто** — значения осей отображаются с использованием единицы измерения первого элемента данных в наборе данных; **Статически** — значения осей отображаются с использованием единицы измерения, указанной в поле ввода *значение*; если поле оставлено пустым, отображаются только числовые значения.

Легенда

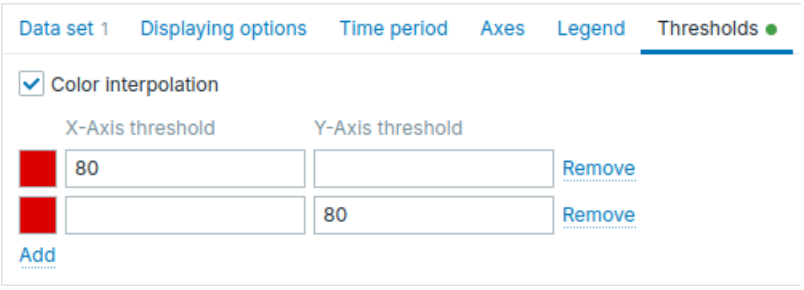
Вкладка **Легенда** позволяет настроить легенду точечной диаграммы:



Показывать легенду	Снимите этот флажок, чтобы скрыть легенду на точечной диаграмме (по умолчанию установлен).
Показывать функцию агрегации	Установите этот флажок, чтобы отображать функцию агрегации в легенде.
Строки	Выберите режим отображения строк легенды: **Фиксированное** — количество отображаемых строк определяется значением параметра *Количество строк*; **Переменное** — количество отображаемых строк определяется числом настроенных элементов данных, но не превышает значение параметра *Максимальное количество строк*.
Количество строк/ Максимальное количество строк	Если для параметра *Строки* установлено значение "Фиксированное", задайте количество отображаемых строк легенды (1-10). Если для параметра *Строки* установлено значение "Переменное", задайте максимальное количество отображаемых строк легенды (1-10).
Количество столбцов	Задайте количество отображаемых столбцов легенды (1-4). Этот параметр доступен, если флажок *Отображать min/avg/max* снят.

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 оси X/оси Y	Нажмите *Добавить*, чтобы добавить пороговое значение, выберите цвет порогового значения 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, когда будут выполнены оба условия. При сохранении список пороговых значений будет отсортирован по возрастанию. Поддерживаются суффиксы (например, ”1d”, ”2w”, ”4K”, ”8G”). Поддерживаются преобразования значений.

26 Отчёт SLA

Обзор

Виджет *Отчет SLA* отображает отчеты SLA, помогая вам контролировать производительность сервисов относительно заданных целевых показателей.

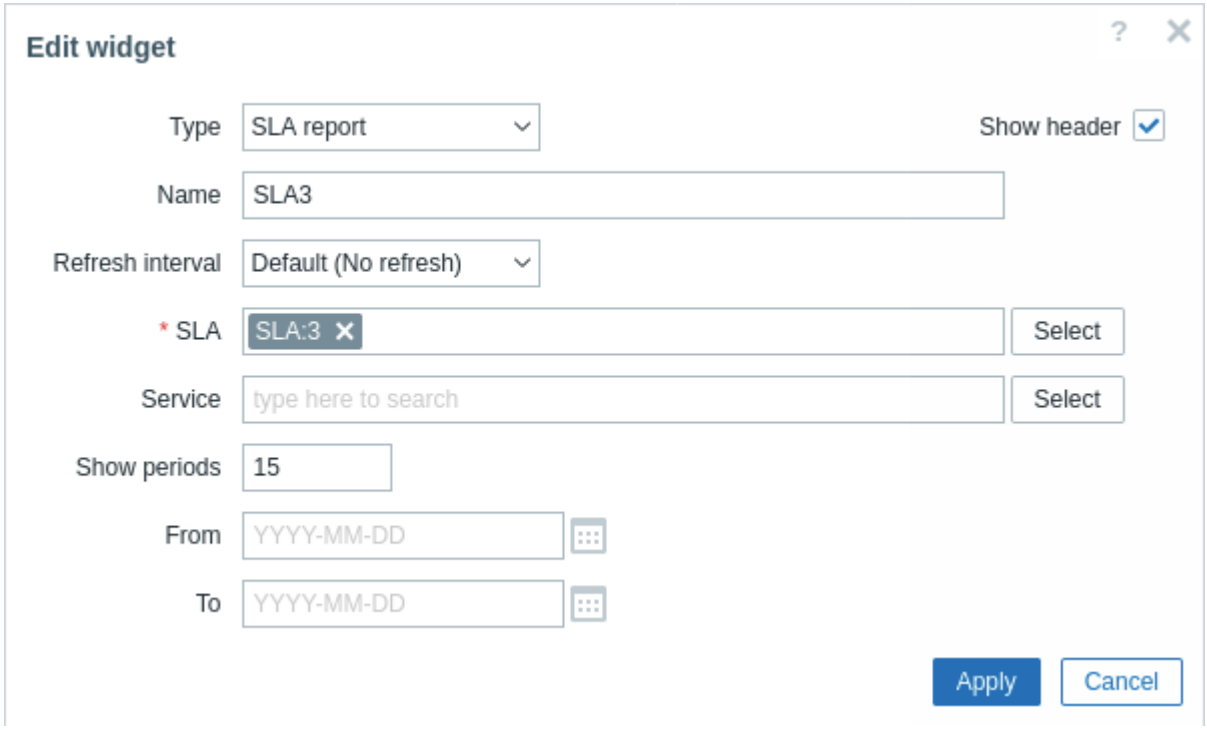
SLA report

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

Он отображает те же данные, что и *Сервисы* > *Отчет SLA*.

Настройка

Для настройки выберите *Отчет SLA* в качестве типа:



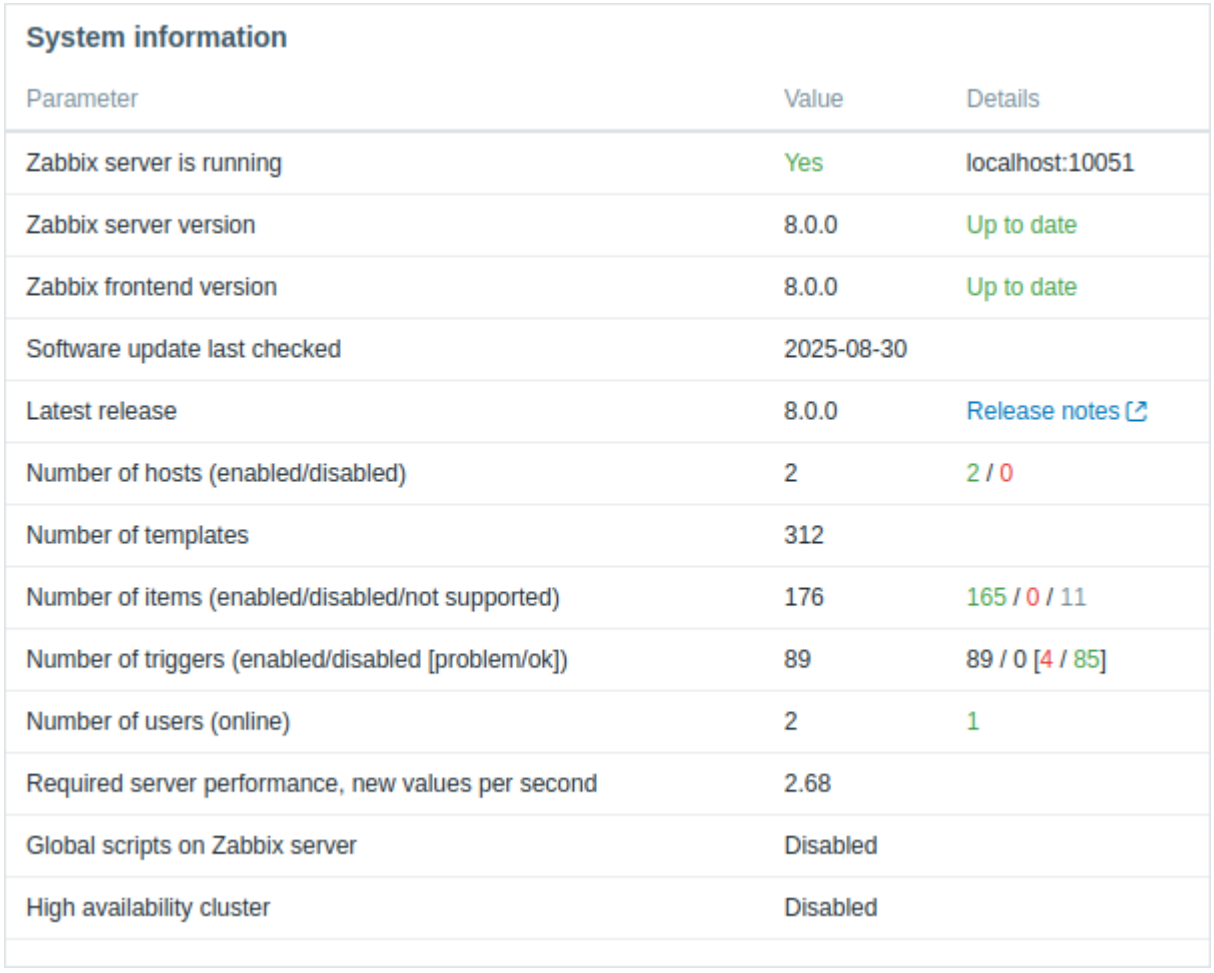
В дополнение к параметрам, общим для всех виджетов, вы можете задать следующие специальные параметры:

SLA	Выберите SLA для отчета.
Service	Выберите сервис для отчета.
Show periods	Укажите, сколько периодов будет отображаться в виджете (20 по умолчанию, 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.

27 Системная информация

Обзор

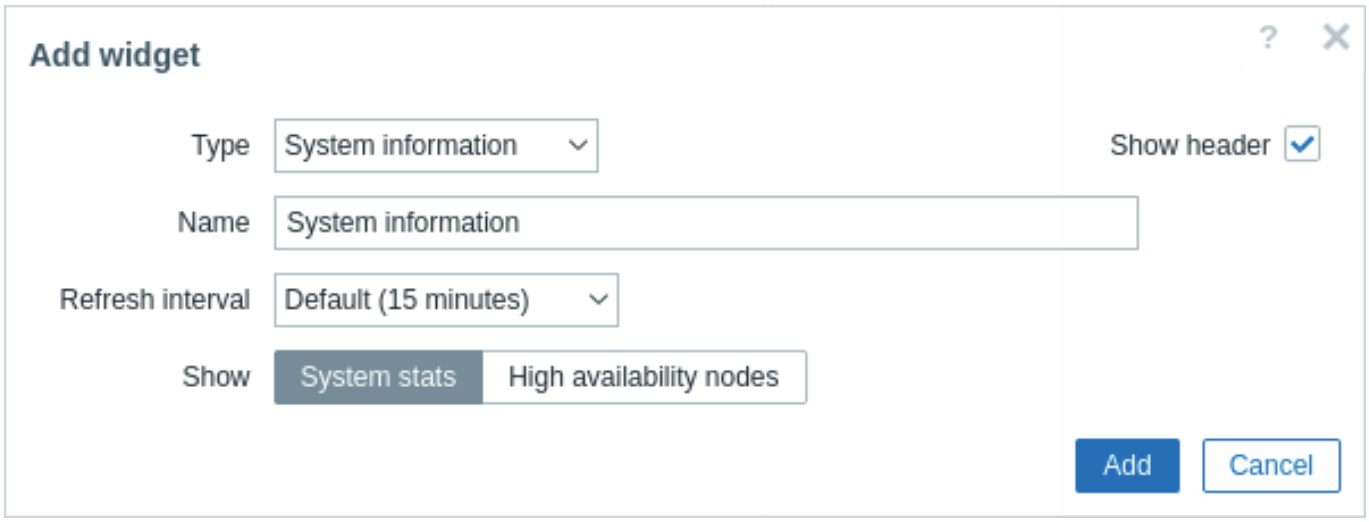
Виджет *Информация о системе* отображает сводку по ключевым данным сервера Zabbix и системы или сведения об узлах высокой доступности.



Он отображает те же данные, что и *Отчеты* > *Информация о системе*.

Настройка

Для настройки выберите *System information* в качестве типа:



В дополнение к параметрам, которые являются общими для всех виджетов, вы можете задать следующие специальные параметры:

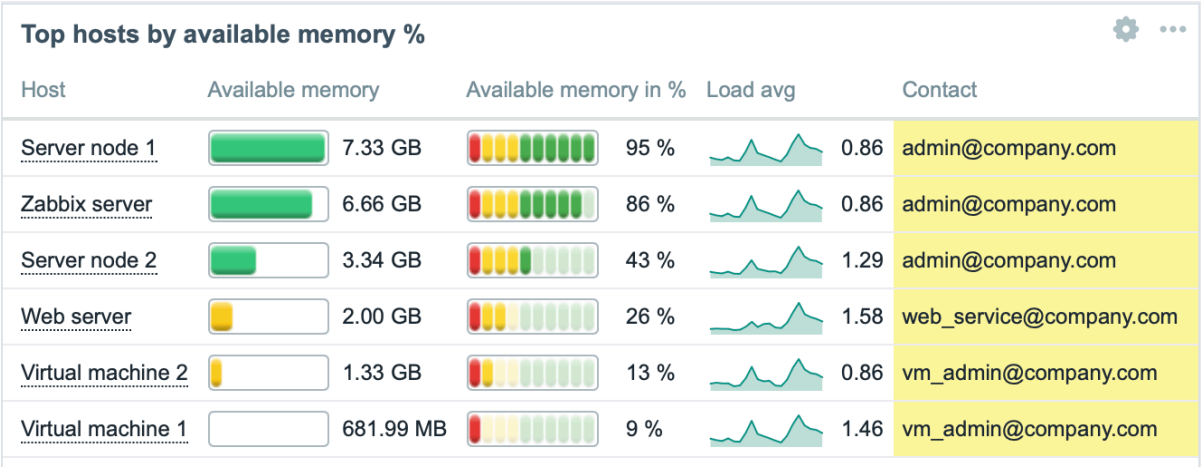
Show	Выберите, что отображать: **System stats** — отображать сводку по ключевым данным сервера Zabbix и системы; **High availability nodes** — отображать состояние узлов высокой доступности (если включен кластер высокой доступности).

Show software update check details	Установите этот флажок, чтобы отображать подробности проверки обновлений программного обеспечения Zabbix. Этот параметр доступен только в том случае, если проверка обновлений программного обеспечения включена в конфигурации сервера Zabbix и в поле *Show* выбрано значение ”System stats”.

28 Топ узлов сети

Обзор

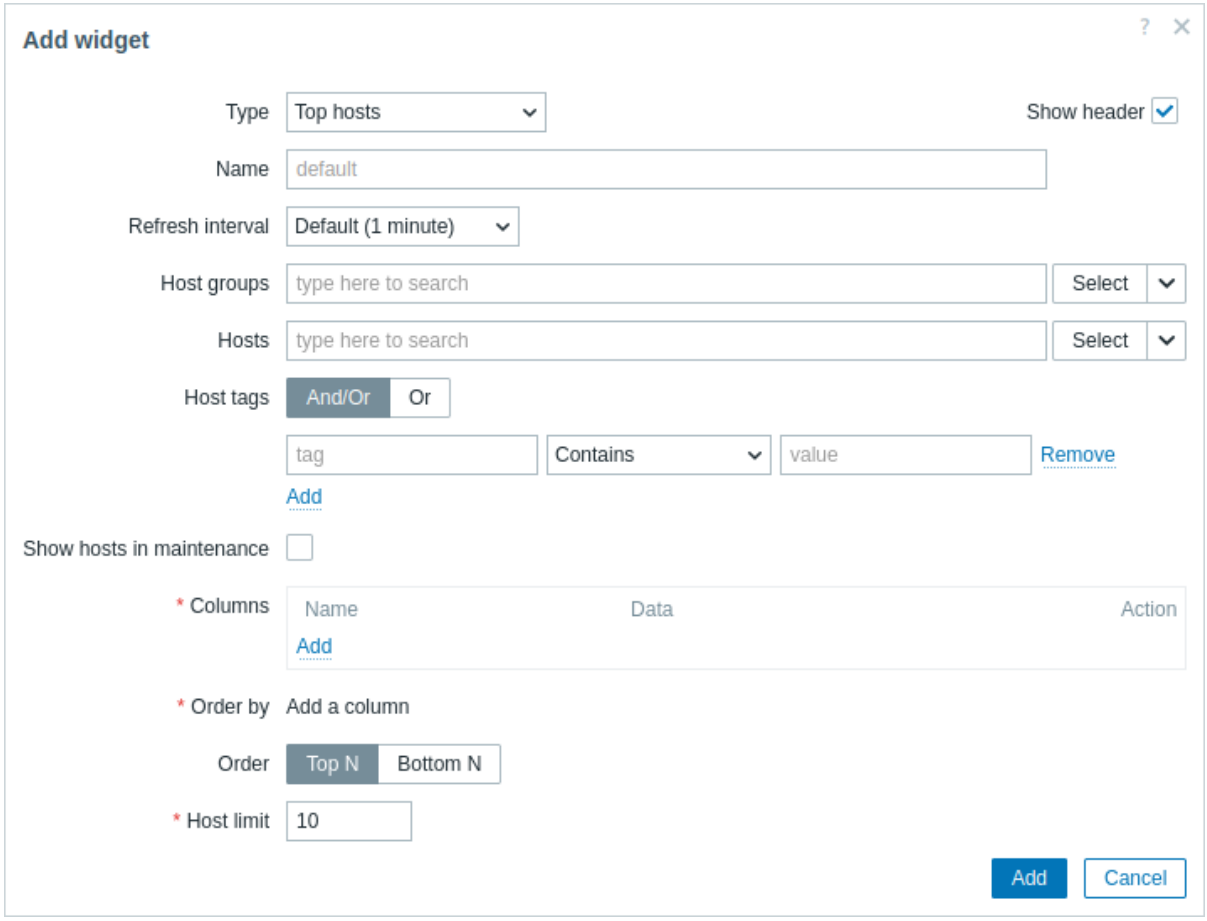
Виджет *Top hosts* отображает наибольшие/наименьшие значения для выбранных элементов данных с нескольких узлов сети.



Виджет поддерживает одновременное отображение до 1000 узлов сети.

Настройка

Для настройки выберите *Top hosts* 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 для отображения в виджете. 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 панели шаблона.
Узлы сети	Выберите узлы сети для отображения в виджете. Либо выберите совместимый виджет или панель в качестве источника данных для узлов сети. Это поле поддерживает автодополнение, поэтому при вводе имени узла сети будет предложен выпадающий список подходящих узлов сети. Этот параметр недоступен при настройке виджета на панели шаблона.
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 имен тегов всегда чувствительно к регистру. Для каждого условия доступны несколько операторов: **Существует** — включить указанные имена тегов; **Равно** — включить указанные имена и значения тегов (с учетом регистра); **Содержит** — включить указанные имена тегов, если значения тегов содержат введенную строку (поиск подстроки, без учета регистра); **Не существует** — исключить указанные имена тегов; **Не равно** — исключить указанные имена и значения тегов (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: **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 панели шаблона.

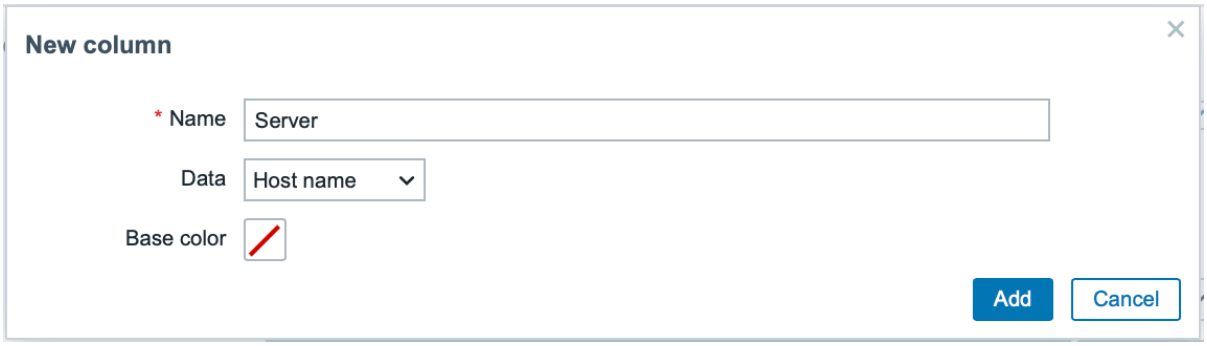
Показывать узлы сети на обслуживании	Установите этот флажок, чтобы также отображались узлы сети, находящиеся на обслуживании (в этом случае рядом с именем узла сети будет показан значок обслуживания). По умолчанию не отмечено.
Столбцы	Добавьте столбцы данных для отображения. Порядок столбцов определяет их отображение слева направо. Столбцы можно переупорядочить, перетаскивая их вверх и вниз за маркер перед именем столбца.
Сортировать по	Укажите столбец из заданного списка *Столбцы*, который будет использоваться для сортировки *Top N* или *Bottom N*.
Порядок	Укажите порядок строк: **Top N** — в порядке убывания согласно агрегированному значению *Сортировать по*; **Bottom N** — в порядке возрастания согласно агрегированному значению *Сортировать по*.
Лимит узлов сети	Количество строк узлов сети для отображения (1-1000). Этот параметр недоступен при настройке виджета на панели шаблона.

Настройка столбцов

Можно добавлять столбцы трех типов данных: имя узла сети, значение элемента данных или текст. Список доступных параметров столбца зависит от типа данных столбца, а для типа «значение 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.



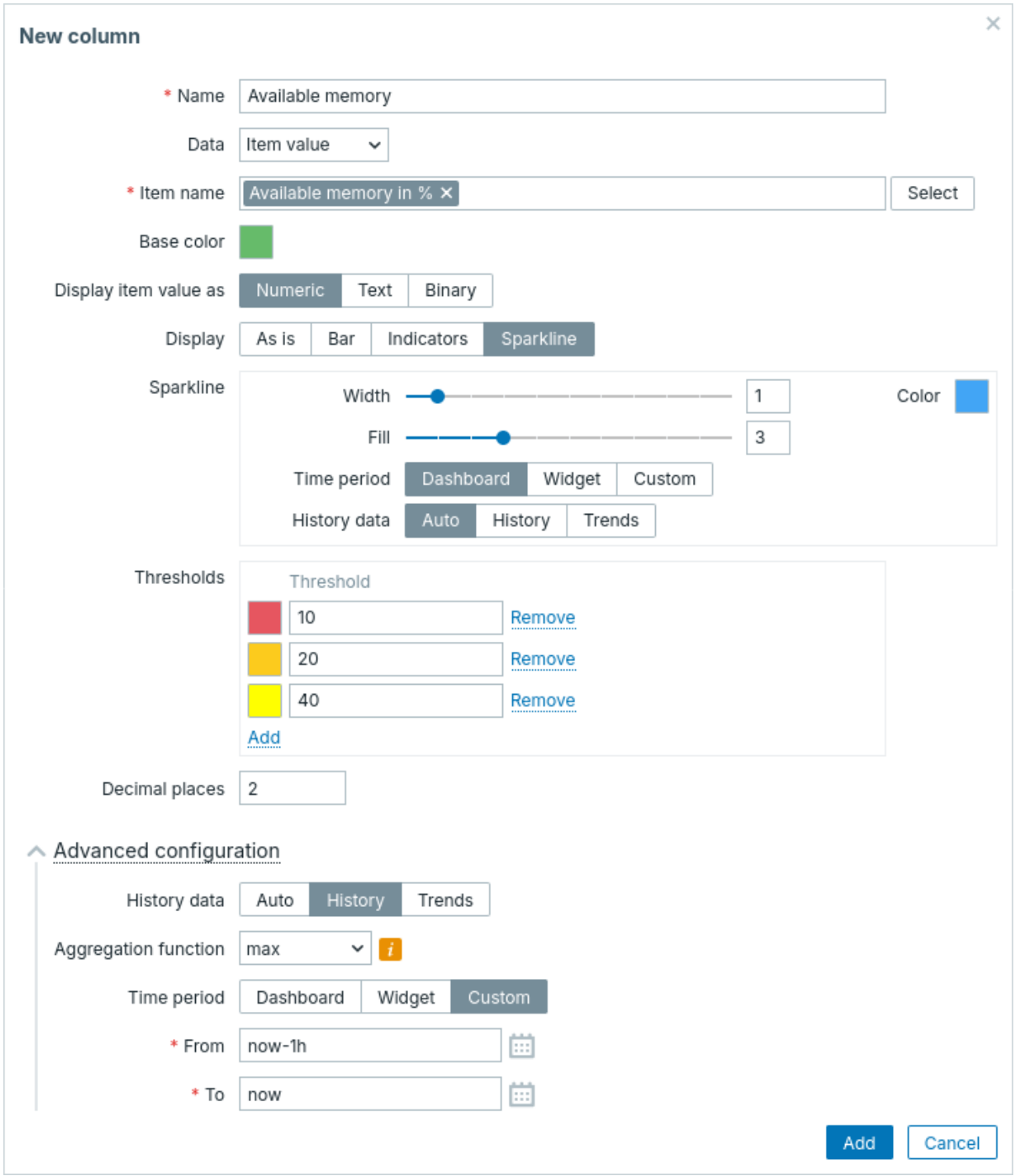
Поддерживаемые параметры:

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; он поддерживает несколько форматов и параметров 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 сопоставления и отображения элементов данных с одинаковым именем на всех выбранных узлах сети. Если у узла сети несколько таких элементов данных, виджет отображает элемент данных, чей ключ идет первым в алфавитном порядке (например, *CPU utilization* с ключом `proc.cpu.util` вместо `system.cpu.util`). При настройке виджета на панели шаблона можно выбирать только элементы данных, настроенные в шаблоне.
Base color	Цвет фона столбца; цвет заливки, если данные *Item value* отображаются в виде полосы/индикаторов. Цвет по умолчанию может быть переопределен 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, какие дополнительные параметры будут доступны. См. ниже список параметров для каждого формата.
Advanced configuration	
History data	Брать данные из истории или трендов: **Auto** — автоматический выбор; **History** — брать данные из истории; **Trends** — брать данные из трендов. Этот параметр доступен только для числовых значений элементов данных. Для текстовых и двоичных значений данные всегда будут браться из истории.
Aggregation function	Укажите, какую функцию агрегации использовать: **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; **not used** — отображать самое последнее значение (без агрегации). Агрегация позволяет отображать агрегированное значение за выбранный интервал (5 минут, час, день) вместо самого последнего значения. Для *min*, *max*, *avg* и *sum* могут отображаться только числовые данные. Для *count* нечисловые данные будут преобразованы в числовые.
Time period	Укажите период времени, который будет использоваться для агрегации значений. Выберите источник данных для периода времени: **Dashboard** — использовать селектор периода времени панели; **Widget** — использовать совместимый виджет (задается 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 курсора. Обратите внимание, что независимо от настройки *Time period* виджета, совместимые виджеты все равно могут использовать его как источник данных для периода времени. Этот параметр недоступен, если *Aggregation function* установлена в значение "not used".
Widget	Введите или выберите совместимый виджет в качестве источника данных для периода времени. Этот параметр доступен, если *Time period* установлено в 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 *Time period* установлено в 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 *Time period* установлено в значение "Custom".

Параметры числового значения элемента данных:

Display	Определяет, как должно отображаться значение: **As is** — как обычный текст; **Bar** — как сплошная полоса с цветной заливкой; **Indicators** — как сегментированная полоса с цветной заливкой; **Sparkline** — мини-график линии.
Thresholds	Укажите пороговые значения, при достижении которых должен изменяться цвет фона/заливки. При сохранении список будет отсортирован по возрастанию. Для спарклайнов пороги применяются только к последнему значению элемента данных.
Decimal places	Укажите, сколько знаков после запятой будет отображаться у значения.
Bar/Indicators	
Min	Минимальное значение.
Max	Максимальное значение.
Sparkline	
Width	Задайте толщину линии графика с помощью ползунка или вручную введя значение в диапазоне от 0 до 10.
Color	Выберите цвет линии и заливки.
Fill	Задайте уровень прозрачности цвета заливки с помощью ползунка или вручную введя значение в диапазоне от 0 до 10.
Time period	Укажите период времени, значения за который будут включены в график sparkline. Выберите источник данных для периода времени: **Dashboard** — использовать селектор периода времени панели **Widget** — использовать совместимый виджет (задается 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 курсора. Обратите внимание, что независимо от настройки *Time period* виджета, совместимые виджеты все равно 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 *Time period* установлено в 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 *Time period* установлено в 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 *Time period* установлено в значение ”Custom”.
History data	Брать данные из истории или трендов: **Auto** — автоматический выбор; **History** — брать данные из истории; **Trends** — брать данные из трендов.

Параметры текстового значения элемента данных:

Highlights	Укажите регулярные выражения, при совпадении с которыми должен изменяться цвет фона/заливки.

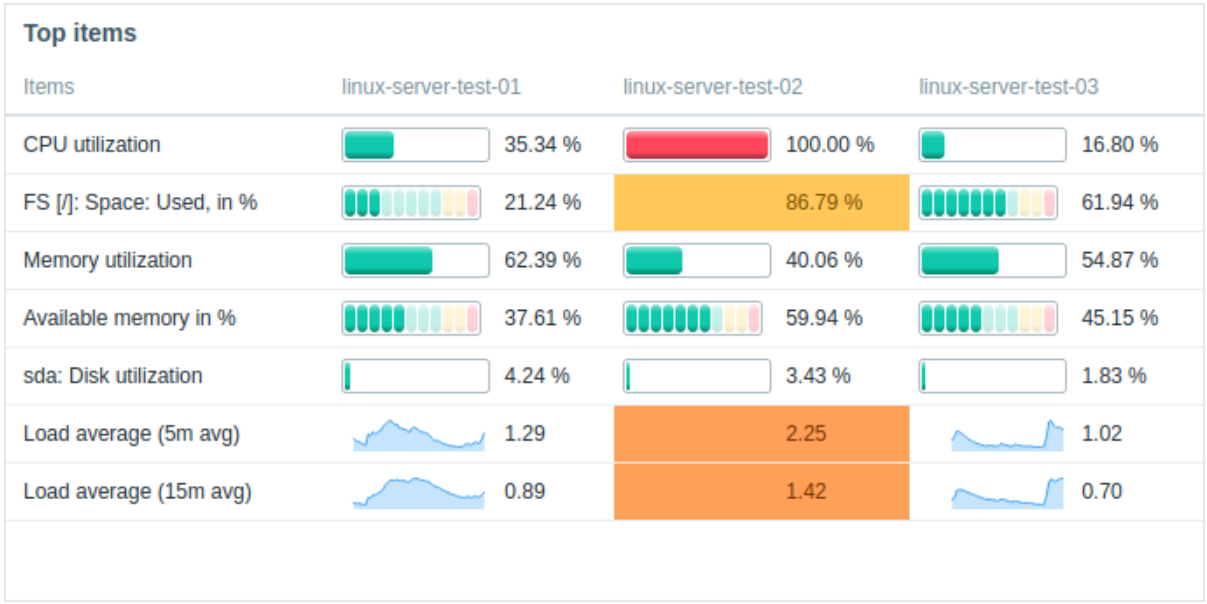
Параметры двоичного значения элемента данных:

Show thumbnail	Укажите, следует ли создавать и отображать миниатюру для изображения, содержащего двоичные данные, или отображать в столбце значения гиперссылку *Show*, ведущую к полноразмерному изображению.

29 Элементы данных верхнего уровня

Обзор

Виджет *Top items* отображает наибольшие/наименьшие значения для выбранных элементов данных, обеспечивая быстрый обзор их производительности.

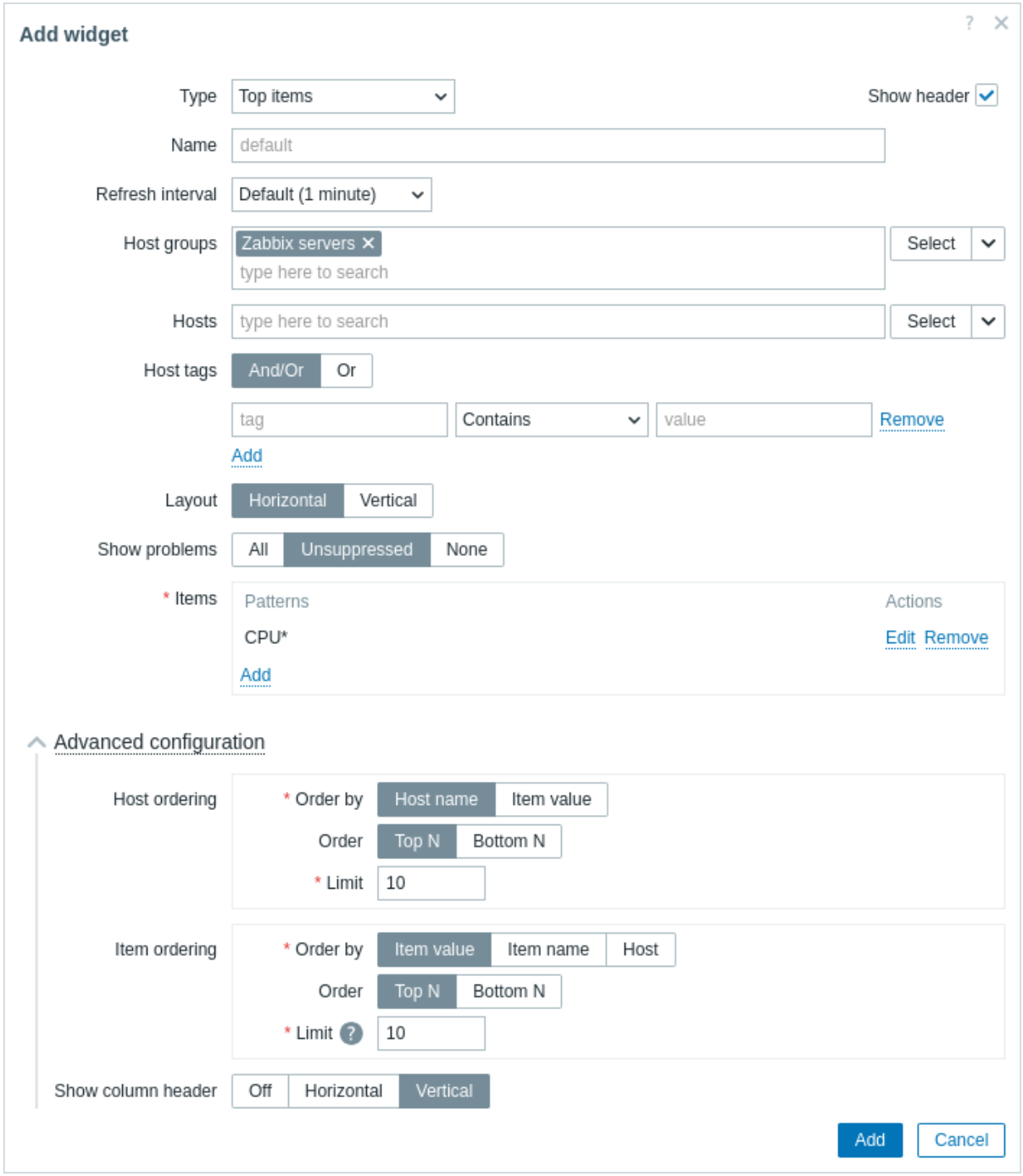


По умолчанию виджет отображает значения за последние 24 часа. Этот период можно изменить в разделе *Администрирование* > *Общие* > *GUI* (*Максимальный период отображения истории*).

Щелчок по значению элемента данных открывает меню элемента данных.

Настройка

Для настройки выберите *Top items* в качестве типа:



В дополнение к параметрам, которые являются общими для всех виджетов, вы можете задать следующие специальные параметры:

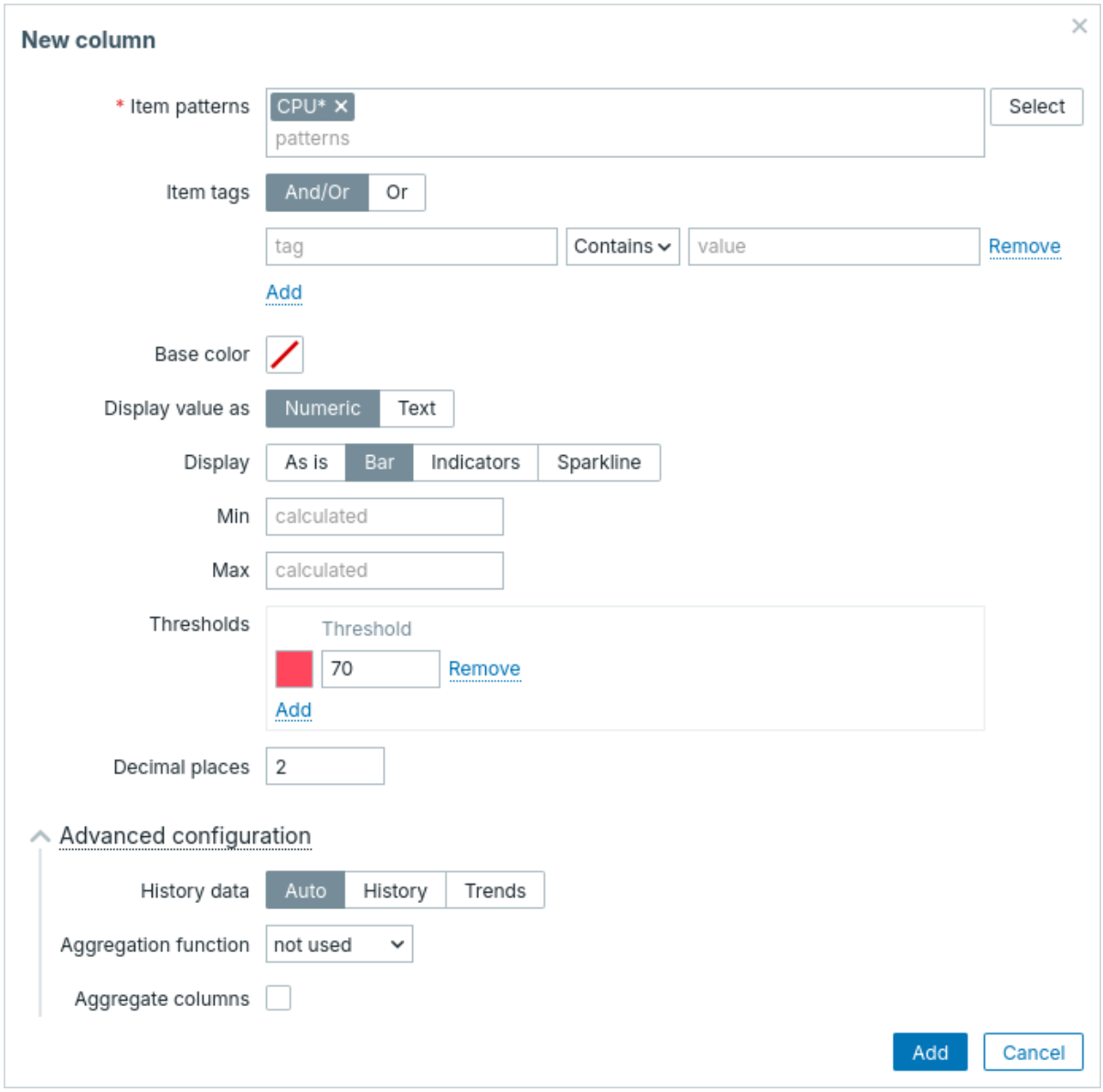
Группы узлов сети	Выберите группы узлов сети. Либо выберите совместимый виджет в качестве источника данных для групп узлов сети. Это поле поддерживает автодополнение, поэтому при вводе имени группы будет предложен выпадающий список подходящих групп. Этот параметр недоступен при настройке виджета на панели шаблона.
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 предложен выпадающий список подходящих узлов сети. Этот параметр недоступен при настройке виджета на панели шаблона.

Теги узлов сети	Укажите теги, чтобы ограничить количество данных узлов сети, отображаемых 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 несколько операторов: **Существует** - включить указанные имена тегов; **Равно** - включить указанные имена тегов и значения (с учетом регистра); **Содержит** - включить указанные имена тегов, где значения тегов 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, где значения тегов содержат введенную строку (поиск подстроки, без учета 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.
Макет	Выберите вариант макета: **Горизонтальный** - имена узлов сети будут отображаться слева; **Вертикальный** - имена узлов сети будут отображаться сверху.
Показывать проблемы	Отфильтруйте, какие проблемы отображать, на основе их состояния (все, неподтвержденные подавлением, нет). Проблемы обозначаются цветными значениями элементов данных, переопределяя настройки отображения столбца элементов данных. Цвета основаны на важности проблемы, которую можно настроить в *Администрирование* > *Общие* > *Параметры отображения триггеров*. Важности проблем можно изменить при обновлении проблем.
Элементы данных	Добавьте шаблоны элементов данных или конкретные элементы данных для отображения (см. Настройка столбцов).
Расширенная настройка	
Сортировка узлов сети	Выберите параметры сортировки для столбца/строки узлов сети.
Сортировать по	Сортировать узлы сети по: **Имени узла сети** - узлы сети будут отсортированы по имени узла сети; **Значению элемента данных** - узлы сети будут отсортированы по значению выбранных элементов данных.
Порядок	Выберите, отображать ли наибольшие или наименьшие значения: **TopN** - верхние N значений; **BottomN** - нижние N значений. Значение N выбирается в поле *Лимит*.
Лимит	Введите ограничение на количество отображаемых узлов сети (диапазон 1-1000; по умолчанию 10). Это значение станет значением N в поле *Порядок*.
Сортировка элементов данных	Выберите параметры сортировки для столбца/строки элементов данных.
Сортировать по	Сортировать элементы данных по: **Значению элемента данных** - элементы данных будут отсортированы по значению элемента данных; **Имени элемента данных** - элементы данных будут отсортированы по имени элемента данных; **Узлу сети** - элементы данных будут отсортированы по выбранному шаблону узла сети.
Порядок	Выберите, отображать ли наибольшие или наименьшие значения: **TopN** - верхние N значений; **BottomN** - нижние N значений. Значение N выбирается в поле *Лимит*.

Лимит	Введите ограничение на количество отображаемых элементов данных (диапазон 1-1000; по умолчанию 10). Это значение станет значением N в поле *Порядок*.
Показывать заголовок столбца	Выберите параметры отображения заголовка столбца: **Выкл.** - не отображать заголовок столбца; **Горизонтально** - отображать текст в заголовке горизонтально; **Вертикально** - 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** — как обычный текст; **Полоса** — как сплошная полоса, заполненная цветом; **Индикаторы** — как сегментированная полоса, заполненная цветом; **Спарклайн** — мини-график в виде линии.
Мин	Минимальное значение для отображения в виде полосы/индикаторов.
Макс	Максимальное значение для отображения в виде полосы/индикаторов.
Ширина	Установите толщину линии графика с помощью ползунка или вручную введя значение в диапазоне от 0 до 10. Этот параметр используется только для отображения в виде спарклайна.
Заливка	Установите уровень прозрачности цвета заливки с помощью ползунка или вручную введя значение в диапазоне от 0 до 10. Этот параметр используется только для отображения в виде спарклайна.
Цвет	Выберите цвет линии и заливки. Этот параметр используется только для отображения в виде спарклайна.
Описание полей, связанных с выбором периода времени и данных истории, см. в разделе *Расширенная конфигурация*.	
Пороги	Укажите пороговые значения, при достижении которых должен изменяться цвет фона/заливки. При сохранении список будет отсортирован по возрастанию. Для спарклайнов пороги применяются только к последнему значению элемента данных.
Десятичные знаки ***Расширенная конфигурация***	Укажите, сколько знаков после запятой должно отображаться у значения.

Данные истории	Брать данные из истории или трендов: **Авто** — автоматический выбор; **История** — брать данные истории; **Тренды** — брать данные трендов. Этот параметр доступен только для числовых значений элементов данных. Для текстовых значений данные всегда будут браться из истории.
Функция агрегации	Укажите, какую функцию агрегации использовать: **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; **not used** — отображать самое последнее значение (без агрегации). Агрегация позволяет отображать агрегированное значение за выбранный интервал (5 минут, час, день) вместо самого последнего значения. Для *min*, *max*, *avg* и *sum* могут отображаться только числовые данные. Для *count* нечисловые данные будут преобразованы в числовые.
Период времени	Укажите период времени, который следует использовать для агрегации значений. 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 он задан, в правом верхнем углу виджета будет отображаться значок часов, показывающий установленное время при наведении курсора. Обратите внимание, что независимо от конфигурации *Период времени* виджета, совместимые виджеты все равно могут использовать его как источник данных для периода времени. Этот параметр недоступен, если для *Функция агрегации* установлено значение "not used".
Виджет	Введите или выберите совместимый виджет в качестве источника данных для периода времени. Этот параметр доступен, если для *Период времени* установлено значение "Widget".
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 "Custom".
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 "Custom".
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 агрегированы с использованием выбранной *Функция агрегации столбца* и отображены под именем *Имя объединенного столбца*. По умолчанию: выключено (флажок снят). Этот параметр отображается только если для *Отображать значение как* выбрано **Числовое**.
Функция агрегации столбца	Выберите, как объединять значения 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 для агрегированного столбца. Это поле является обязательным, если установлен флажок *Агрегировать столбцы*, и становится заголовком столбца для объединенных значений.

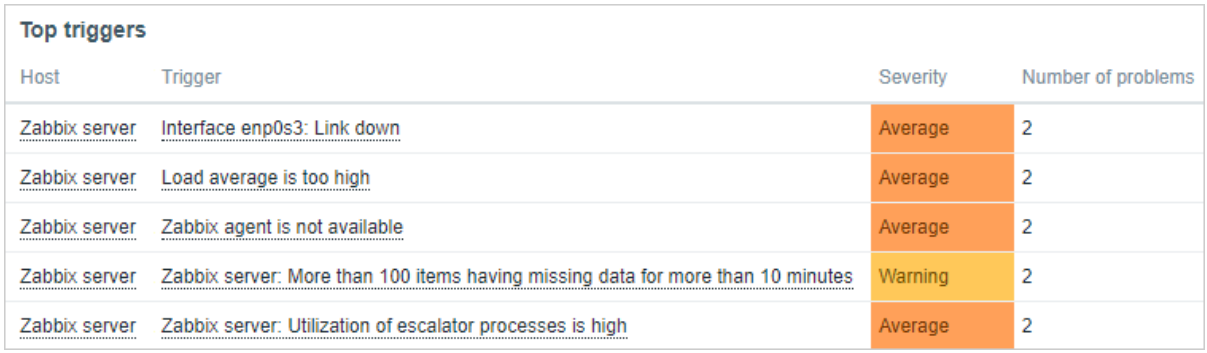
Параметры столбца для текстовых значений:

Подсветка	Укажите регулярные выражения, при совпадении с которыми должен изменяться цвет фона/заливки.

30 лучших триггеров

Обзор

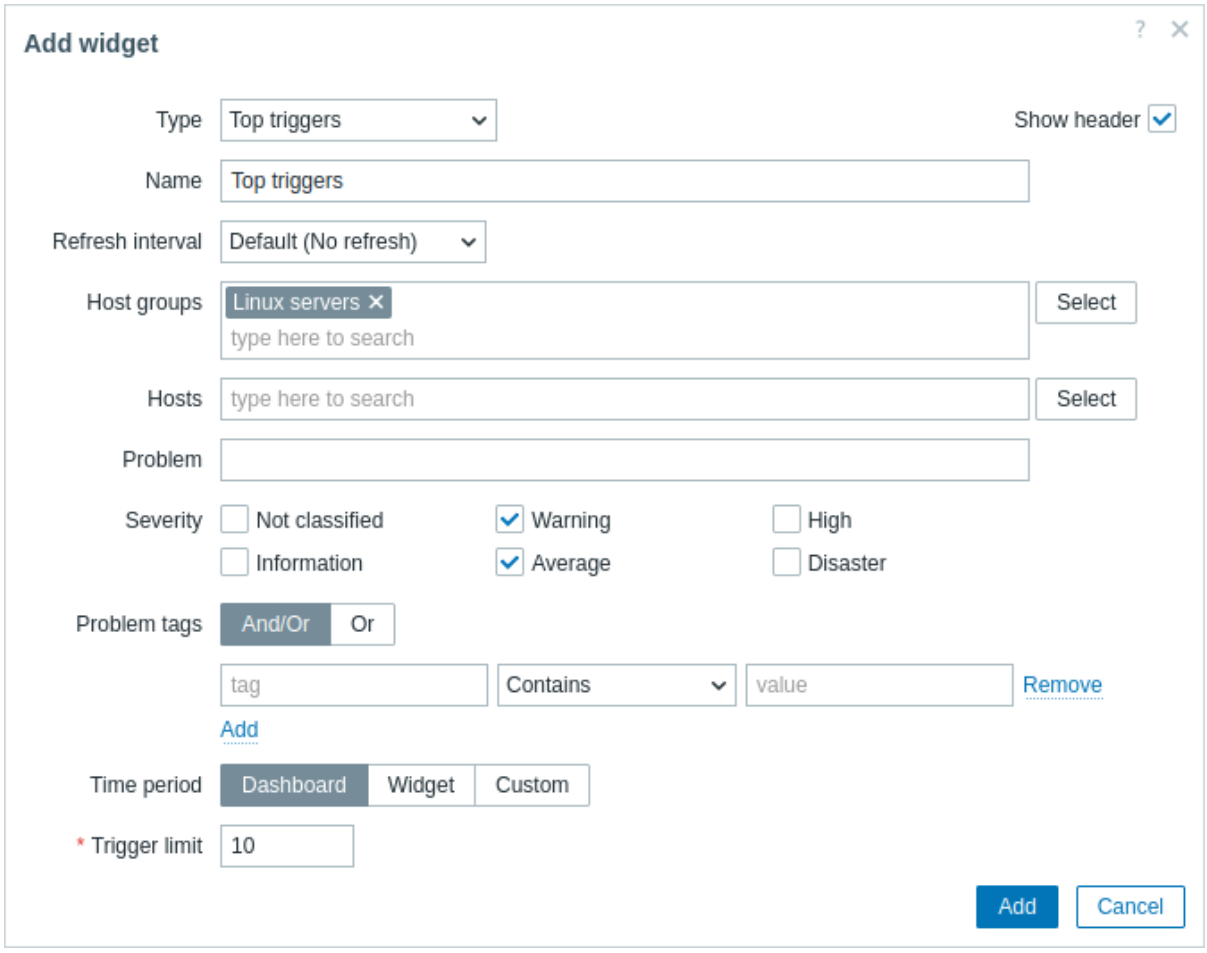
Виджет *Топ триггеров* отображает триггеры с наибольшим количеством проблем.



Он отображает те же данные, что и *Отчёты > Топ 100 триггеров*, и может отображать до 1000 записей.

Настройка

Для настройки выберите *Top triggers* в качестве типа виджета:



В дополнение к параметрам, общим для всех виджетов, вы можете задать следующие специальные параметры:

Группы узлов сети	Выберите группы узлов сети, проблемы из которых будут отображаться в виджете. Это поле поддерживает автодополнение, поэтому при вводе имени группы будет предложен 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 дашборде шаблона.

Узлы сети	Выберите узлы сети, проблемы которых будут отображаться в виджете. Это поле поддерживает автодополнение, поэтому при вводе имени узла сети будет предложен выпадающий список подходящих узлов сети. Если узлы сети не указаны, будут отображаться проблемы всех узлов сети. Этот параметр недоступен при настройке виджета на дашборде шаблона.
Проблема	Вы можете просматривать триггеры только для определённых проблем. Для этого введите строку, которая должна совпадать с именем проблемы. Макросы не раскрываются.
Важность	Отметьте важности триггеров, чтобы отфильтровать триггеры, отображаемые в виджете. Если не отмечена ни одна важность, будут отображаться все триггеры.
Теги проблем	Укажите теги проблем, которые должны отображаться 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, условия с одинаковым именем тега будут сгруппированы по условию *Или*; **Или** - достаточно выполнения одного условия.
Период времени	Выберите источник данных для периода времени: **Дашборд** - использовать селектор периода времени дашборда; **Виджет** - использовать совместимый виджет (задаётся 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, показывающий установленное время при наведении курсора. Обратите внимание, что независимо от настройки *Период времени* у виджета, совместимые виджеты всё 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”.
Лимит триггеров	Задайте количество отображаемых триггеров. Допустимый диапазон значений: 1-1000.

31 Обзор триггеров

Обзор

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.



Виджет использует несколько визуальных элементов для отображения сведений о триггерах:

- Цвета блоков для триггеров в состоянии PROBLEM соответствуют важности проблем.
- Мигание блоков указывает 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 отображался, должны быть подтверждены все активные или устраненные проблемы для триггера.

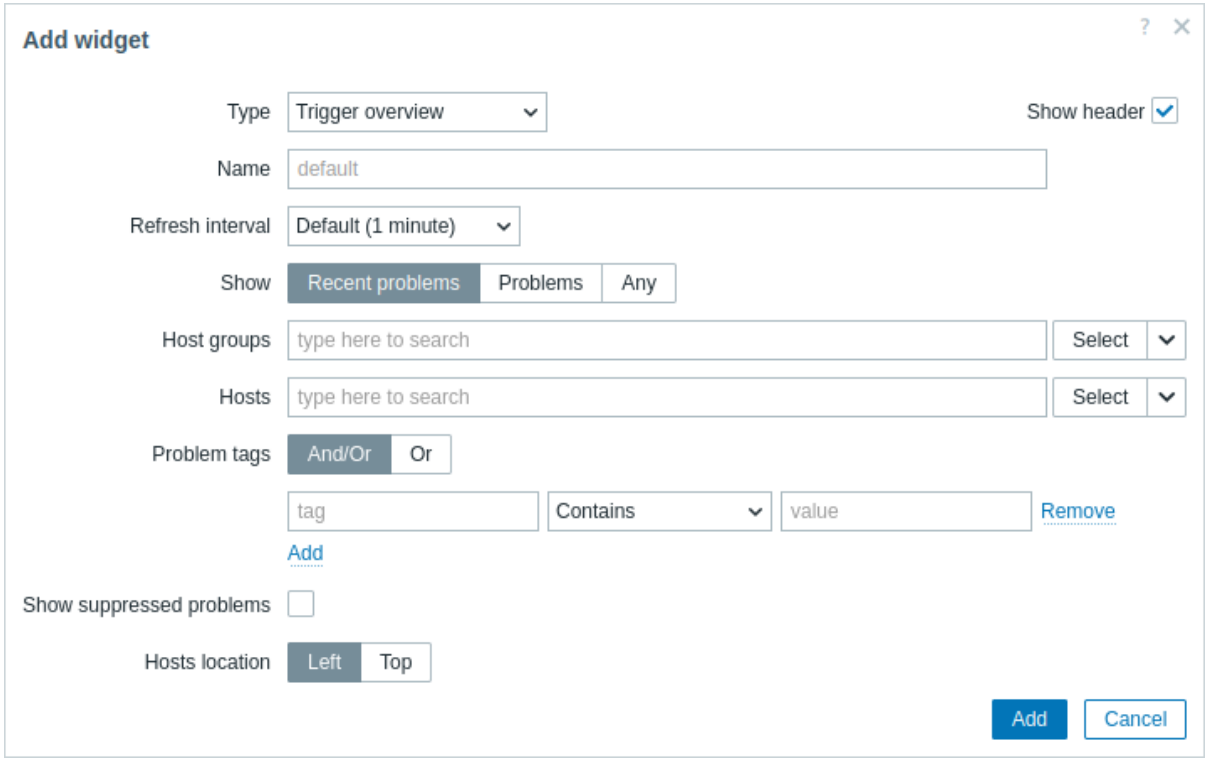
Note:
Цвета важности триггеров можно настроить в разделе *Administration* > *General*. Важность триггеров можно изменить при обновлении проблем.

Щелчок по блоку предоставляет ссылки на информацию, связанную с триггером (проблемы, конфигурация триггера и т. д.).

По умолчанию отображается до 50 записей (настраивается в *Administration* > *General* > *GUI* в параметре *Max number of columns and rows in overview tables*). Если триггеров больше, будет показано сообщение с предложением уточнить критерии фильтрации. Это ограничение применяется до любой дополнительной фильтрации (например, по тегам).

Конфигурация

Для настройки выберите *Обзор триггеров* в качестве типа:



В дополнение к параметрам, которые являются общими для всех виджетов, можно задать следующие специальные параметры:

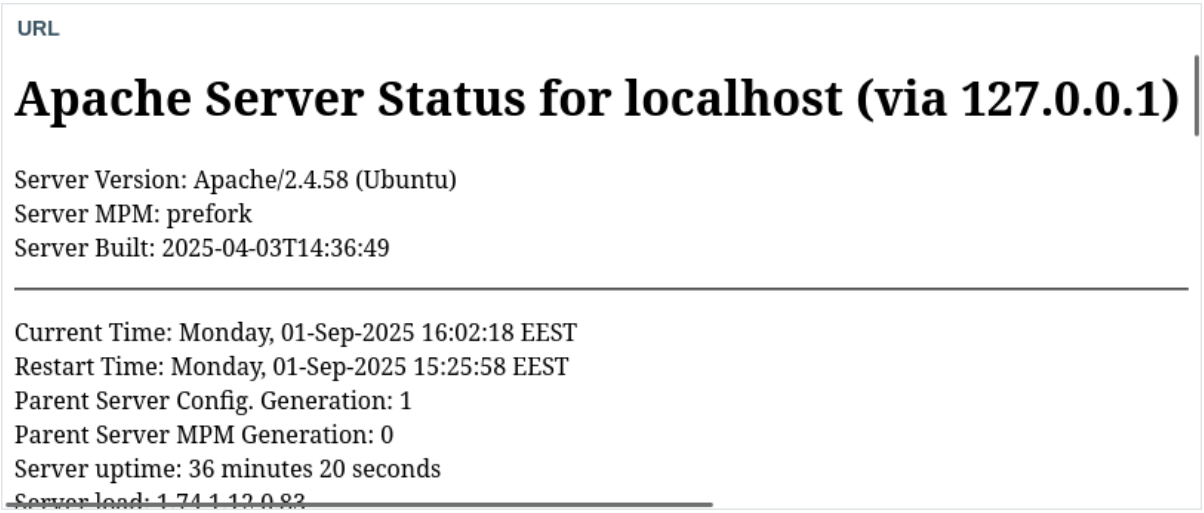
Показывать	Фильтрация триггеров по состоянию триггера: **Недавние проблемы** — *(по умолчанию)* показывать триггеры, которые недавно были или всё ещё находятся в состоянии ПРОБЛЕМА (решённые и нерешённые); **Проблемы** — показывать триггеры, находящиеся в состоянии ПРОБЛЕМА (нерешённые); **Любые** — показывать все триггеры.
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 предложен выпадающий список подходящих групп. Этот параметр недоступен при настройке виджета на панели шаблона.
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 предложен выпадающий список подходящих узлов сети. Этот параметр недоступен при настройке виджета на панели шаблона.

Теги проблем	Укажите теги для фильтрации триггеров, 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. **Примечание:** Если параметр *Показывать* установлен в значение «Любые», будут отображаться все триггеры, даже если указаны теги. Однако, хотя недавние изменения состояния триггера (отображаемые как мигающие блоки) будут обновляться для всех триггеров, подробности состояния триггера (цвет важности проблемы и факт подтверждения проблемы) будут обновляться только для триггеров, соответствующих указанным тегам. Для каждого условия доступны несколько операторов: **Существует** — включить указанные имена тегов; **Равно** — включить указанные имена и значения тегов (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 и значения тегов (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.
Показывать подавленные проблемы	Установите флажок, чтобы отображать проблемы, которые в противном случае были бы подавлены (не показаны) из-за обслуживания узла сети.
Макет	Выберите вариант макета: **Горизонтальный** — имена узлов сети будут отображаться слева; **Вертикальный** — имена узлов сети будут отображаться сверху.

32 URL

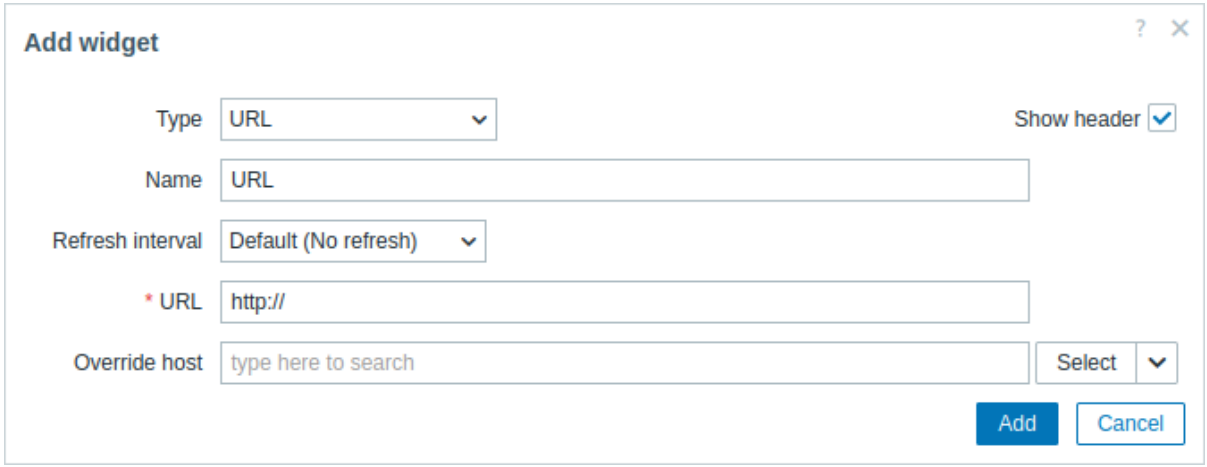
Обзор

Виджет *URL* отображает содержимое, полученное по URL.



Конфигурация

Для настройки выберите *URL* в качестве типа:



В дополнение к параметрам, которые являются общими для всех виджетов, вы можете задать следующие специальные параметры:

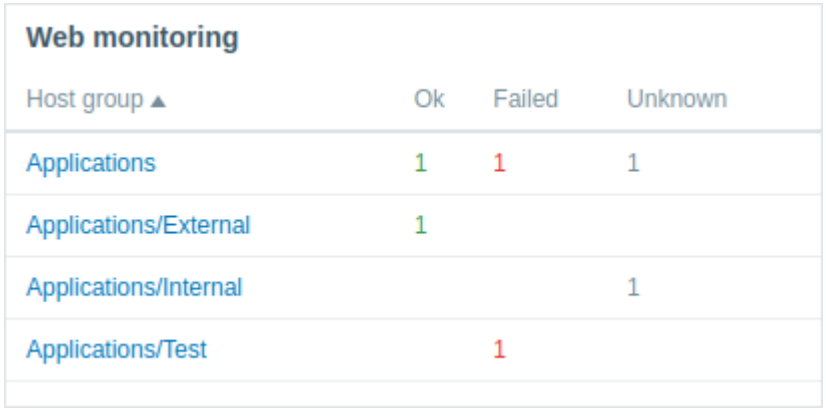
URL	Введите URL для отображения (до 2048 символов). Внешние URL должны начинаться с `http://` или `https://`. Внутренние URL поддерживают относительные пути (например, `zabbix.php?action=report.status`). Макросы {HOST.*} поддерживаются.
Override host	Выберите совместимый виджет или селектор узла сети панели как источник данных для узлов сети. Этот параметр недоступен при настройке виджета на панели шаблона.

Attention:
Браузеры могут не загружать HTTP-страницу, настроенную в виджете, если доступ к веб-интерфейсу Zabbix осуществляется по HTTPS. Кроме того, вам может потребоваться изменить HTTP-заголовок ответа X-Frame-Options, чтобы отображать содержимое, внешнее по отношению к веб-интерфейсу Zabbix.

33 Веб-мониторинг

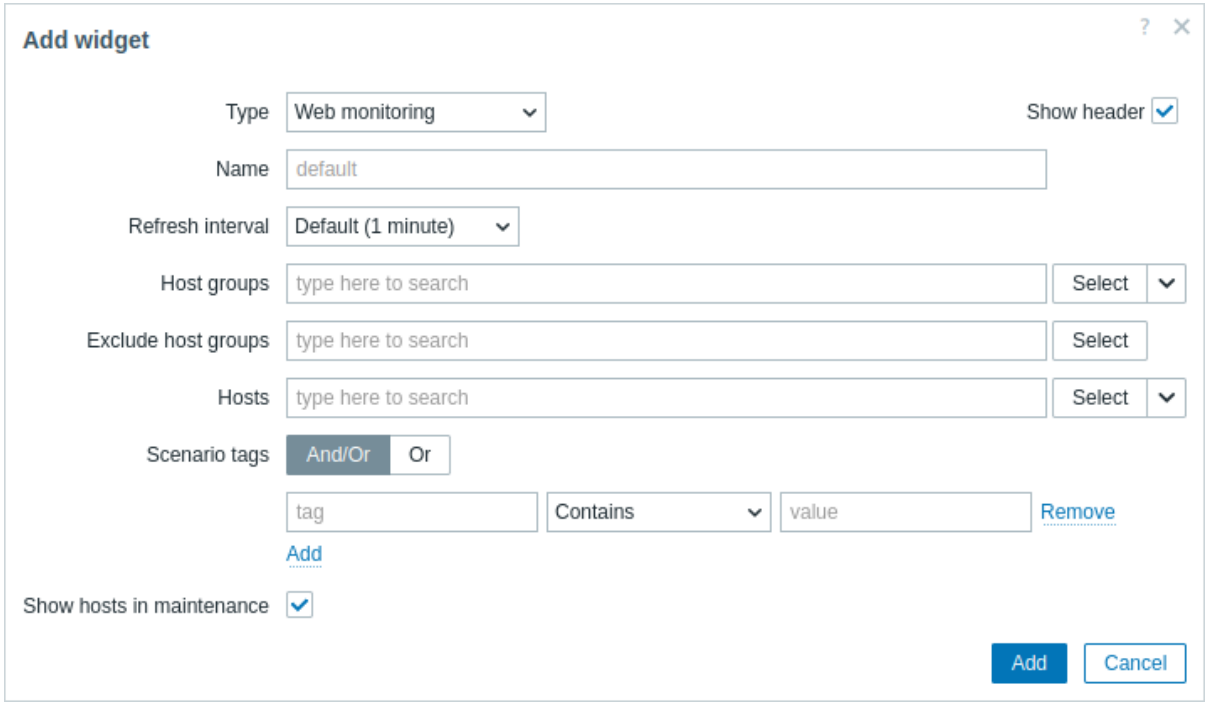
Обзор

Виджет *Веб-мониторинг* отображает состояние активных веб-сценариев.



Щелчок по имени группы узлов сети открывает *Мониторинг* > *Узлы сети* > *Веб-сценарии*, где можно просмотреть подробную информацию о сценарии.

Настройка



Помимо параметров, которые являются общими для всех виджетов, вы можете задать следующие специальные параметры:

Группы узлов сети	Выберите группы узлов сети для отображения в виджете. В качестве источника данных для групп узлов сети также можно выбрать совместимый виджет. Это поле поддерживает автодополнение, поэтому при вводе имени группы будет предложен выпадающий список подходящих групп. Указание родительской группы узлов сети неявно выбирает все вложенные группы узлов сети. В виджете будут отображаться данные узлов сети из этих групп узлов сети; если группы узлов сети не указаны, будут отображаться все группы узлов сети. Этот параметр недоступен при настройке виджета на панели шаблона.
Исключить группы узлов сети	Выберите группы узлов сети, которые нужно скрыть в виджете. Это поле поддерживает автодополнение, поэтому при вводе имени группы будет предложен выпадающий список подходящих групп. Указание родительской группы узлов сети неявно выбирает все вложенные группы узлов сети. Данные узлов сети из этих групп узлов сети не будут отображаться в виджете. Например, узлы сети 001, 002, 003 могут входить в Группу A, а узлы сети 002, 003 также в Группу B. Если одновременно выбрать *показывать* Группу A и *исключить* Группу B, на панели будут отображаться только данные узла сети 001. Этот параметр недоступен при настройке виджета на панели шаблона.
Узлы сети	Выберите узлы сети для отображения в виджете. В качестве источника данных для узлов сети также можно выбрать совместимый виджет или панель. Это поле поддерживает автодополнение, поэтому при вводе имени узла сети будет предложен выпадающий список подходящих узлов сети. Если узлы сети не указаны, будут отображаться все узлы сети. Этот параметр недоступен при настройке виджета на панели шаблона.

Теги сценариев	Укажите теги, чтобы ограничить количество веб-сценариев, 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 следующие операторы: **Существует** — включить указанные имена тегов; **Равно** — включить указанные имена тегов и значения (с учётом регистра); **Содержит** — включить указанные имена тегов, у которых 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, у которых 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.
Показывать узлы сети в обслуживании	Включать в статистику узлы сети, находящиеся на обслуживании. При настройке виджета на панели шаблона этот параметр называется *Показывать данные в обслуживании*.

2 Мониторинг

Обзор

Меню «Мониторинг» предназначено для отображения данных. Какая бы информация ни была настроена в Zabbix для сбора, визуализации и выполнения действий на её основе, она будет отображаться в различных разделах меню «Мониторинг».

Кнопки режима просмотра

Следующие кнопки, расположенные в правом верхнем углу, являются общими для каждого раздела:

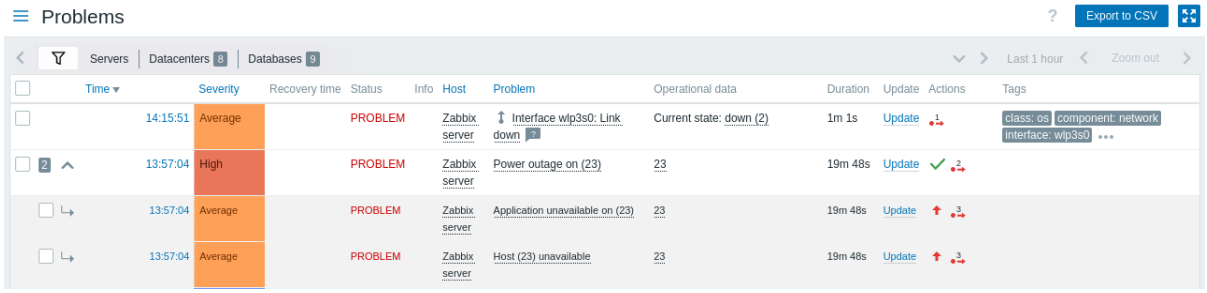
	Отображение страницы в режиме киоска. В этом режиме отображается только содержимое страницы. Чтобы выйти из режима киоска, переместите курсор мыши, пока не появится кнопка выхода  , и нажмите на неё. Вы вернётесь в обычный режим.

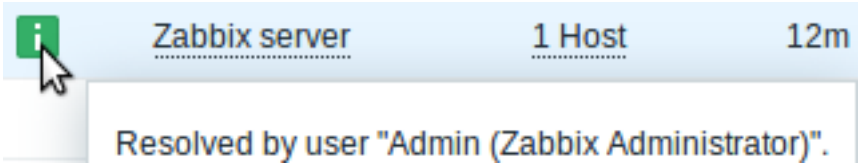
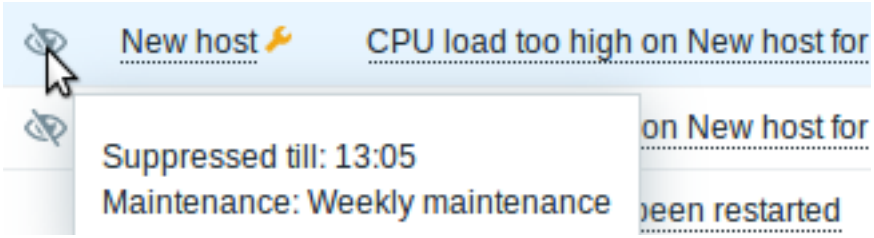
1 Проблемы

Обзор

В *Мониторинг* > *Проблемы* вы можете увидеть, какие проблемы имеются в данный момент. Проблемами считаются те триггеры, которые находятся в состоянии ”Проблема”.

По умолчанию все новые проблемы классифицируются как причинные проблемы. Некоторые проблемы можно вручную переклассифицировать как симптоматические проблемы причинной проблемы. Подробнее см. в разделе причинные и симптоматические события.



Столбец	Описание
Флажок	Отображаются флажки для выбора проблем. Значки рядом с флажками имеют следующее значение:  - количество симптоматических событий для причинной проблемы;  - развернуть, чтобы показать симптоматические события;  - свернуть, чтобы скрыть симптоматические события;  - это симптоматическое событие.
Время	Отображается время начала проблемы.
Важность	Отображается важность проблемы. Изначально важность проблемы основана на важности исходного триггера проблемы, однако после возникновения события она может быть обновлена с помощью экрана *Обновить проблему*. Цвет важности проблемы используется как фон ячейки в течение времени проблемы.
Время восстановления	Отображается время устранения проблемы.
Статус	Отображается статус проблемы: **Проблема** - неустраненная проблема **Устранено** - недавно устраненная проблема. Недавно устраненные проблемы можно скрыть с помощью фильтра. Новые и недавно устраненные проблемы мигают в течение 2 минут. Устраненные проблемы отображаются в общей сложности 5 минут. Оба этих значения настраиваются в *Администрирование* > *Общие* > *Параметры отображения триггеров*.
Информация	Зеленый значок информации отображается, если проблема закрыта глобальной корреляцией или вручную при обновлении проблемы. При наведении указателя мыши на значок отображаются дополнительные сведения:  Следующий значок отображается, если показана подавленная проблема (см. параметр *Показывать подавленные проблемы* в фильтре). При наведении указателя мыши на значок отображаются дополнительные сведения: 
Узел сети	Отображается узел сети проблемы. Щелчок по имени узла сети открывает меню узла сети.
Проблема	Отображается имя проблемы. Имя проблемы основано на имени исходного триггера проблемы. Макросы в имени триггера разрешаются в момент возникновения проблемы, и полученные значения больше не обновляются. *Обратите внимание*, что к имени проблемы можно добавить оперативные данные, показывающие некоторые последние значения элементов данных. Щелчок по имени проблемы открывает меню события. Наведение указателя на значок  после имени проблемы открывает описание триггера (для тех проблем, у которых оно есть).
Оперативные данные	Отображаются оперативные данные, содержащие последние значения элементов данных. Оперативные данные могут представлять собой комбинацию текста и макросов значений элементов данных, если это настроено на уровне триггера. Если оперативные данные не настроены на уровне триггера, отображаются последние значения всех элементов данных из выражения. Этот столбец отображается только в том случае, если для параметра *Показывать оперативные данные* в фильтре выбрано значение *Отдельно*.
Длительность	Отображается длительность проблемы. См. также: Отрицательная длительность проблемы

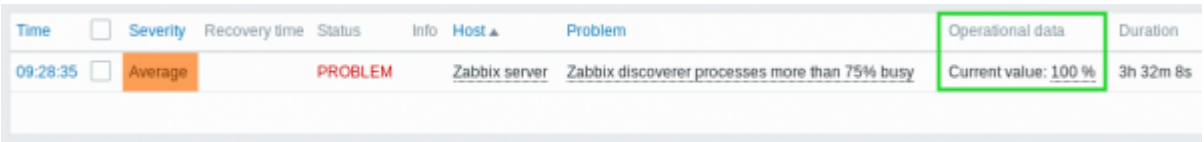
Столбец	Описание
Обновить	Щелкните по ссылке *Обновить*, чтобы перейти к экрану обновления проблемы, где с проблемой можно выполнить различные действия, включая комментирование и подтверждение проблемы.
Действия	История активности по проблеме отображается с помощью символических значков:  - проблема была подтверждена. Этот значок всегда отображается первым.  - были добавлены комментарии. Также отображается количество комментариев.  - важность проблемы была повышена (например, Информация > Предупреждение).  - важность проблемы была понижена (например, Предупреждение > Информация).  - важность проблемы была изменена, но возвращена к исходному уровню (например, Предупреждение > Информация > Предупреждение).  - проблема была подавлена.  - подавление проблемы было снято.  - действия были выполнены. Также отображается количество действий.  - действия были выполнены, по крайней мере одно находится в процессе. Также отображается количество действий.  - действия были выполнены, по крайней мере одно завершилось ошибкой. Также отображается количество действий. При наведении указателя мыши на значки отображаются всплывающие окна с подробностями об активности. Подробнее о значках, используемых во всплывающем окне для выполненных действий, см. в разделе просмотр подробностей.
Теги	Отображаются теги (если есть). Кроме того, могут отображаться теги из внешней системы обработки заявок (см. параметр *Обрабатывать теги* при настройке вебхуков).

Операционные данные проблем

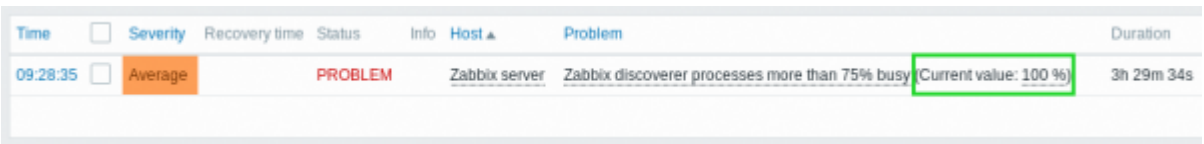
Для текущих проблем можно отображать операционные данные, то есть последние значения элементов данных, в отличие от значений элементов данных на момент возникновения проблемы.

Отображение операционных данных можно настроить в фильтре *Мониторинг > Проблемы* или в настройках соответствующего виджета панели, выбрав один из трех вариантов:

- *Нет* — операционные данные не отображаются
- *Отдельно* — операционные данные отображаются в отдельном столбце



- *С именем проблемы* — операционные данные добавляются к имени проблемы в скобках. Операционные данные добавляются к имени проблемы только в том случае, если поле *Операционные данные* не пустое в настройках триггера.



Содержимое операционных данных можно настроить для каждого триггера в поле *Операционные данные*. Это поле принимает произвольную строку с макросами, в первую очередь макрос `{ITEM.LASTVALUE<1-9>}`.

`{ITEM.LASTVALUE<1-9>}` в этом поле всегда будет разрешаться в последние значения элементов данных в выражении триггера. `{ITEM.VALUE<1-9>}` в этом поле будет разрешаться в значения элементов данных на момент изменения

состояния триггера (то есть при переходе в ** **, при переходе в ** **, при ручном закрытии пользователем или при закрытии корреляцией).

Обратите внимание, что ручное закрытие проблемы не создает нового значения, поэтому разрешенное значение {ITEM.LASTVALUE<1-9>} или {ITEM.VALUE<1-9>} по-прежнему будет показывать значение на момент проблемы.

{ITEM.LASTVALUE<1-9>} или {ITEM.VALUE<1-9>} будет разрешаться в *UNKNOWN*, если последнее значение истории было собрано более *Max history display period* назад (см. *Администрирование > Общее*).

Отрицательная длительность проблемы

В некоторых распространённых ситуациях действительно возможна отрицательная длительность проблемы, то есть когда время устранения проблемы раньше времени её создания, например:

- Если какой-либо узел сети мониторится прокси и происходит сетевая ошибка, из-за которой некоторое время не поступают данные от прокси, триггер nodata(/host/key) будет сработан сервером. Когда соединение восстанавливается, сервер получает от прокси данные элемента данных с меткой времени из прошлого. Затем проблема nodata(/host/key) будет устранена, и у неё будет отрицательная длительность проблемы;
- Если данные элемента данных, устраняющие событие проблемы, отправляются с помощью Zabbix sender и содержат метку времени более раннюю, чем время создания проблемы, также будет отображаться отрицательная длительность проблемы.

Note:
Отрицательная длительность проблемы никак не влияет на расчёт SLA или отчёт о доступности конкретного триггера; она не уменьшает и не увеличивает время проблемы.

Параметры массового редактирования

Кнопки под списком предоставляют несколько вариантов массового редактирования:

- *Массовое обновление* — обновить выбранные проблемы, перейдя на экран обновления проблем

Чтобы использовать этот параметр, установите флажки напротив соответствующих проблем, затем нажмите кнопку *Массовое обновление*.

Кнопки

Кнопка справа предоставляет следующую возможность:

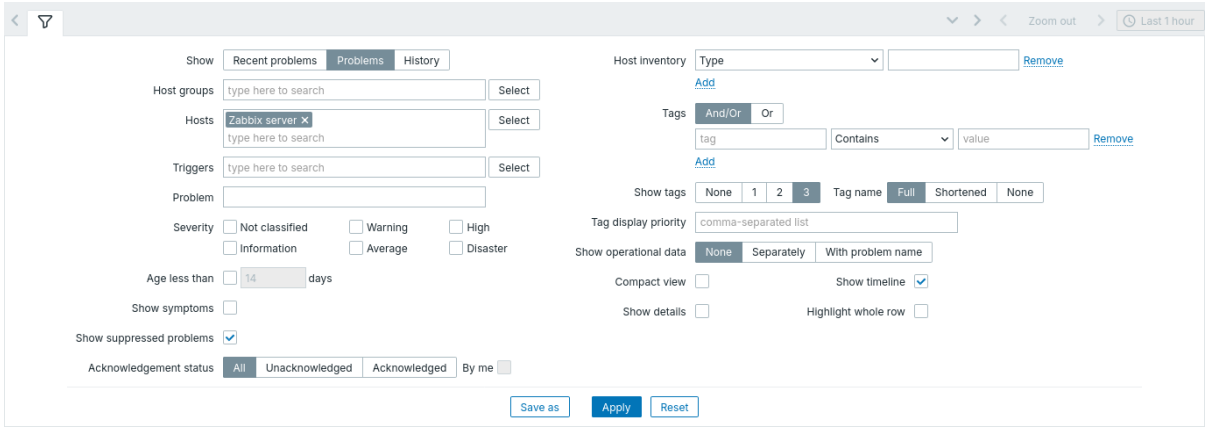
Export to CSV	Экспортировать содержимое со всех страниц в CSV-файл.

Кнопки режима просмотра, общие для всех разделов, описаны на странице Monitoring.

Использование фильтра

Вы можете использовать фильтр, чтобы отображать только те проблемы, которые вас интересуют. Для повышения производительности поиска данные ищутся с нераскрытыми макросами.

Фильтр расположен над таблицей. Избранные настройки фильтра можно сохранить в виде вкладок, а затем быстро открывать, нажимая на вкладки над фильтром.



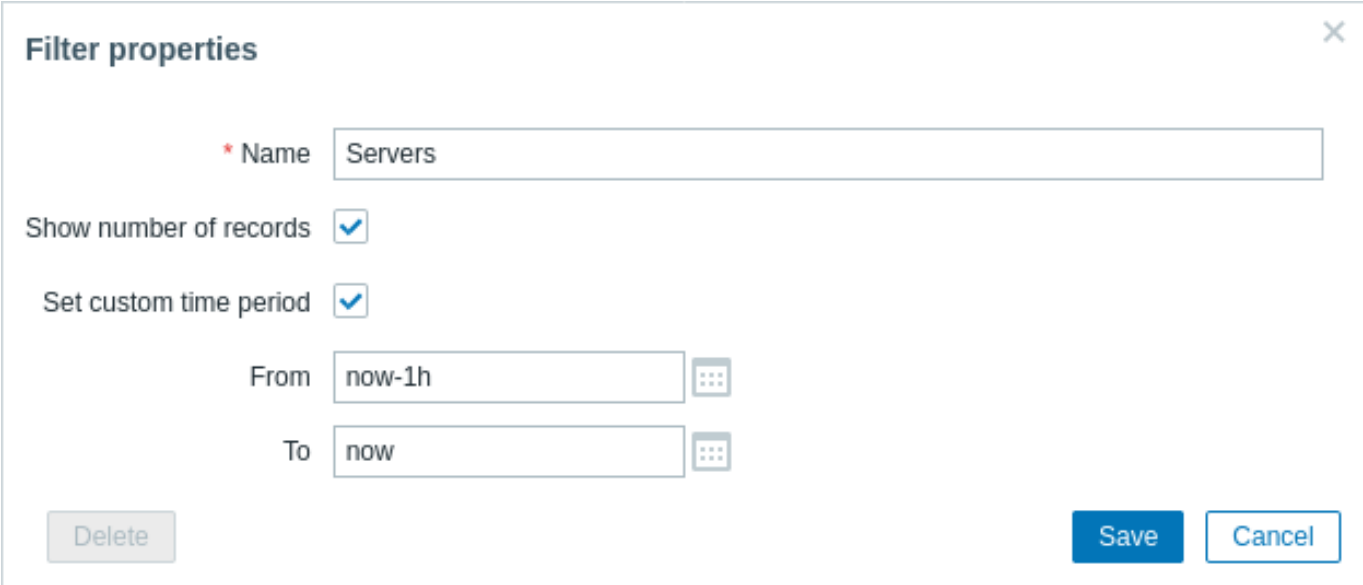
Параметр	Описание
Показывать	Фильтрация по состоянию проблемы: **Недавние проблемы** — отображаются неустраненные и недавно устраненные проблемы (по умолчанию) **Проблемы** — отображаются неустраненные проблемы **История** — отображается история всех событий
Группы узлов сети	Фильтрация по одной или нескольким группам узлов сети. Указание родительской группы узлов сети неявно выбирает все вложенные группы узлов сети.
Узлы сети	Фильтрация по одному или нескольким узлам сети.
Триггеры	Фильтрация по одному или нескольким триггерам.
Проблема	Фильтрация по имени проблемы.
Важность	Фильтрация по важности триггера (проблемы).
Возраст меньше чем	Фильтрация по давности проблемы.
Показывать симптомы	Установите флажок, чтобы отображать проблемы, классифицированные как симптомы, в отдельной строке.
Показывать подавленные проблемы	Установите флажок, чтобы отображать проблемы, которые в противном случае были бы подавлены (не показаны) из-за обслуживания узла сети или отдельного подавления проблем.
Статус подтверждения	Фильтрация для отображения всех проблем, только неподтвержденных проблем или только подтвержденных проблем. Установите дополнительный флажок, чтобы исключить проблемы, которые когда-либо были подтверждены вами.
Инвентарные данные узла сети	Фильтрация по типу и значению инвентарных данных.
Теги	Фильтрация по имени и значению тега события. 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 несколько операторов: **Существует** — включить указанные имена тегов **Равно** — включить указанные имена и значения тегов (с учетом регистра) **Содержит** — включить указанные имена тегов, где значения тегов содержат введенную строку (поиск подстроки, без учета регистра) **Не существует** — исключить указанные имена тегов **Не равно** — исключить указанные имена и значения тегов (с учетом регистра) **Не содержит** — исключить указанные имена тегов, где значения тегов содержат введенную строку (поиск подстроки, без учета 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 При фильтрации указанные здесь теги будут отображаться у проблемы первыми, если это не переопределено списком *Приоритет отображения тегов* (см. ниже).
Показывать теги	Выберите количество отображаемых тегов: **Нет** — столбец *Теги* отсутствует в *Мониторинг* > *Проблемы* **1** — столбец *Теги* содержит один тег **2** — столбец *Теги* содержит два тега **3** — столбец *Теги* содержит три тега Чтобы увидеть все теги проблемы, наведите указатель мыши на значок с тремя точками.
Имя тега	Выберите режим отображения имени тега: **Полное** — имена и значения тегов отображаются полностью **Сокращенное** — имена тегов сокращаются до 3 символов; значения тегов отображаются полностью **Нет** — отображаются только значения тегов; имена не показываются
Приоритет отображения тегов	Введите приоритет отображения тегов для проблемы в виде списка тегов, разделенных запятыми (например: `customer,scope,component`). Следует использовать только имена тегов, без значений. Теги из этого списка всегда будут отображаться первыми, переопределяя естественный алфавитный порядок.
Показывать оперативные данные	Выберите режим отображения оперативных данных: **Нет** — оперативные данные не отображаются **Отдельно** — оперативные данные отображаются в отдельном столбце **С именем проблемы** — оперативные данные добавляются к имени проблемы в скобках
Компактный вид	Установите флажок, чтобы включить компактный вид.

Параметр	Описание
Показывать детали	Установите флажок, чтобы отображать базовые выражения триггеров проблем. Недоступно, если установлен флажок *Компактный вид*.
Показывать временную шкалу	Установите флажок, чтобы отображать визуальную временную шкалу и группировку. Недоступно, если установлен флажок *Компактный вид*.
Подсвечивать всю строку	Установите флажок, чтобы подсвечивать всю строку для неустраненных проблем. Для подсветки используется цвет важности проблемы. *Подсвечивать всю строку* недоступно в темах с высокой контрастностью.

Вкладки для избранных фильтров

Часто используемые наборы параметров фильтра можно сохранять во вкладках.

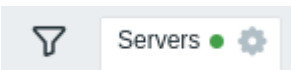
Чтобы сохранить новый набор параметров фильтра, откройте основную вкладку и настройте параметры фильтра, затем нажмите кнопку *Save as*. В новом всплывающем окне задайте *Свойства фильтра*.



Параметр	Описание
Name	Имя фильтра, отображаемое в списке вкладок.
Show number of records	Установите флажок, если хотите, чтобы рядом с именем вкладки отображалось количество проблем.
Override time period selector	Установите флажок, чтобы задать определённый период времени по умолчанию для этого набора фильтров. Если параметр задан, изменить период времени для этой вкладки можно будет только через обновление настроек фильтра. Для вкладок без пользовательского периода времени временной диапазон можно изменить, нажав кнопку выбора периода 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 формате времени (`now-1d`). Доступно, если установлен флажок *Set custom time period*.

После сохранения фильтр создаётся как именованная вкладка фильтра и сразу становится активным.

Чтобы изменить свойства существующего фильтра, нажмите значок шестерёнки рядом с именем активной вкладки.



Примечания:

- Чтобы скрыть область фильтра, нажмите на имя текущей вкладки. Нажмите на имя активной вкладки ещё раз, чтобы снова открыть область фильтра.
- Поддерживается навигация с клавиатуры: используйте стрелки для переключения между вкладками, нажмите *Enter* для открытия.

- Кнопки влево/вправо над фильтром можно использовать для переключения между сохранёнными фильтрами. Либо кнопка со стрелкой вниз открывает выпадающее меню со всеми сохранёнными фильтрами, где можно выбрать нужный.
- Вкладки фильтра можно переупорядочивать перетаскиванием.
- Если настройки сохранённого фильтра были изменены (но не сохранены), после имени фильтра отображается зелёная точка. Чтобы обновить фильтр в соответствии с новыми настройками, нажмите кнопку *Update*, которая отображается вместо кнопки *Save as*.
- Текущие настройки фильтра запоминаются в профиле пользователя. Когда пользователь снова откроет страницу, настройки фильтра останутся прежними.

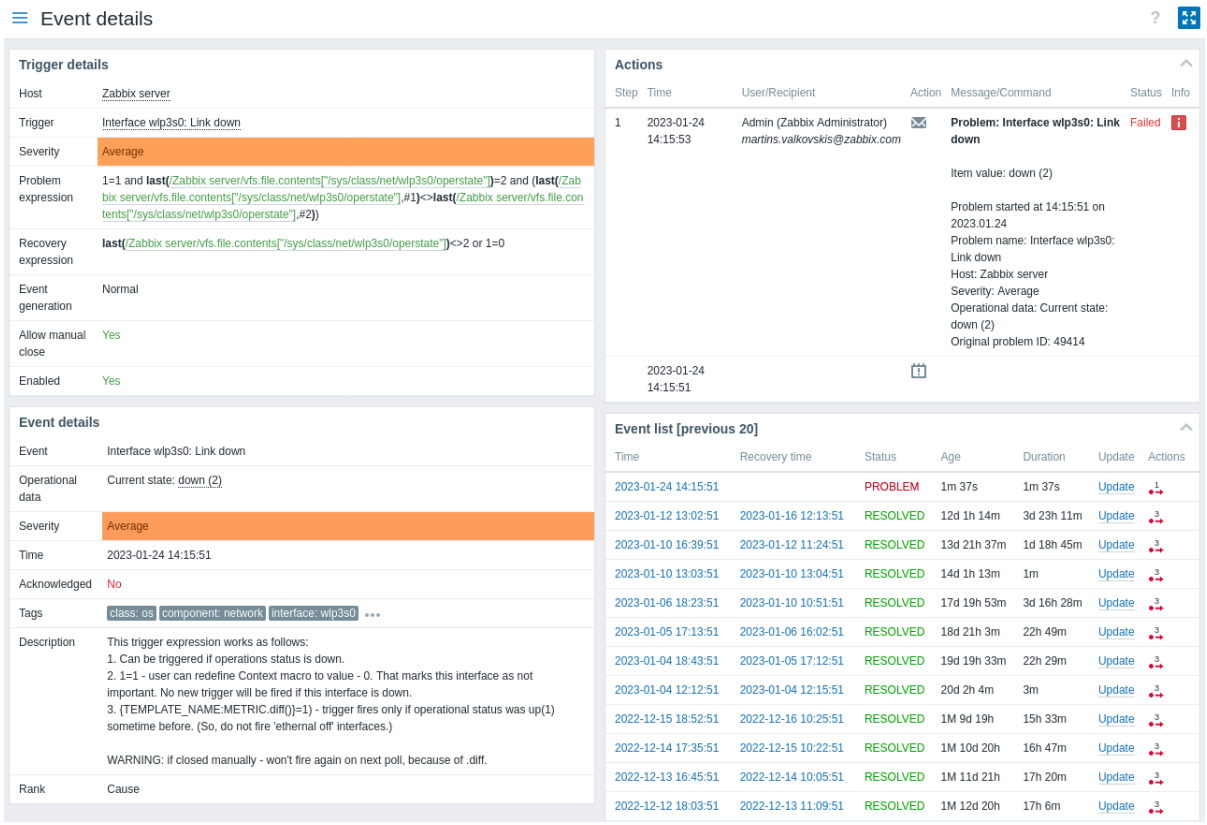
Note:
Чтобы поделиться фильтрами, скопируйте URL активного фильтра и отправьте его другим пользователям. После открытия этого URL другие пользователи смогут сохранить этот набор параметров как постоянный фильтр в своей учётной записи Zabbix.
См. также: Параметры страницы.

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 сведений

Время начала проблемы и восстановления в *Мониторинг* > *Проблемы* являются ссылками. При нажатии на них открываются более подробные сведения о событии.



Обратите внимание, что важность проблемы может отличаться для триггера и события проблемы, если она была обновлена для события проблемы с помощью экрана *Обновить проблему*.

В списке действий для обозначения типа активности используются следующие значки:

-  — событие проблемы создано.
-  — сообщение было отправлено.
-  — событие проблемы подтверждено.
-  — подтверждение события проблемы снято.
-  — комментарий был добавлен.
-  — важность проблемы была повышена (например, Information > Warning).
-  — важность проблемы была понижена (например, Warning > Information).
-  — важность проблемы была изменена, но возвращена к исходному уровню (например, Warning > Information > Warning).
-  — удалённая команда была выполнена.
-  — событие проблемы восстановлено.
-  — проблема была закрыта вручную.
-  — проблема была подавлена.
-  — подавление проблемы было снято.
-  — проблема была преобразована в проблему-симптом.
-  — проблема была преобразована в проблему-причину.

1 Проблемы причин и симптомов

Обзор

По умолчанию все новые проблемы классифицируются как причинные проблемы. Можно вручную переклассифицировать определённые проблемы в симптоматические проблемы причинной проблемы.

Например, отключение электропитания может быть фактической первопричиной того, почему какой-либо узел сети недоступен или какой-либо сервис не работает. В этом случае проблемы «узел сети недоступен» и «сервис не работает» должны быть классифицированы как симптоматические проблемы проблемы «отключение электропитания» — причинной проблемы.

Иерархия причин и симптомов поддерживает только два уровня. Проблеме, которая уже является симптомом, нельзя назначить «подчинённые» симптоматические проблемы; любые проблемы, назначенные как симптомы симптоматической проблемы, станут симптомами той же причинной проблемы.

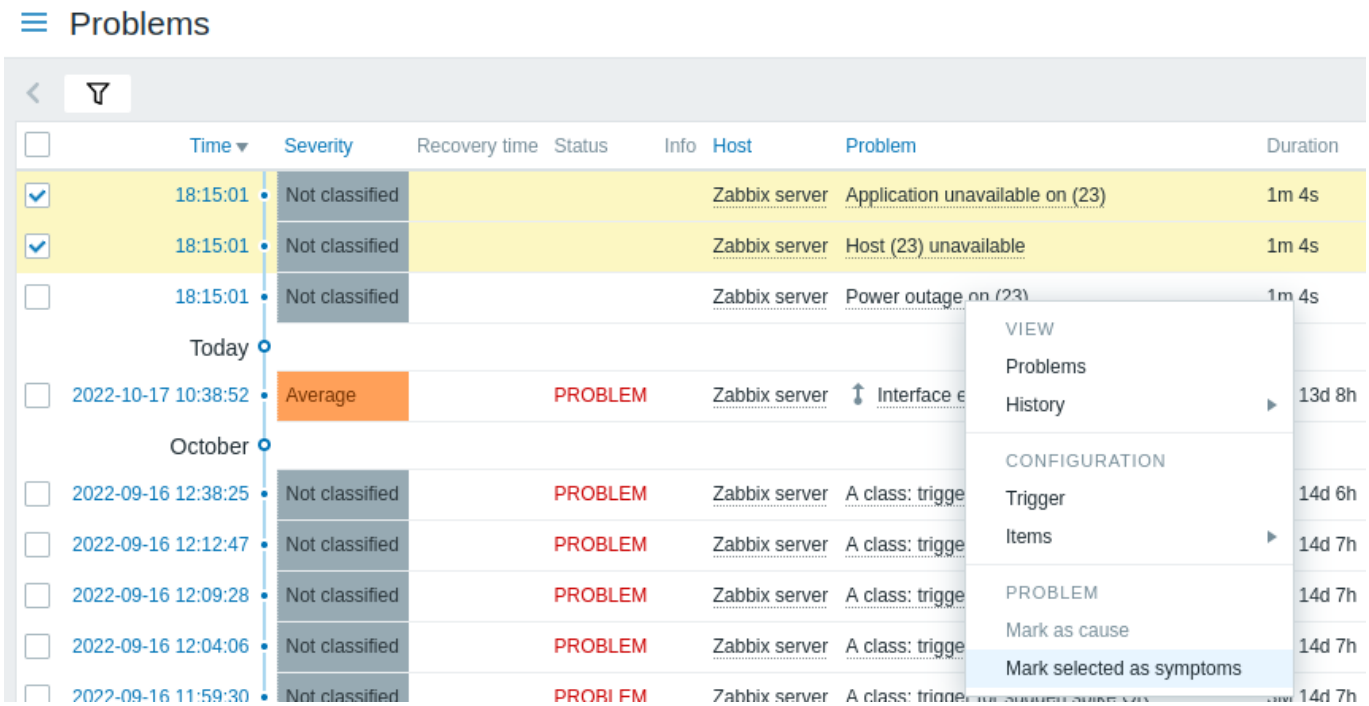
В общее количество проблем на картах, в виджетах панели мониторинга, таких как *Проблемы по важности* или *Узлы сети с проблемами*, и т. д., включаются только причинные проблемы. Однако ранжирование проблем не влияет на сервисы.

С симптоматической проблемой можно связать только одну причинную проблему. Симптоматические проблемы не устраняются автоматически, если причинная проблема устранена или закрыта.

Настройка

Чтобы переклассифицировать проблему в проблему-симптом, сначала выберите её в списке проблем. Можно выбрать одну или несколько проблем.

Затем перейдите к проблеме-причине и в её контекстном меню нажмите *Отметить выбранные как симптомы*.



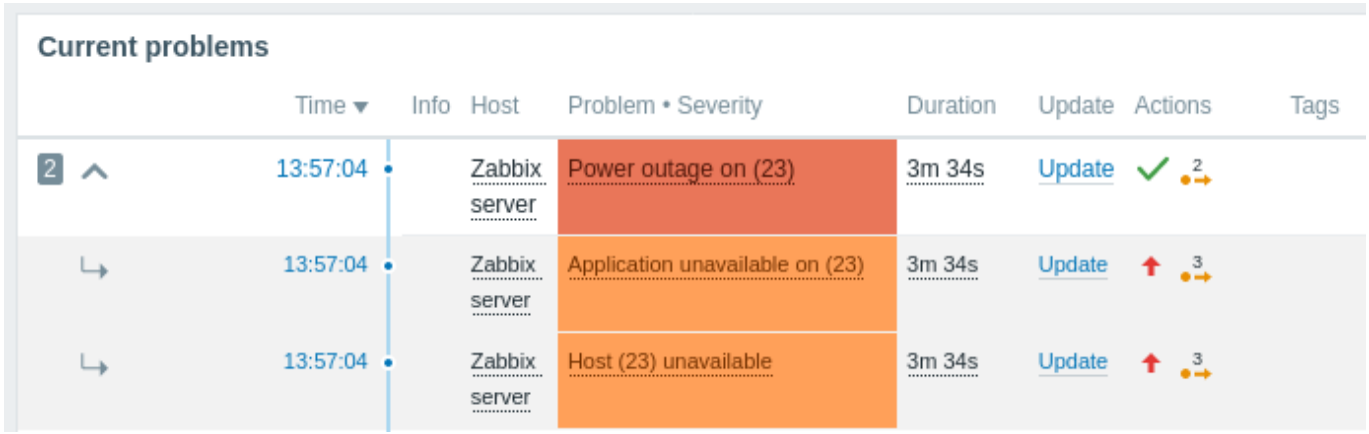
После этого выбранные проблемы будут обновлены сервером как проблемы-симптомы проблемы-причины.

Пока статус проблемы обновляется, она отображается одним из двух способов:

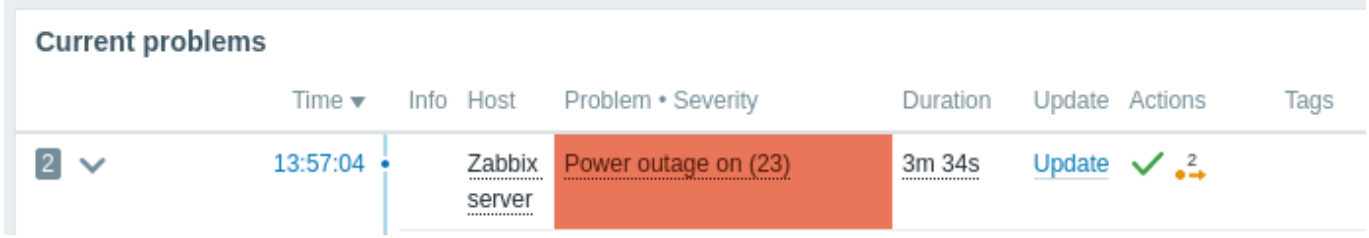
- В столбце Status отображается мигающий статус "UPDATING";
- В столбце Info отображается мигающий значок или (это действует, если в фильтре выбраны только *Problems*, и поэтому столбец Status не отображается).

Отображение

Проблемы-симптомы отображаются ниже проблемы-причины и соответствующим образом помечаются в *Мониторинг -> Проблемы* (а также в виджете панели *Проблемы*) — значком, меньшим шрифтом и другим фоном.



В свернутом виде видна только проблема-причина; о наличии проблем-симптомов свидетельствуют число в начале строки и значок для разворачивания представления.



Также можно дополнительно отображать проблемы-симптомы обычным шрифтом и в отдельной строке. Для этого выберите *Показывать симптомы* в настройках фильтра или конфигурации виджета.

Возврат к проблеме-причине

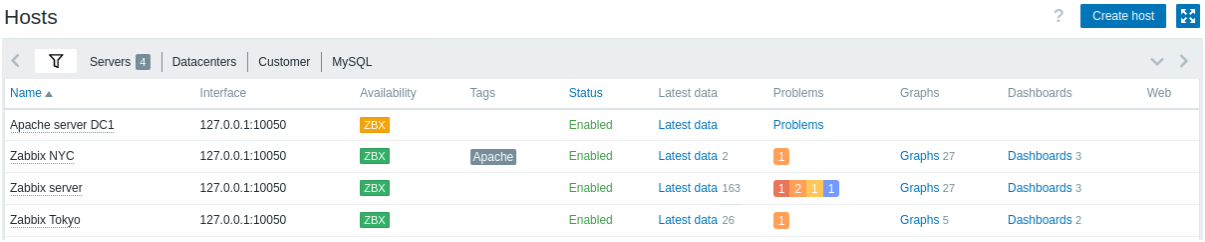
Проблему-симптом можно вернуть обратно в проблему-причину. Для этого выполните одно из следующих действий:

- нажмите *Отметить как причину* в контекстном меню проблемы-симптома;
- установите флажок *Преобразовать в причину* на экране обновления проблемы и нажмите *Обновить* (этот вариант также работает, если выбрано несколько проблем).

2 Узлы сети

Обзор

Раздел *Мониторинг* → *Узлы сети* отображает полный список отслеживаемых узлов сети с подробной информацией об интерфейсе узла сети, доступности, тегах, текущих проблемах, состоянии (включен/отключен), а также ссылками для удобного перехода к последним данным узла сети, истории проблем, графикам, панелям и веб-сценариям.



Столбец	Описание
Имя	Отображаемое имя узла сети. Щелчок по имени открывает меню узла сети. Оранжевый значок гаечного ключа после имени указывает, что этот узел сети находится на обслуживании. Щелкните по заголовку столбца, чтобы отсортировать узлы сети по имени в порядке возрастания или убывания.
Интерфейс	Отображается основной интерфейс узла сети.

Столбец	Описание
Доступность	Отображается доступность узла сети для каждого настроенного интерфейса. Значки доступности показывают текущее состояние интерфейса узла сети на сервере Zabbix. Поэтому, если вы отключите узел сети в 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 подробной информацией, состоянием и ошибками. Для интерфейса Agent во всплывающем окне отображаются интерфейсы (пассивные) и активные проверки. Если у узла сети есть только активные проверки, значок интерфейса Agent отображается, даже если у узла сети не настроен интерфейс Agent. Для узлов сети без интерфейсов столбец пуст. 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 было успешным; **Недоступен** — соединение с узлом сети было неуспешным (тайм-аут, проблемы с межсетевым экраном и т. д.); **Неизвестно** — попытка соединения с узлом сети не выполнялась или результат неизвестен. Дополнительные сведения о том, как сервер Zabbix определяет состояние интерфейса, см. в разделах Неизвестное состояние интерфейса и Настройки недостижимого/недоступного интерфейса узла сети. Состояние всех интерфейсов узла сети одного типа (Agent, SNMP, IPMI, JMX) определяется по тем интерфейсам, которые используются хотя бы одним включенным элементом данных. Состояние обозначается цветом значка: **Зеленый** — все интерфейсы доступны; **Желтый** — как минимум один интерфейс недоступен, и как минимум один доступен или имеет неизвестное состояние; **Красный** — все интерфейсы недоступны; **Серый** — как минимум один интерфейс имеет неизвестное состояние, 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 в потоке активных проверок агента отправляются heartbeat-сообщения. Частота heartbeat-сообщений управляется параметром `HeartbeatFrequency` в конфигурации Zabbix агент или агент 2 (по умолчанию 60 секунд, диапазон 0-3600). Активные проверки считаются недоступными, если heartbeat активных проверок старше 2 x HeartbeatFrequency секунд. **Примечание:** агенты Zabbix старше версии 6.2.x не отправляют heartbeat активных проверок, поэтому доступность их узлов сети остается неизвестной.
Теги	Теги узла сети и всех связанных шаблонов, с неразрешенными макросами.
Состояние	Состояние узла сети — *Включен* или *Отключен*. Щелкните по заголовку столбца, чтобы отсортировать узлы сети по состоянию в порядке возрастания или убывания.
Последние данные	Щелчок по ссылке откроет страницу *Мониторинг* - *Последние данные* со всеми последними данными, собранными с узла сети. Количество элементов данных с последними данными отображается серым цветом.

Столбец	Описание
Проблемы	Количество открытых проблем узла сети, отсортированных по важности. Цвет квадрата указывает на важность проблемы. Число в квадрате означает количество проблем данной важности. Щелчок по значку откроет страницу *Мониторинг* - *Проблемы* для текущего узла сети. Если у узла сети нет проблем, ссылка на раздел *Проблемы* для этого узла сети отображается как текст. Используйте фильтр, чтобы выбрать, следует ли включать подавленные проблемы (по умолчанию не включаются).
Графики	Щелчок по ссылке отобразит графики, настроенные для узла сети. Количество графиков отображается серым цветом. Если у узла сети нет графиков, ссылка не отображается.
Панели	Щелчок по ссылке отобразит панели, настроенные для узла сети. Количество панелей отображается серым цветом. Если у узла сети нет панелей, ссылка не отображается.
Веб	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.

Кнопки

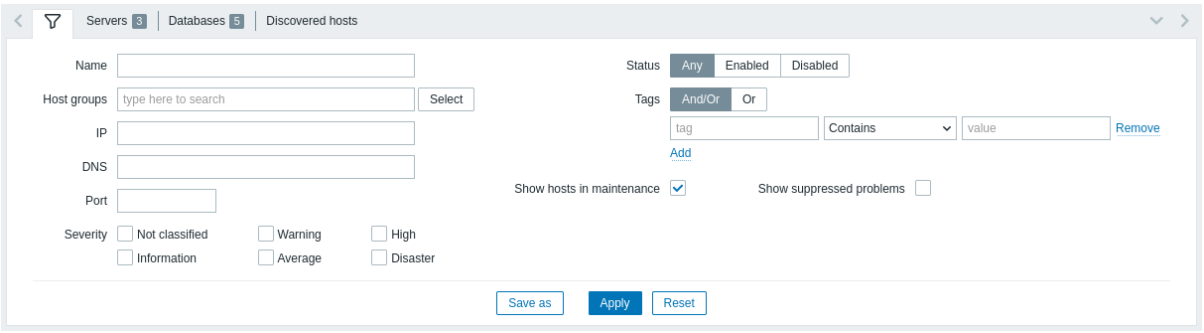
Создать узел сети позволяет создать новый узел сети. Эта кнопка доступна только для пользователей Admin и Super Admin.

Кнопки режима просмотра, общие для всех разделов, описаны на странице Мониторинг.

Использование фильтра

Вы можете использовать фильтр, чтобы отображать только те узлы сети, которые вас интересуют. Для повышения производительности поиска данные ищутся с неразрешёнными макросами.

Фильтр расположен над таблицей. Можно фильтровать узлы сети по имени, группе узлов сети, IP или DNS, порту интерфейса, тегам, серьёзности проблем, статусу (включён/отключён/любой); также можно выбрать, отображать ли подавленные проблемы и узлы сети, которые в данный момент находятся на обслуживании.



Параметр	Описание
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.
Severity	Фильтрация по серьёзности проблем. По умолчанию отображаются проблемы всех уровней серьёзности. Проблемы отображаются, если они не подавлены.
Status	Фильтрация по статусу узла сети.

Параметр	Описание
Tags	Фильтрация по имени и значению тега узла сети. Узлы сети можно фильтровать по тегам уровня узла сети, а также по тегам из всех связанных шаблонов, включая вложенные шаблоны. Можно как включать, так и исключать определённые 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 учётом регистра) **Contains** — включить указанные имена тегов, где значения тегов содержат введённую строку (поиск подстроки, без учёта регистра) **Does not exist** — исключить указанные имена тегов **Does not equal** — исключить указанные имена и значения тегов (с учётом регистра) **Does not contain** — исключить указанные имена тегов, где значения тегов содержат введённую строку (поиск подстроки, без учёта 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).
Show suppressed problems	Установите флажок, чтобы отображать проблемы, которые в противном случае были бы подавлены (не показаны) из-за обслуживания узла сети или отдельного подавления проблем.

Сохранение фильтра

Избранные настройки фильтра можно сохранить как вкладки, а затем быстро открывать, нажимая соответствующую вкладку над фильтром.

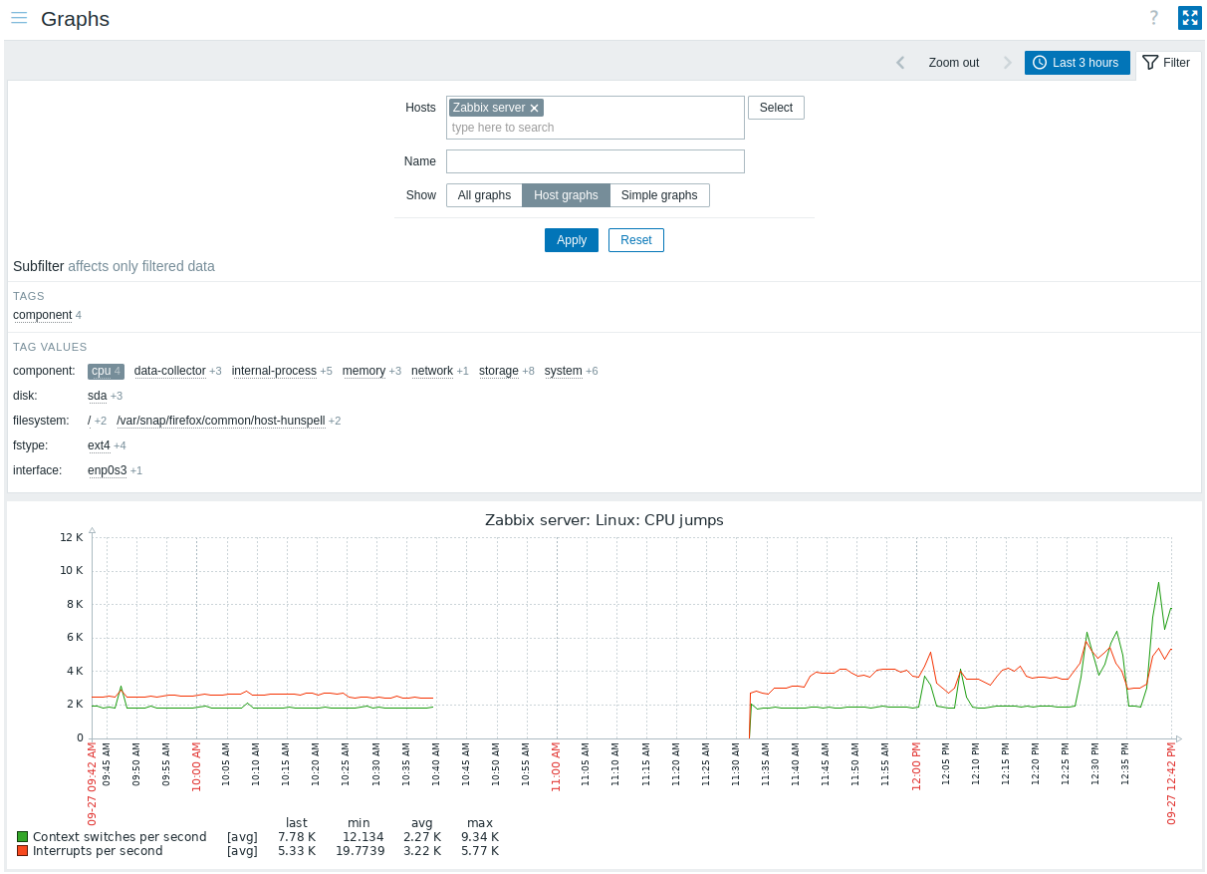
Подробнее см. в разделе сохранение фильтров.

1 Графики

Обзор

К графикам узла сети можно перейти из *Мониторинг* → *Узлы сети*, щелкнув по пункту «Графики» для соответствующего узла сети.

Можно отобразить 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.

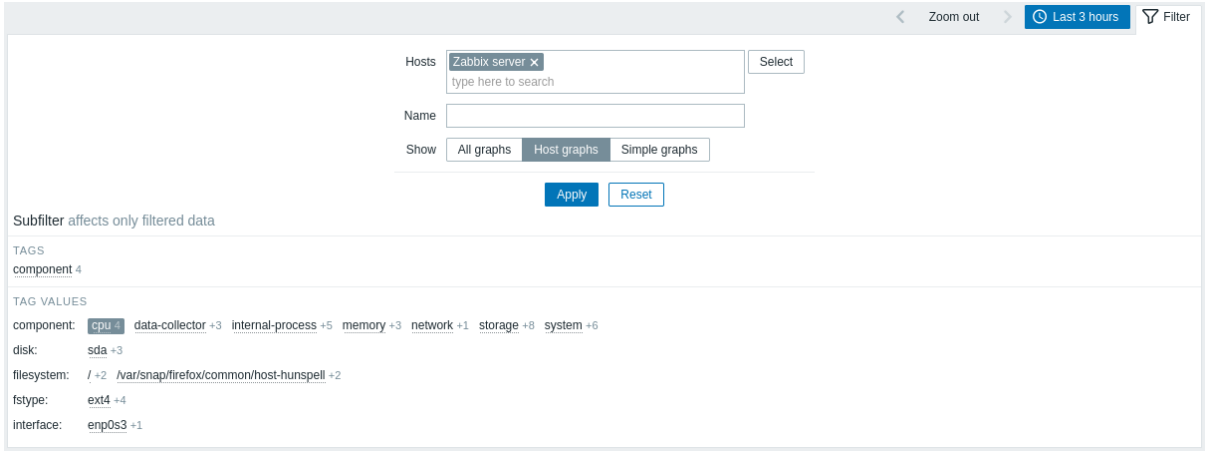
Селектор периода времени

Обратите внимание на селектор периода времени над графиком. Он позволяет выбирать часто требуемые периоды одним щелчком мыши.

См. также: *Селектор периода времени*

Использование фильтра

Чтобы просмотреть определённый график, выберите его в фильтре. Фильтр позволяет указать узел сети, имя графика и параметр *Показать* (все/графики узла сети/простые графики).



Если в фильтре не выбран узел сети, графики не отображаются.

Использование подфильтра

Подфильтр позволяет дополнительно изменять фильтрацию из основного фильтра.

Он содержит кликабельные ссылки для быстрого доступа к связанным графикам. Графики связаны общей сущностью — именем тега или его значением. При нажатии на имя/значение тега оно подсвечивается серым фоном, и графики немедленно фильтруются (нет необходимости нажимать *Применить* в основном фильтре). Нажатие на другое имя/значение тега добавляет его в отфильтрованные результаты. Повторное нажатие на имя/значение тега убирает фильтрацию.

Подфильтры формируются на основе отфильтрованных данных, которые ограничены 1000 записями. Если вы хотите видеть больше записей в подфильтре, необходимо увеличить значение параметра *Limit for search and filter results* (в *Administration -> General -> GUI*).

В отличие от основного фильтра, подфильтр обновляется вместе с каждым запросом на обновление таблицы, чтобы всегда получать актуальную информацию о доступных вариантах фильтрации и их количестве.

Количество сущностей, отображаемых по горизонтали, ограничено 100. Если их больше, в конце отображается значок с тремя точками; он не является кликабельным. Вертикальные списки (например, теги с их значениями) ограничены 20 записями. Если их больше, отображается значок с тремя точками; он не является кликабельным.

Число рядом с каждой кликабельной сущностью указывает количество графиков, которые она содержит в результатах основного фильтра.

После выбора одной сущности числа у других доступных сущностей отображаются со знаком плюс, указывающим, сколько графиков может быть добавлено к текущему выбору.

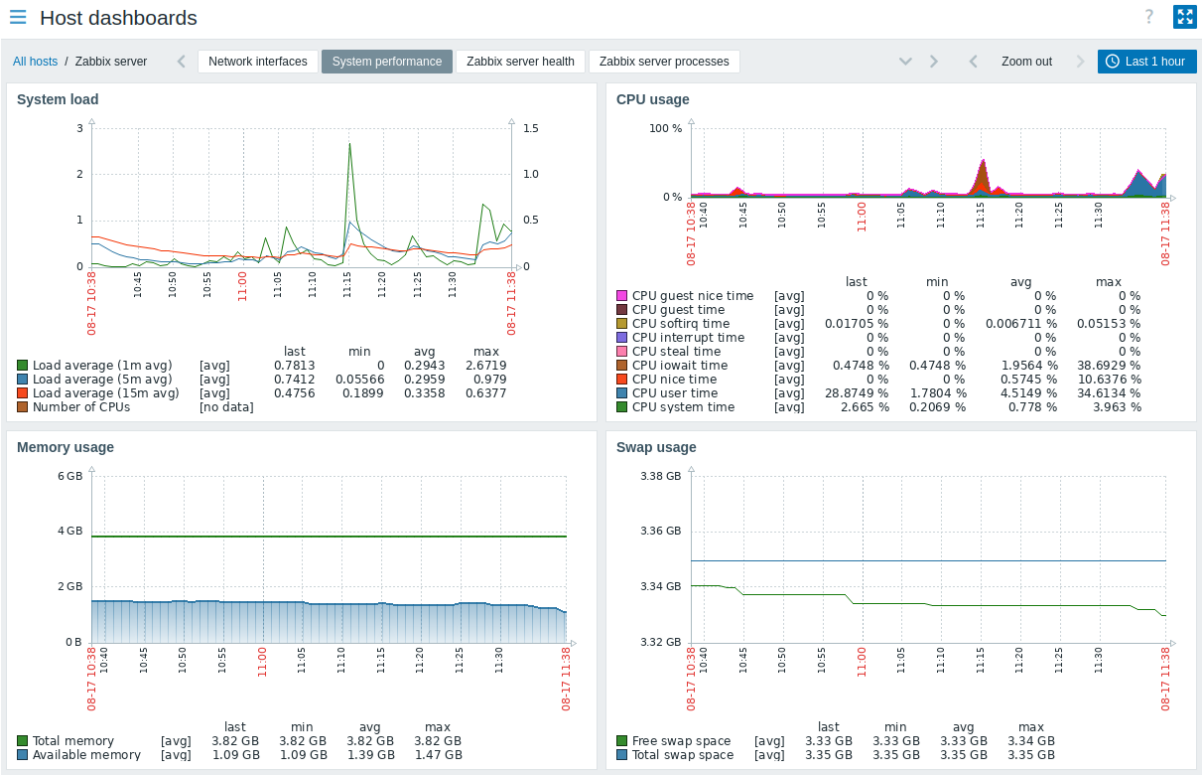
Кнопки

Кнопки режима просмотра, общие для всех разделов, описаны на странице Мониторинг.

2 Панели узла сети

Обзор

Панели узла сети похожи на глобальные панели; однако у панелей узла сети нет владельца, и они отображают данные только для выбранного узла сети.



При просмотре панелей узла сети вы можете переключаться между настроенными панелями, нажимая:

- вкладки панели;
- кнопки со стрелками  под заголовком;
- кнопку со стрелкой  под заголовком, которая отобразит полный список доступных панелей узла сети.

Чтобы перейти в раздел *Мониторинг → Узлы сети*, нажмите навигационную ссылку *Все узлы сети* под заголовком в левом верхнем углу.

Настройка

Панели узлов сети настраиваются на уровне шаблона. После привязки шаблона к узлу сети для этого узла сети создаются панели. Обратите внимание, что панели узлов сети *нельзя* настраивать в разделе *Панели* веб-интерфейса, который предназначен для глобальных панелей.

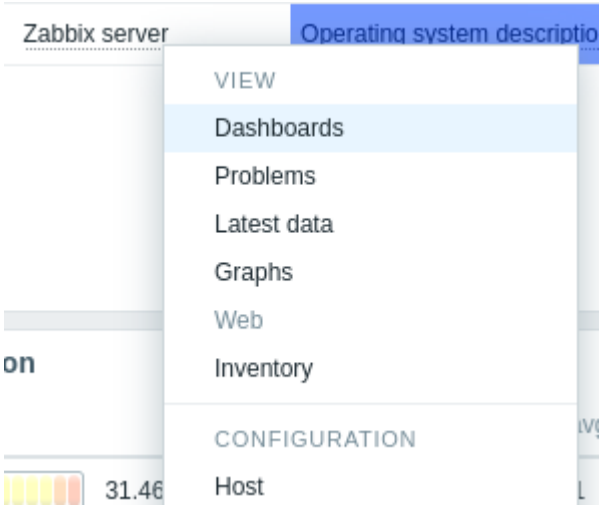
Виджеты панелей узлов сети также можно настраивать только на уровне шаблона, за исключением изменения интервала обновления. Кроме того, виджеты панелей узлов сети можно копировать только в другие панели узлов сети в пределах того же шаблона. Обратите внимание, что виджеты из глобальных панелей *нельзя* копировать в панели узлов сети.

Note:
Панели узлов сети до Zabbix 5.2 назывались экранами узлов сети. При импорте старого шаблона, содержащего экраны, импорт экранов будет проигнорирован.

Доступ

Доступ к панелям узла сети можно получить:

- после поиска имени узла сети в глобальном поиске (нажмите ссылку *Dashboards* в результатах поиска);
- после нажатия на имя узла сети в *Инвентаризация → Узлы сети* (нажмите ссылку *Dashboards* в обзоре узла сети);
- из меню узла сети, нажав *Dashboards*.

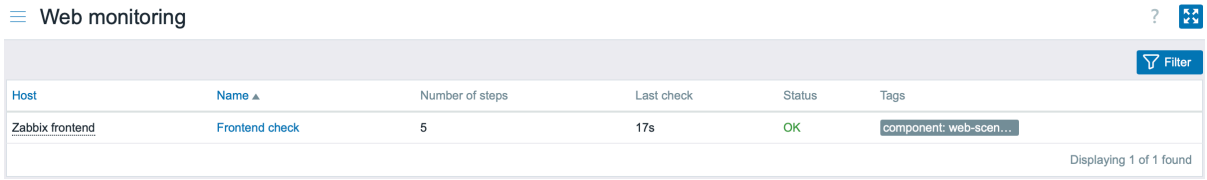


Обратите внимание, что к панелям узла сети *нельзя* получить прямой доступ в разделе *Панели*, который предназначен для глобальных панелей.

3 Веб-сценарии

Обзор

Информация о веб-сценарии узла сети доступна в разделе *Мониторинг → Узлы сети*, если нажать на Web для соответствующего узла сети.

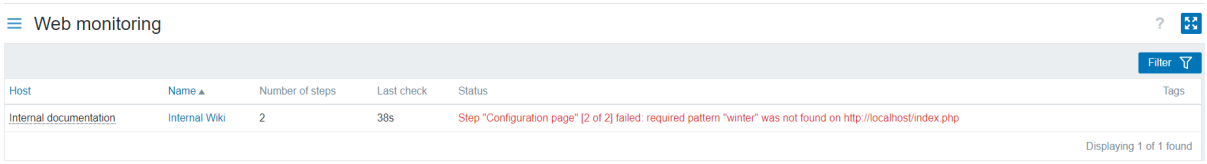


При нажатии на имя узла сети открывается меню узла сети. Данные отключенных узлов сети также доступны. Имя отключенного узла сети отображается красным цветом.

Статус веб-сценария может быть следующим:

- OK (зеленый) — все шаги успешно выполнены, и Zabbix собрал данные по сценарию.
- Error (красный) — один или несколько шагов завершились ошибкой (см. сообщение об ошибке, чтобы понять, какие параметры, возможно, требуют исправления).

- Unknown (серый) — статус шагов пока 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 за последние 24 часа. Это ограничение было введено для улучшения времени первоначальной загрузки больших страниц с последними данными. Вы можете увеличить этот период времени, изменив значение параметра *Max history display period* в разделе меню *Администрирование → Общие → 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 *Мониторинг → Узлы сети*, выберите этот узел сети или группу в фильтре. Вы также можете фильтровать сценарии по тегам.

Кнопки

Кнопки режима просмотра, являющиеся общими для всех разделов, описаны на странице Мониторинг.

3 Последние данные

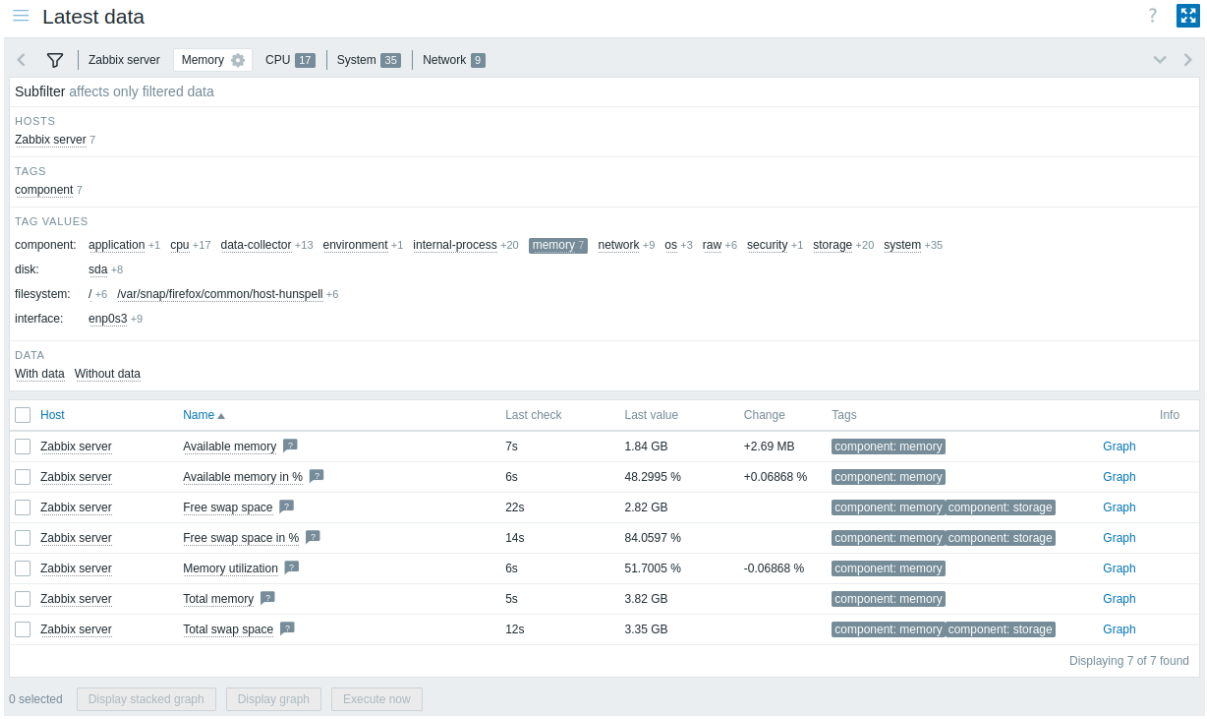
Обзор

Раздел *Мониторинг → Последние данные* отображает последние значения, собранные элементами данных.

Этот раздел содержит следующие элементы:

- Фильтр
- Подфильтр
- Список элементов данных

Note:
Подфильтр и список элементов данных отображаются только в том случае, если задан фильтр и имеются результаты для отображения.



Column	Description
Узел сети	Имя узла сети, которому принадлежит элемент данных. При нажатии на имя открывается контекстное меню узла сети. Если узел сети находится на 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 *Последние данные*.
Имя	Имя элемента данных. При нажатии на имя открывается меню элемента данных. Значок вопросительного знака отображается рядом с именем элемента данных для всех элементов данных, у которых есть описание. Наведите указатель мыши на значок, чтобы отобразить всплывающую подсказку с описанием элемента данных.
Последняя проверка	Время, прошедшее с момента последней проверки элемента данных.

Column	Description
Последнее значение	Самое последнее значение элемента данных. Значения отображаются с применением преобразования единиц измерения и сопоставления значений. Наведите указатель мыши на значение, чтобы отобразить исходные данные (усеченные до 8192 символов). Двоичные значения отображаются в виде заполнителя (*binary value*) вместо фактического значения. По умолчанию отображаются только значения, полученные за последние 24 часа. Это ограничение улучшает время первоначальной загрузки больших страниц последних данных; чтобы увеличить этот период, обновите значение параметра *Max history display period* в *Администрирование* → *Общие* → *GUI*.
Изменение	Разница между предыдущим значением и самым последним значением. Для элементов данных с частотой обновления 1 день или более величина изменения никогда не будет отображаться (при настройках по умолчанию). В этом случае последнее значение вообще не будет отображаться, если оно было получено более 24 часов назад.
Теги	Теги, связанные с элементом данных. Теги в списке элементов данных являются кликабельными. При нажатии на тег он включается в подфильтре, в результате чего в списке элементов данных отображаются только элементы данных, содержащие этот тег (и любые другие теги, ранее выбранные в подфильтре). Обратите внимание, что после такой фильтрации элементов данных теги в списке элементов данных больше не являются кликабельными. Дальнейшее изменение на основе тегов (например, для удаления тегов или указания других фильтров) должно выполняться в подфильтре.
График/История	Ссылка на простой график/историю значений элемента данных.
Инфо	Дополнительная информация об элементе данных. Если у элемента данных есть ошибки (например, он стал неподдерживаемым), отображается значок информации. Наведите указатель мыши на значок для получения подробностей.

Кнопки

Кнопки режима просмотра, общие для всех разделов, описаны на странице Мониторинг.

Массовые действия

Кнопки под списком позволяют выполнять массовые действия с одним или несколькими выбранными элементами данных:

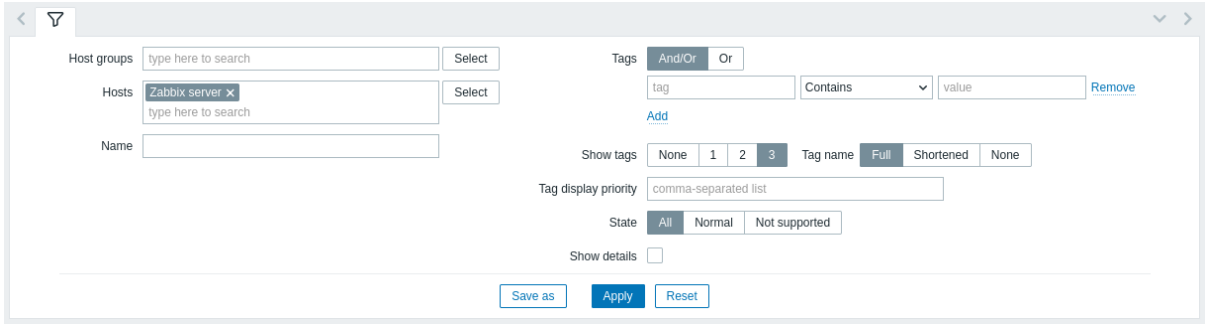
- *Показать составной график* - отобразить составной ad-hoc graph.
- *Показать график* - отобразить простой ad-hoc graph.
- *Выполнить сейчас* - немедленно выполнить проверку для получения новых значений элемента данных. Поддерживается только для **пассивных** проверок (см. подробнее). Эта опция доступна только для узлов сети с доступом на чтение и запись. Доступность этой опции для узлов сети с правами только на чтение зависит от настройки роли пользователя *Вызывать "Выполнить сейчас" на узлах сети только для чтения*.

Чтобы использовать эти параметры, установите флажки напротив соответствующих элементов данных, затем нажмите нужную кнопку.

Использование фильтра

Вы можете использовать фильтр, чтобы отображать только те элементы данных, которые вас интересуют. Для повышения производительности поиска данные ищутся с неразрешёнными макросами.

Значок фильтра  расположен над списком элементов данных и подфильтром. Щёлкните по нему, чтобы развернуть фильтр.



Фильтр позволяет сузить список элементов данных по группе узлов сети, узлу сети, имени элемента данных, тегу, состоянию и другим параметрам. При указании родительской группы узлов сети в фильтре неявно выбираются все вложенные группы узлов сети. Подробную информацию о фильтрации по тегам смотрите в разделе *Мониторинг* → *Проблемы*.

Параметр фильтра *Показывать подробности* позволяет расширить информацию, отображаемую для элементов данных. Отметьте его, чтобы отображать такие сведения, как интервал обновления элемента данных, настройки истории и трендов, тип элемента данных и ошибки элемента данных (исправен/неподдерживаемый).

Сохранение фильтра

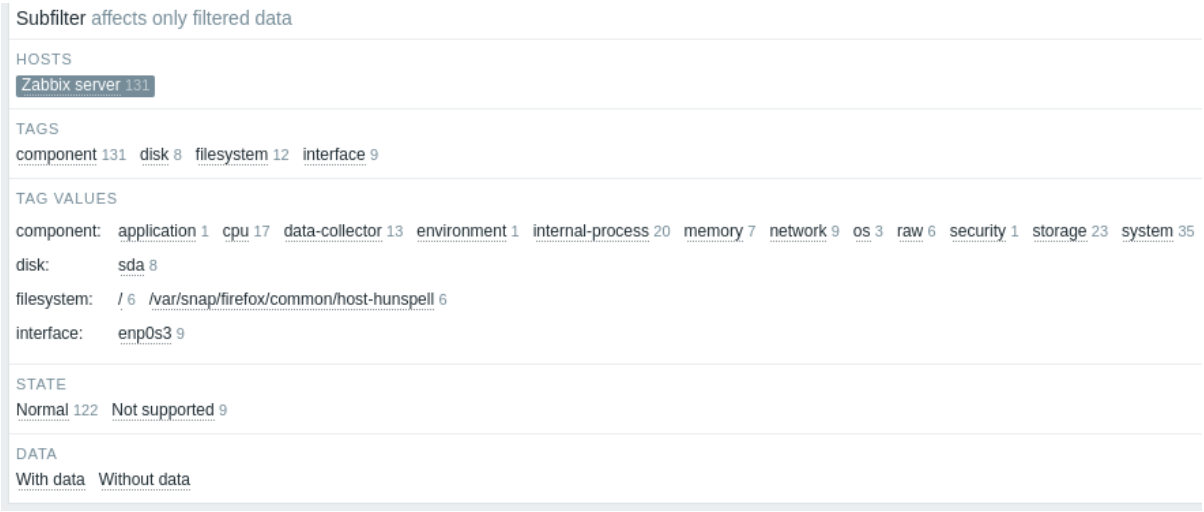
Настройки избранного фильтра можно сохранить в виде вкладок, а затем быстро получать к ним доступ, щёлкнув по соответствующей вкладке над фильтром.

Подробнее см. в разделе сохранение фильтров.

Использование подфильтра

Подфильтр позволяет дополнительно изменять фильтрацию, заданную в основном фильтре.

Он содержит кликабельные ссылки для быстрого доступа к связанным элементам данных. Элементы данных связаны общей сущностью — узлом сети, именем тега или его значением, состоянием элемента данных или статусом данных. При нажатии на сущность она выделяется серым фоном, а элементы данных немедленно фильтруются (нажимать *Применить* в основном фильтре не требуется). Нажатие на другую сущность добавляет её к отфильтрованным результатам. Повторное нажатие на сущность убирает фильтрацию.



Подфильтры формируются на основе отфильтрованных данных, количество которых ограничено 1000 записями. Если имеется 20 узлов сети, каждый из которых содержит 100 элементов данных (то есть всего 2000 записей), в подфильтре будет видна только половина узлов сети. Если вы хотите видеть в подфильтре больше записей, необходимо увеличить значение параметра *Limit for search and filter results* (в *Administration -> General -> GUI*).

В отличие от основного фильтра, подфильтр обновляется при каждом запросе на обновление таблицы, чтобы всегда содержать актуальную информацию о доступных вариантах фильтрации и количестве элементов данных для каждого из них.

Для каждой группы сущностей (узлы сети, теги, значения тегов и т. д.) отображается до 10 строк сущностей. Если сущностей больше, этот список можно развернуть для отображения максимум 1000 записей (значение SUBFILTER_VALUES_PER_GROUP в frontend definitions), нажав значок с тремя точками ••• в конце списка. Для *Tag values* список можно развернуть для отображения максимум 200 имен тегов с соответствующими им значениями. Обратите внимание, что после полного разворачивания список нельзя свернуть обратно.

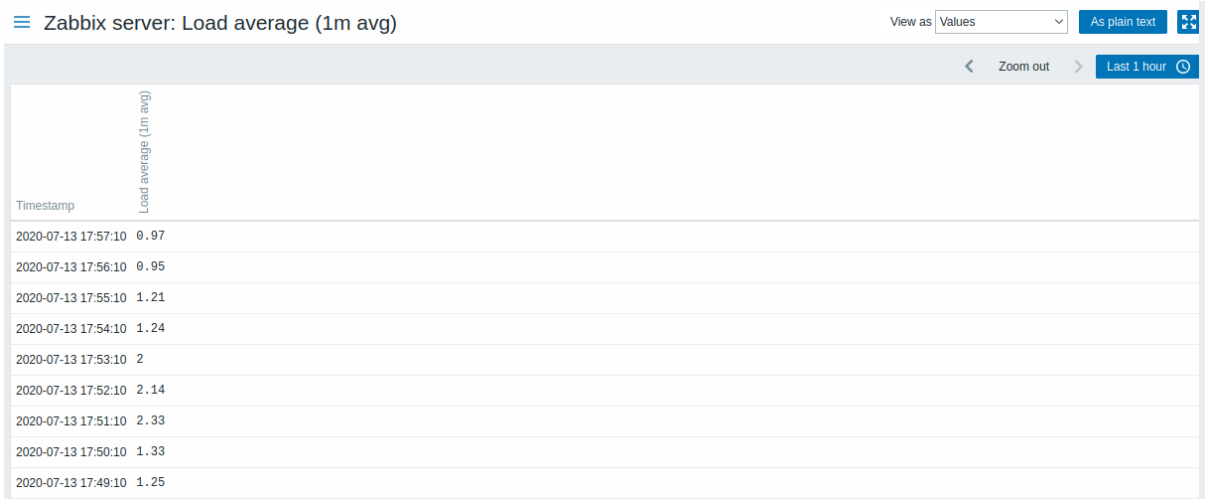
Число рядом с каждой кликабельной сущностью указывает количество элементов данных, сгруппированных в ней (на основе результатов основного фильтра). При нажатии на сущность числа у других доступных сущностей отображаются со знаком «плюс», указывающим, сколько элементов данных может быть добавлено к текущему выбору. Сущности без элементов данных не отображаются, если только они не были ранее выбраны в подфильтре.

Графики и история

Столбец *График/История* в списке элементов данных содержит следующие ссылки:

- **История** — для всех текстовых элементов данных; ведет к спискам (*Значения/500 последних значений*), отображающим историю предыдущих значений элемента данных.

- **График** — для всех числовых элементов данных; ведет к простому графику. Обратите внимание, что при отображении графика выпадающий список в правом верхнем углу также позволяет переключиться на *Значения/500 последних значений*.



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*, заданного в разделе *Administration → General → GUI*.

4 Карты

Обзор

В разделе *Мониторинг → Карты сети* вы можете настраивать, управлять и просматривать карты сети.

При открытии этого раздела вы увидите либо последнюю карту, к которой обращались, либо список всех карт, к которым у вас есть доступ.

Все карты могут быть как общедоступными, так и частными. Общедоступные карты доступны всем пользователям, тогда как частные карты доступны только их владельцу и тем пользователям, с которыми карта была совместно использована.

Список карт



Отображаемые данные:

Столбец	Описание
Имя	Имя карты. Нажмите на имя, чтобы просмотреть карту.
Ширина	Отображается ширина карты.
Высота	Отображается высота карты.
Действия	Доступны два действия: **Свойства** - настройка общих свойств карты **Редактировать** - переход к сетке для добавления элементов карты

Чтобы настроить новую карту, нажмите кнопку *Создать карту* в правом верхнем углу. Чтобы импортировать карту из файла YAML, XML или JSON, нажмите кнопку *Импорт* в правом верхнем углу. Пользователь, импортировавший карту, будет назначен её владельцем.

Две кнопки под списком предоставляют несколько вариантов массового редактирования:

- *Экспорт* - экспортировать карты в файл YAML, XML или JSON

- *Удалить* - удалить карты

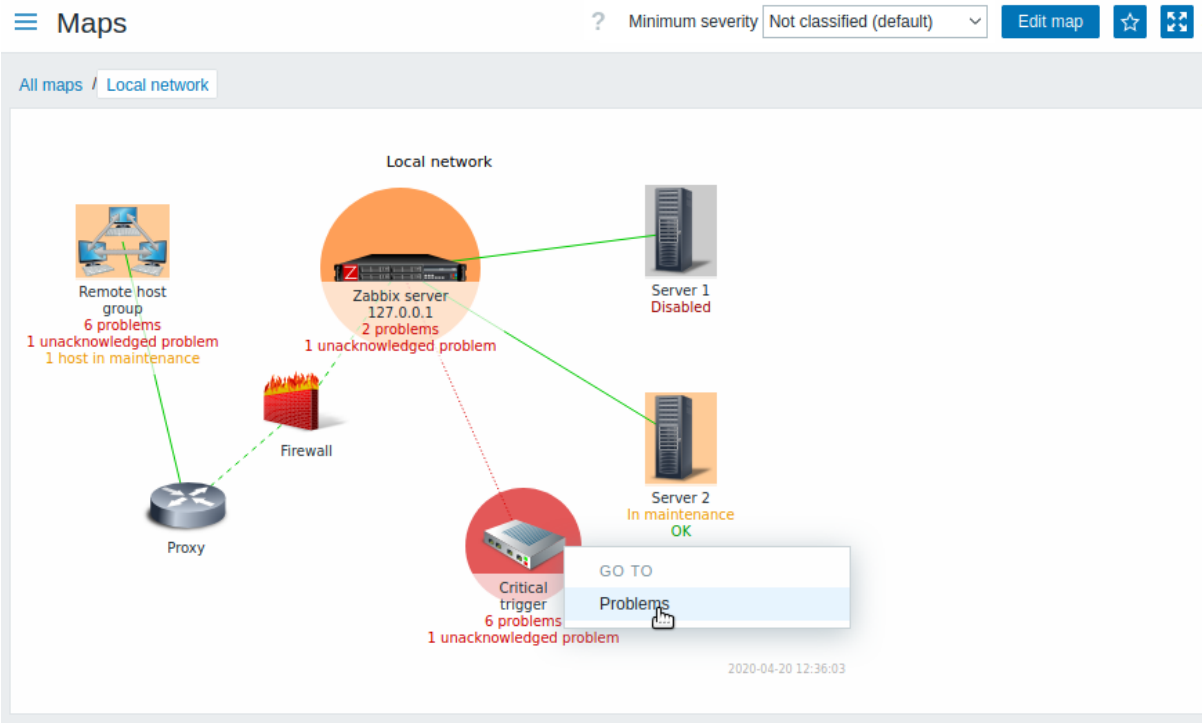
Чтобы использовать эти параметры, установите флажки напротив соответствующих карт, затем нажмите нужную кнопку.

Использование фильтра

Вы можете использовать фильтр, чтобы отображать только те карты, которые вас интересуют. Для повышения производительности поиска данные ищутся с неразрешёнными макросами.

Просмотр карт

Чтобы просмотреть карту, нажмите на её имя в списке всех карт.



Вы можете использовать выпадающий список в строке заголовка карты, чтобы выбрать самый низкий уровень серьезности проблемных триггеров для отображения. Уровень серьезности, отмеченный как *по умолчанию*, — это уровень, заданный в настройках карты. Если карта содержит вложенную карту, при переходе к вложенной карте будет сохранён уровень серьезности карты более высокого уровня (за исключением случая, когда это *Не классифицировано* — в этом случае он не будет передан во вложенную карту).

Подсветка иконок

Если элемент карты находится в состоянии проблемы, он подсвечивается круглым кружком. Цвет заливки кружка соответствует цвету уровня серьезности проблемы. Вместе с элементом будут отображаться только проблемы на выбранном уровне серьезности или выше. Если все проблемы подтверждены, то вокруг кружка отображается толстая зеленая рамка.

Дополнительно:

- узел сети в обслуживании подсвечивается оранжевым закрашенным квадратом. Обратите внимание, что подсветка обслуживания имеет приоритет над подсветкой уровня серьезности проблемы, если элемент карты — это узел сети.
- отключенный (не отслеживаемый) узел сети подсвечивается серым закрашенным квадратом.

Подсветка отображается, если установлен флажок *Подсветка иконок* в настройках карты.

Маркеры недавних изменений

Красные треугольники, направленные внутрь вокруг элемента данных, указывают на недавнее изменение состояния триггера — произошедшее в течение последних 30 минут. Эти треугольники отображаются, если установлен флажок *Mark elements on trigger status change* в настройках карты.

Ссылки

Щелчок по элементу карты открывает меню с некоторыми доступными ссылками. Щелчок по имени узла сети вызывает меню узла сети.

Кнопки

Кнопки справа предоставляют следующие возможности:

Edit map	Перейти к редактированию содержимого карты.
☆	Добавить карту в виджет избранного на Панелях.
★	Карта находится в виджете избранного на Панелях. Нажмите, чтобы удалить карту из виджета избранного.

Кнопки режима просмотра, общие для всех разделов, описаны на странице Мониторинг.

Удобочитаемая сводка на картах

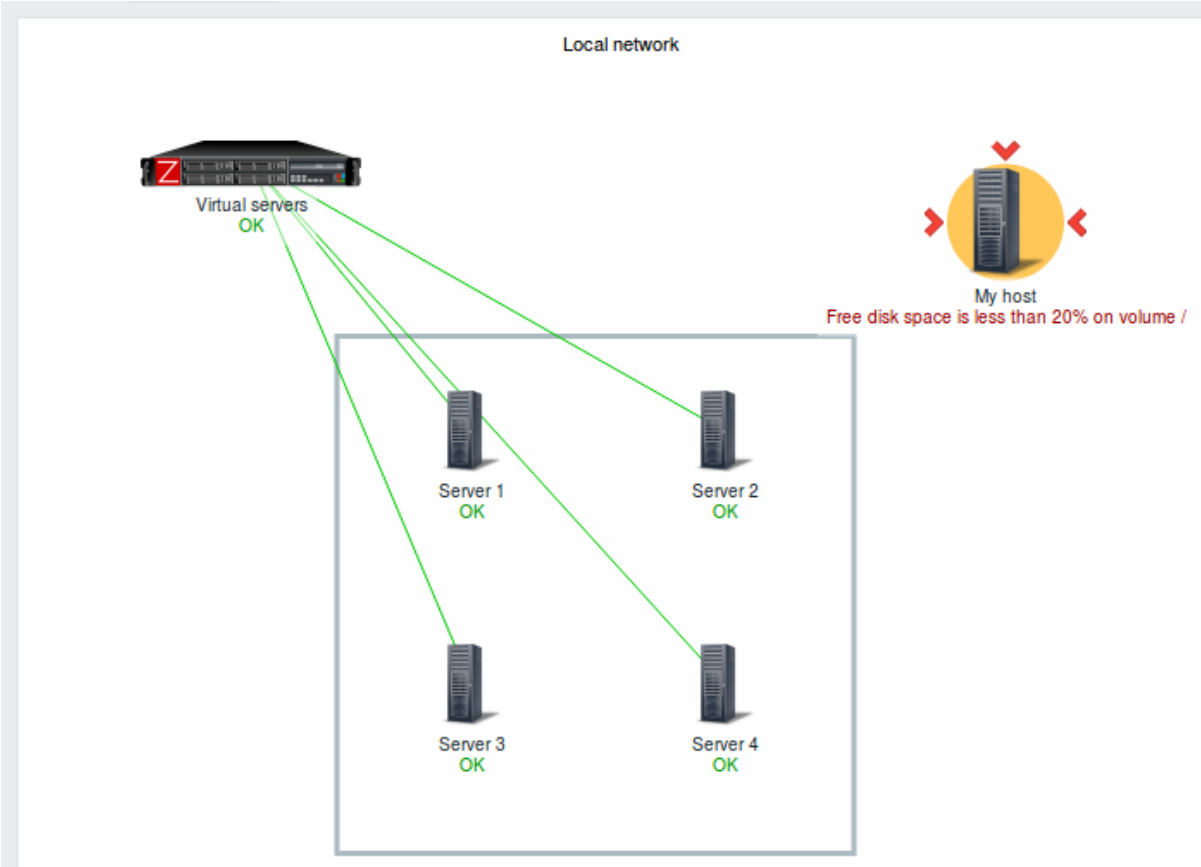
Доступно скрытое свойство "aria-label", позволяющее считывать информацию карты с помощью программы экранного доступа. Доступны как общее описание карты, так и описание отдельных элементов в следующем формате:

- 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        .      ,         , My host,                  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`

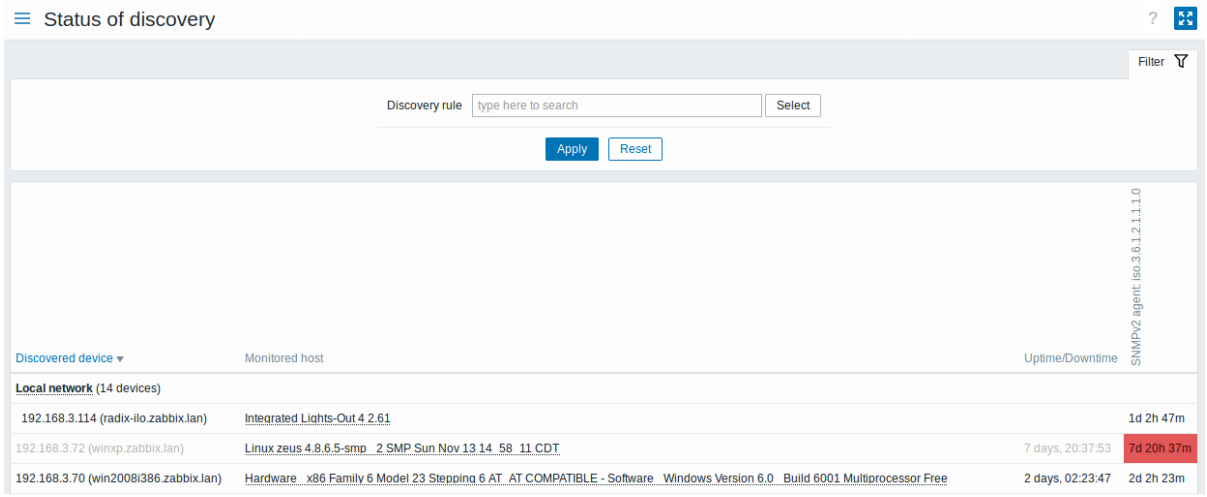
откроет карту с таким именем (Local network).

Если указаны оба параметра — `sysmapid` (ID карты) и `mapname` (имя карты), то `mapname` имеет более высокий приоритет.

5 Обнаружение

Обзор

В разделе *Мониторинг* → *Обнаружение* отображаются результаты обнаружения сети. Обнаруженные устройства сортируются по правилу обнаружения.



Отображаемые данные:

Столбец	Описание
Обнаруженное устройство	Обнаруженные устройства перечислены и сгруппированы по правилу обнаружения. Щелчок по правилу обнаружения открывает меню правила, содержащее ссылку на форму настройки правила обнаружения.
Отслеживаемый узел сети	Если устройство уже находится под мониторингом, в этом столбце будет указано имя узла сети. Щелчок по имени узла сети открывает меню узла сети.
Время работы/Простоя	В этом столбце отображается длительность состояния, в котором устройство было обнаружено или потеряно после предыдущего обнаружения.
Проверка обнаружения	Отображается состояние отдельного сервиса (проверки обнаружения) для каждого обнаруженного устройства. Красная ячейка показывает, что сервис недоступен. В ячейке также указывается время работы или простоя сервиса. Этот столбец отображается только в том случае, если сервис был найден хотя бы на одном обнаруженном устройстве.

Кнопки

Кнопки режима просмотра, являющиеся общими для всех разделов, описаны на странице Мониторинг.

Использование фильтра

Вы можете использовать фильтр, чтобы отображать только те правила обнаружения, которые вас интересуют. Для повышения производительности поиска данные ищутся с неразрешёнными макросами.

Если в фильтре ничего не выбрано, отображаются все включённые правила обнаружения. Чтобы выбрать для отображения определённое правило обнаружения, начните вводить его имя в фильтре. Для выбора будут перечислены все совпадающие включённые правила обнаружения. Можно выбрать более одного правила обнаружения.

3 Сервисы

Обзор

Меню ”Услуги” предназначено для функций мониторинга услуг Zabbix.

1 Сервисы

Обзор

В этом разделе вы можете увидеть общее состояние всех сервисов, которые были настроены в Zabbix, на основе вашей инфраструктуры.

Сервис может представлять собой иерархию, состоящую из нескольких уровней других сервисов, называемых дочерними сервисами, которые влияют на общее состояние сервиса (см. также обзор функциональности мониторинга сервисов.)

Основными категориями состояния сервиса являются *OK* или *Проблема*, где состояние *Проблема* выражается соответствующим названием и цветом важности проблемы.

Хотя режим просмотра позволяет отслеживать сервисы с их состоянием и другими сведениями, вы также можете настроить иерархию сервисов в этом разделе (добавлять/редактировать сервисы, дочерние сервисы), переключившись в режим редактирования.

Чтобы переключиться из режима просмотра в режим редактирования (и обратно), нажмите соответствующую кнопку в правом верхнем углу:

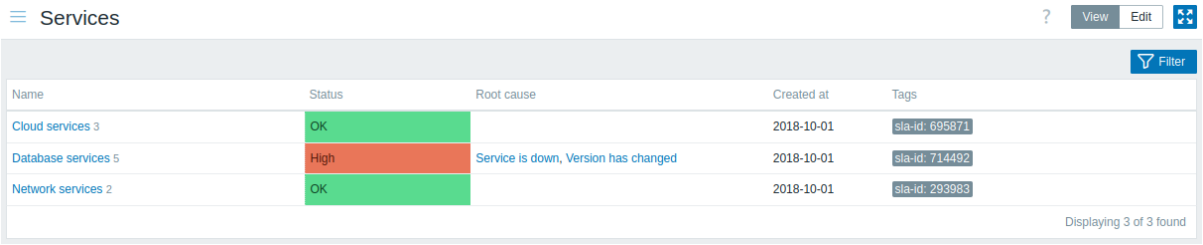
- 

- просмотр сервисов
- 

- добавление/редактирование сервисов и дочерних сервисов

Обратите внимание, что доступ к редактированию зависит от настроек роли пользователя.

Просмотр сервисов



Отображается список существующих сервисов.

Отображаемые данные:

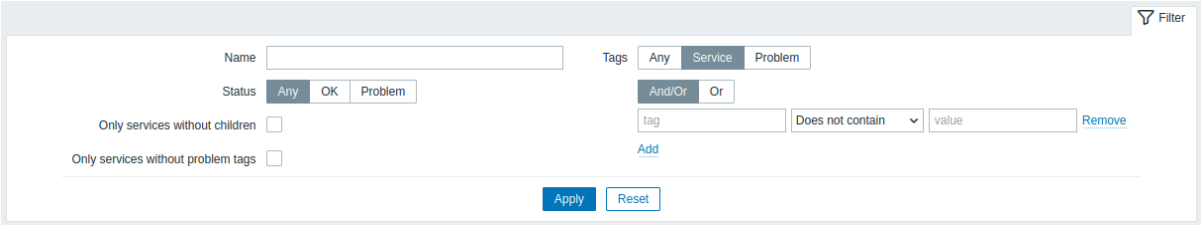
Параметр	Описание
Name	Имя сервиса. Имя сервиса является ссылкой на сведения о сервисе. Число после имени указывает, сколько у сервиса дочерних сервисов.
Status	Состояние сервиса: **OK** — проблем нет **<цвет проблемы и серьезность>** — указывает на проблему и ее серьезность. При наличии нескольких проблем отображаются цвет и серьезность наиболее критичной проблемы.
Root cause	Перечислены первопричины, которые прямо или косвенно влияют на состояние сервиса. Эти же проблемы перечисляются так же, как их возвращает макрос {SERVICE.ROOTCAUSE}. Щелкните по имени проблемы, чтобы увидеть более подробную информацию о ней в *Мониторинг → Проблемы*. Проблемы, которые не влияют на состояние сервиса, в список не включаются.
Created at	Отображается время создания сервиса.
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 те сервисы, которые вас интересуют.

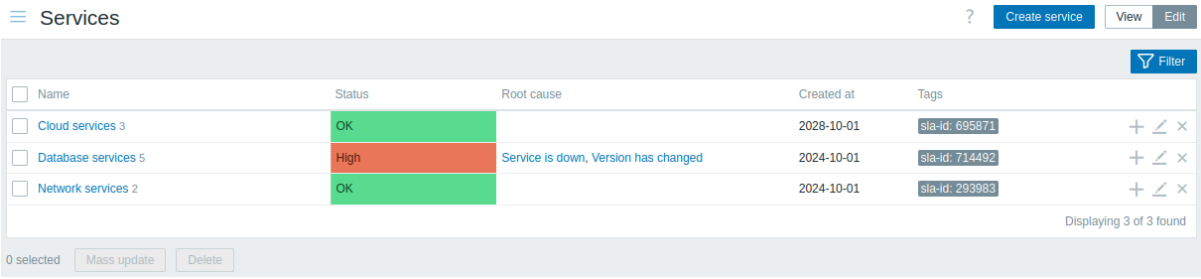


Параметр	Описание
Name	Фильтрация по имени сервиса.
Status	Фильтрация по состоянию сервиса.
Only services without children	Установите этот флажок, чтобы отображать только сервисы без дочерних сервисов. Этот параметр доступен только в режиме редактирования сервиса.
Only services without problem tags	Установите этот флажок, чтобы отображать только сервисы без тегов проблем. Этот параметр доступен только в режиме редактирования сервиса.
Tags	Фильтрация по имени и значению тега сервиса либо по имени и значению тега проблемы сервиса (в режиме редактирования сервиса). 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: **Exists** - включить указанные имена тегов; **Equals** - включить указанные имена и значения тегов (с учётом регистра); **Contains** - включить указанные имена тегов, если значения тегов содержат введённую строку (поиск подстроки, без учёта регистра); **Does not exist** - исключить указанные имена тегов; **Does not equal** - исключить указанные имена и значения тегов (с учётом регистра); **Does not contain** - исключить указанные имена тегов, если значения тегов содержат введённую строку (поиск подстроки, без учёта 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.

Редактирование сервисов

Нажмите кнопку *Edit*, чтобы перейти в режим редактирования. В режиме редактирования список дополняется флажками перед записями, а также следующими дополнительными параметрами:

-  - добавить дочерний сервис к этому сервису
-  - редактировать этот сервис
-  - удалить этот сервис

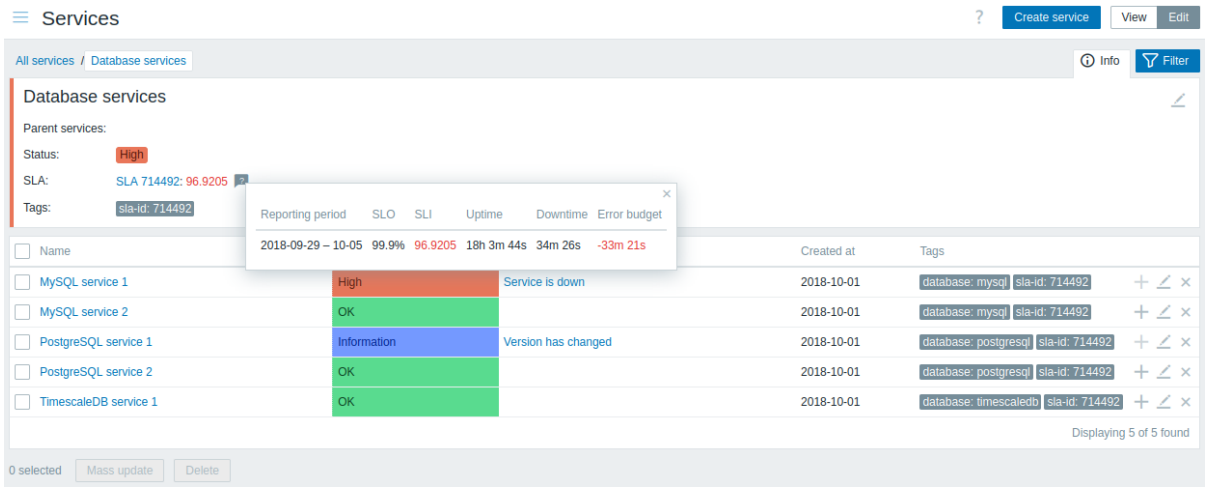


Чтобы настроить новый сервис, нажмите кнопку *Create service* в правом верхнем углу.

Сведения об услуге

Чтобы открыть сведения об услуге, нажмите на имя услуги. Чтобы вернуться к списку всех услуг, нажмите *Все услуги*.

Сведения об услуге включают информационный блок и список дочерних услуг.



Чтобы открыть информационный блок, нажмите вкладку *Инфо*. Информационный блок содержит следующие записи:

- Имена родительских услуг (если есть)
- Текущий статус этой услуги
- Текущие SLA этой услуги в формате `SLA name:service level indicator`. 'SLA name' также является ссылкой на отчёт SLA для этой услуги. Если навести указатель мыши на информационный блок рядом с индикатором уровня услуги (SLI), отобразится всплывающий список с подробностями SLI. Индикатор уровня услуги показывает текущий уровень услуги в процентах.
- Теги услуги

Информационный блок также содержит ссылку на настройку услуги.

Чтобы использовать фильтр для дочерних услуг, нажмите вкладку *Фильтр*.

В режиме редактирования список дочерних услуг дополняется следующими параметрами редактирования:

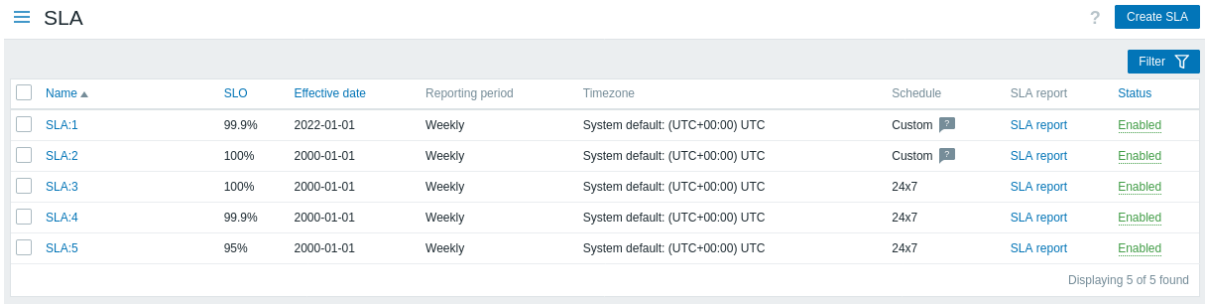
-  - добавить дочернюю услугу к этой услуге
-  - редактировать эту услугу
-  - удалить эту услугу

2 SLA

Обзор

Этот раздел веб-интерфейса позволяет просматривать и настраивать соглашения об уровне обслуживания (SLA).

SLA



Отображается список настроенных SLAs. *Обратите внимание*, что будут отображаться только SLA, относящиеся к услугам, доступным пользователю (в режиме «только для чтения», если для роли пользователя не включён параметр *Управление SLA (Manage SLA)*).

Отображаемые данные:

Параметр	Описание
Имя (Name)	Отображается имя SLA. Имя является ссылкой на настройки SLA.
SLO	Отображается целевой уровень обслуживания (service level objective, SLO).

Параметр	Описание
Действительно с (Effective date)	Отображается дата начала расчёта SLA .
Отчётный период (Reporting period)	Отображается период, используемый в отчёте SLA — *ежедневно (daily)*, *еженедельно (weekly)*, *ежемесячно (monthly)*, *ежеквартально (quarterly)* или *ежегодно (annually)*.
Часовой пояс (Time zone)	Отображается часовой пояс SLA.
Расписание (Schedule)	Отображается расписание SLA — 24x7 или пользовательское.
SLA отчёт (SLA report)	Щёлкните по ссылке, чтобы увидеть SLA отчёт для данного SLA.
Состояние (Status)	Отображается состояние SLA — активировано или деактивировано.

3 Отчет SLA

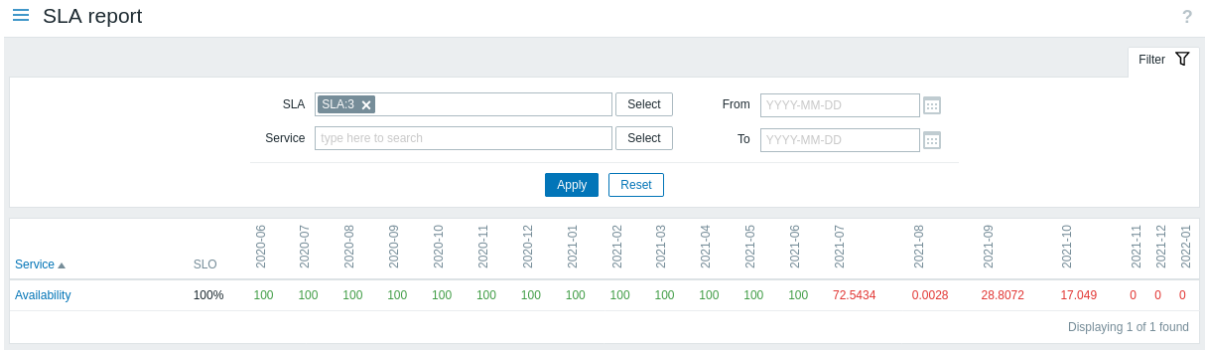
Обзор

Этот раздел позволяет просматривать отчеты SLA, основанные на критериях, выбранных в фильтре.

Отчеты SLA также могут отображаться в виде виджета панели мониторинга.

Отчет

Фильтр позволяет выбрать отчет по имени SLA, а также по имени сервиса. Также можно ограничить отображаемый период.

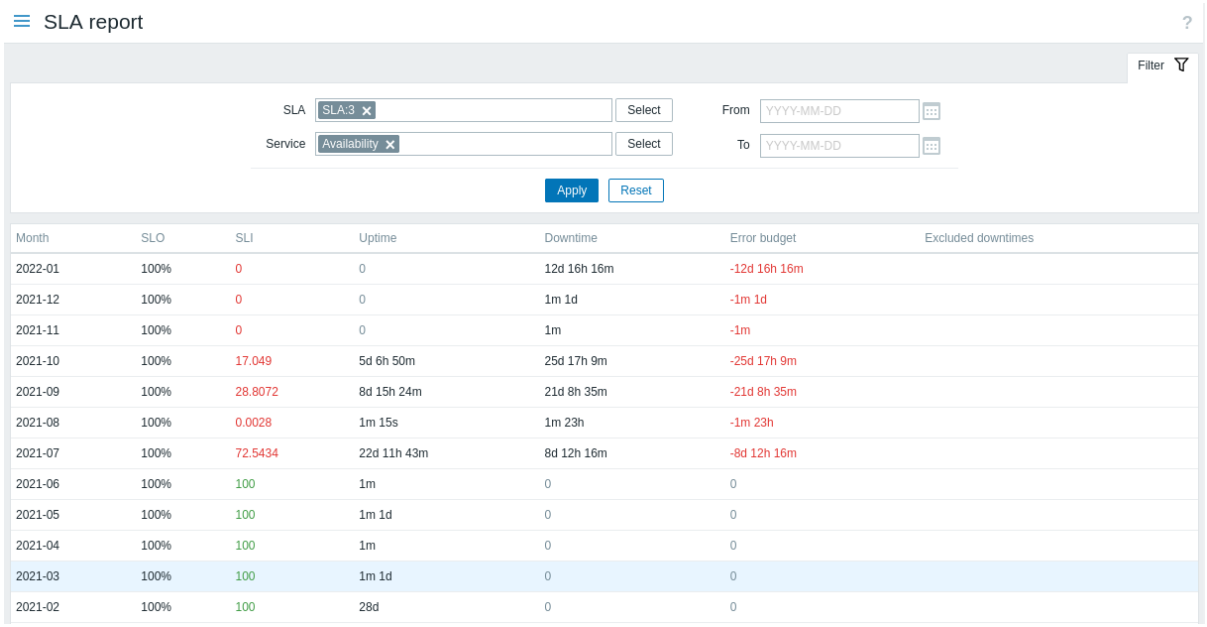


В каждом столбце (периоде) отображается SLI за этот период. SLI, которые нарушают заданный SLO, выделяются красным цветом.

В отчете отображается 20 периодов. Можно отобразить максимум 100 периодов, если указаны обе даты: *От* и *До*.

Сведения об отчёте

Если щёлкнуть по имени сервиса в отчёте, можно открыть другой отчёт, в котором отображается более подробное представление.



Month	SLO	SLI	Uptime	Downtime	Error budget	Excluded downtimes
2022-01	100%	0	0	12d 16h 16m	-12d 16h 16m	
2021-12	100%	0	0	1m 1d	-1m 1d	
2021-11	100%	0	0	1m	-1m	
2021-10	100%	17.049	5d 6h 50m	25d 17h 9m	-25d 17h 9m	
2021-09	100%	28.8072	8d 15h 24m	21d 8h 35m	-21d 8h 35m	
2021-08	100%	0.0028	1m 15s	1m 23h	-1m 23h	
2021-07	100%	72.5434	22d 11h 43m	8d 12h 16m	-8d 12h 16m	
2021-06	100%	100	1m	0	0	
2021-05	100%	100	1m 1d	0	0	
2021-04	100%	100	1m	0	0	
2021-03	100%	100	1m 1d	0	0	
2021-02	100%	100	28d	0	0	

Обратите внимание, что отрицательная длительность проблемы не влияет на расчёт SLA или формирование отчётов.

4 Инвентаризация

Обзор

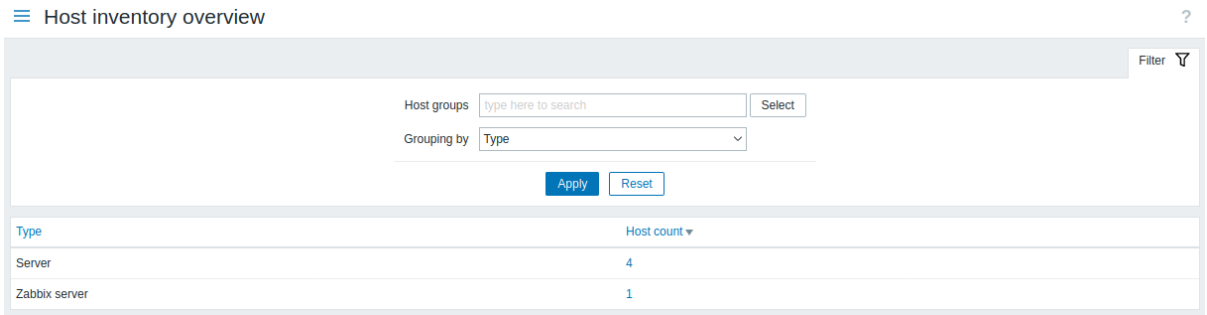
Меню «Инвентаризация» содержит разделы, которые предоставляют обзор данных инвентаризации узлов сети по выбранному параметру, а также возможность просматривать подробные сведения об инвентаризации узлов сети.

1 Обзор

Обзор

Раздел *Инвентаризация → Обзор* предоставляет возможность получить общее представление о данных инвентаризации узлов сети.

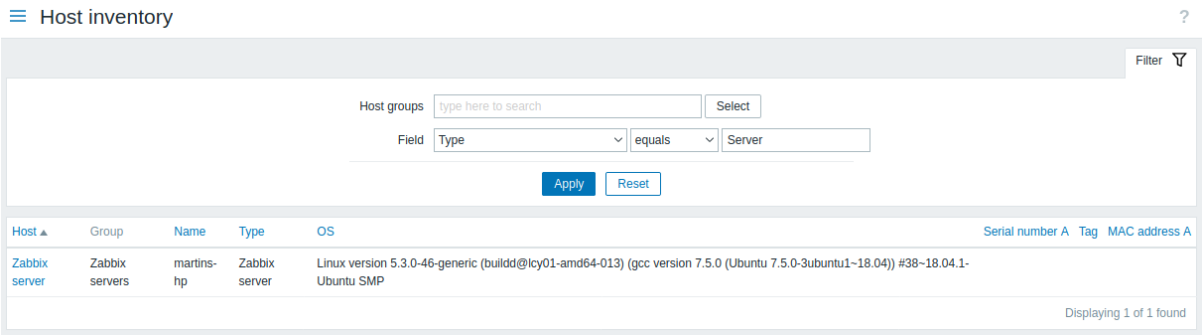
Чтобы отобразить обзор, выберите группы узлов сети (или не выбирайте ни одной), а также поле инвентаризации, по которому будут отображаться данные. Будет показано количество узлов сети, соответствующих каждой записи выбранного поля.



Type	Host count ▾
Server	4
Zabbix server	1

Полнота обзора зависит от того, какой объём инвентаризационной информации поддерживается для узлов сети.

Числа в столбце *Количество узлов сети* являются ссылками; они позволяют отфильтровать эти узлы сети в таблице *Инвентаризация узлов сети*.

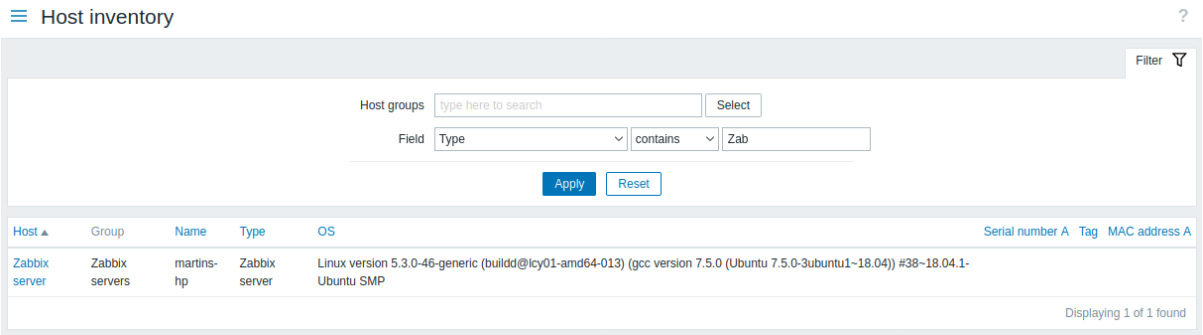


2 Узлы сети

Обзор

В разделе *Инвентаризация → Узлы сети* отображаются данные инвентаризации узлов сети.

Вы можете фильтровать узлы сети по группам узлов сети и по любому полю инвентаризации, чтобы отображать только те узлы сети, которые вас интересуют.

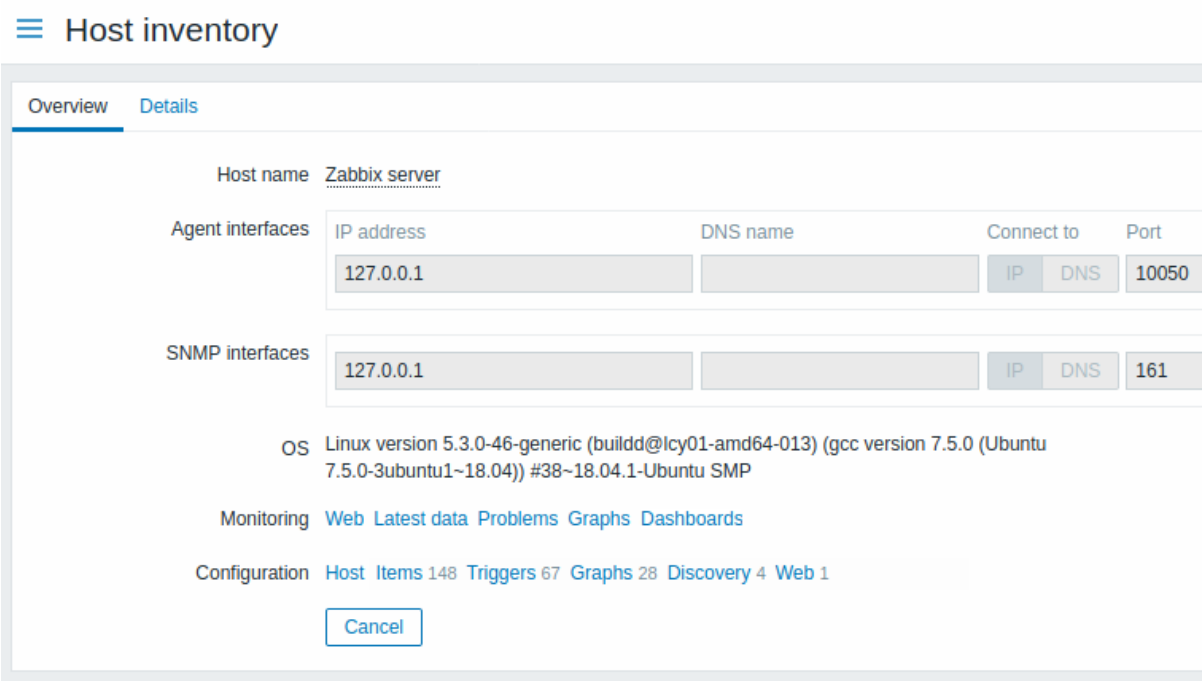


Чтобы отобразить инвентаризацию всех узлов сети, нажмите кнопку ”Сброс”.

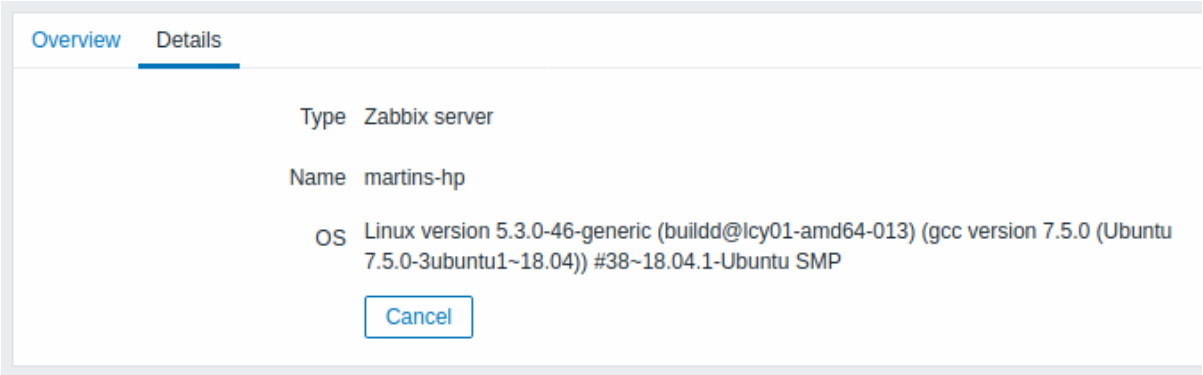
Хотя в таблице отображаются только некоторые ключевые поля инвентаризации, вы также можете просмотреть всю доступную информацию инвентаризации для этого узла сети. Для этого нажмите на имя узла сети в первом столбце.

Сведения об инвентаризации

Вкладка **Обзор** содержит общую информацию об узле сети, последние данные мониторинга и параметры конфигурации узла сети:



Вкладка **Сведения** содержит все доступные сведения инвентаризации для узла сети:



Полнота данных инвентаризации зависит от того, какой объем информации инвентаризации поддерживается для узла сети. Если никакая информация не поддерживается, вкладка *Сведения* отключена.

5 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.

1 Информация о системе

Обзор

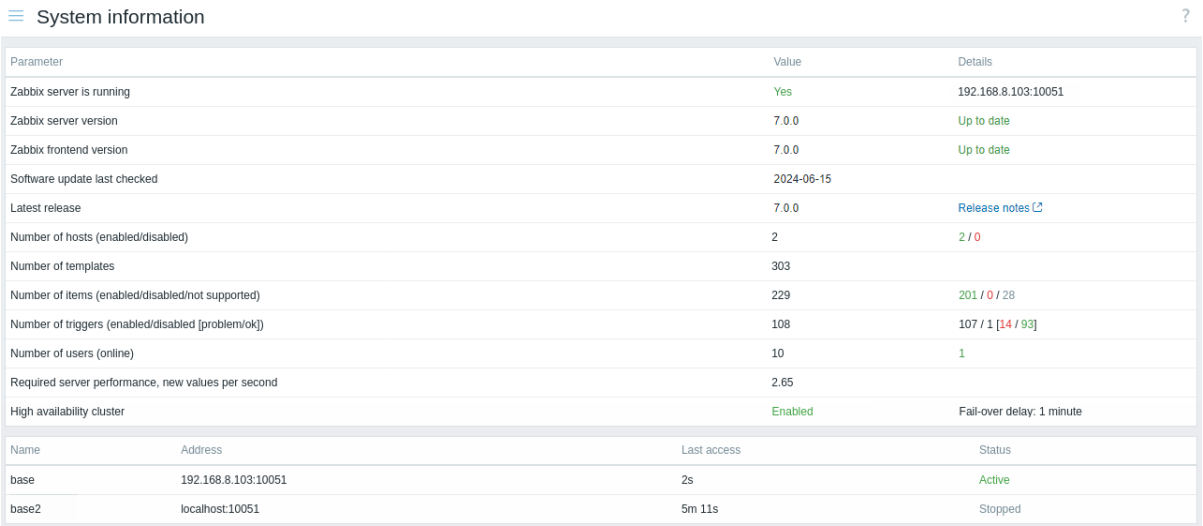
В разделе *Отчеты* → *Информация о системе* отображается сводка по ключевым данным сервера Zabbix и системы. Системные данные собираются с использованием внутренних элементов данных.

Обратите внимание, что в конфигурации высокой доступности можно перенаправить источник системной информации (экземпляр сервера). Для этого отредактируйте файл *zabbix.conf.php* — раскомментируйте и задайте `$ZBX_SERVER` или одновременно `$ZBX_SERVER` и `$ZBX_SERVER_PORT`, указав сервер, отличный от того, который показан как активный. Обратите внимание, что если задан только `$ZBX_SERVER`, для `$ZBX_SERVER_PORT` будет использовано значение по умолчанию (10051).

Если включена конфигурация высокой доступности, под системной статистикой отображается отдельный блок с подробной информацией об узлах высокой доступности. Этот блок виден только пользователям Zabbix *Super Admin*.

Информация о системе также доступна в виде виджета на панели мониторинга.

Системная статистика



Отображаемые данные:

Параметр	Значение	Сведения
Zabbix server is running	Статус сервера Zabbix: **Да** — сервер запущен **Нет** — сервер не запущен *Примечание:* Чтобы отобразить остальную информацию, веб-интерфейсу требуется, чтобы сервер был запущен, и на сервере должен быть запущен как минимум один процесс trapper (параметр StartTrappers в файле zabbix_server.conf > 0).	Расположение и порт сервера Zabbix.
Zabbix server version	Отображается номер текущей версии сервера. *Примечание:* Отображается только при запущенном сервере Zabbix.	Отображается статус версии сервера: **Актуальная** — используется последняя версия; **Доступно новое обновление** — доступна более новая версия; **Устаревшая** — период полной поддержки этой версии истёк. Эта информация доступна только в том случае, если проверка обновлений ПО включена в конфигурации сервера Zabbix. Ничего не отображается, если последняя проверка обновлений ПО выполнялась более недели назад или отсутствуют данные о текущей версии.
Zabbix frontend version	Отображается номер версии веб-интерфейса Zabbix.	Отображается статус версии веб-интерфейса Zabbix: **Актуальная** — используется последняя версия; **Доступно новое обновление** — доступна более новая версия; **Устаревшая** — период полной поддержки этой версии истёк. Эта информация доступна только в том случае, если проверка обновлений ПО включена в конфигурации сервера Zabbix. Ничего не отображается, если последняя проверка обновлений ПО выполнялась более недели назад или отсутствуют данные о текущей версии.
Software update last checked	Отображается дата последней проверки обновлений ПО Zabbix. Эта информация доступна только в том случае, если проверка обновлений ПО включена в конфигурации сервера Zabbix.	
Latest release	Отображается номер более нового релиза (если доступен) для текущей версии Zabbix. Эта информация доступна только в том случае, если проверка обновлений ПО включена в конфигурации сервера Zabbix. Ничего не отображается, если последняя проверка обновлений ПО выполнялась более недели назад или отсутствуют данные о текущей версии.	Отображается ссылка на примечания к выпуску последнего доступного релиза Zabbix.
Number of hosts	Отображается общее количество настроенных узлов сети.	Количество наблюдаемых/ненаблюдаемых узлов сети.
Number of templates	Отображается общее количество шаблонов.	
Number of items	Отображается общее количество элементов данных.	Количество наблюдаемых/отключённых/неподдерживаемых элементов данных на уровне узла сети. Элементы данных на отключённых узлах сети учитываются как отключённые.

Параметр	Значение	Сведения
Number of triggers	Отображается общее количество триггеров.	Количество включённых/отключённых триггеров на уровне узла сети; распределение включённых триггеров по состояниям «Проблема»/«OK». Триггеры, перечисленные в состоянии «OK», включают триггеры со статусом «Неизвестно». Триггеры, зависящие от отключённых элементов данных или назначенные отключённым узлам сети, учитываются как отключённые.
Number of users	Отображается общее количество настроенных пользователей.	Количество пользователей в сети.
Required server performance, new values per second	Отображается ожидаемое количество новых значений, обрабатываемых сервером Zabbix в секунду.	*Required server performance* является оценочным значением и может быть полезен как ориентир. Для получения точных значений количества обработанных данных используйте `zabbix[wcache,values,all]` внутренний элемент данных. В расчёт включаются включённые элементы данных с наблюдаемых узлов сети. Элементы данных журнала учитываются как одно значение на интервал обновления элемента данных. Учитываются значения с обычным интервалом; значения с гибким интервалом и по расписанию не учитываются. Расчёт не корректируется во время периода обслуживания «nodata». Элементы данных trapper не учитываются.
Global scripts on Zabbix server	В этом поле будет отображаться **Disabled**, если глобальные скрипты отключены на сервере Zabbix установкой EnableGlobalScripts=0 в конфигурации сервера.	
High availability cluster	Статус кластера высокой доступности для сервера Zabbix: **Disabled** — автономный сервер **Enabled** — существует как минимум один узел высокой доступности	Если включено, отображается задержка переключения на резервный узел.

System information также отображает сообщение об ошибке в следующих случаях:

- Используемая база данных не имеет требуемой кодировки или сопоставления (UTF-8).
- Версия базы данных ниже или выше поддерживаемого диапазона (доступно только пользователям с типом роли *Super admin role*).
- Housekeeping для TimescaleDB настроен некорректно (таблицы history или trend содержат сжатые чанки, но параметры *Override item history period* или *Override item trend period* отключены).

Узлы высокой доступности

Если кластер высокой доступности включен, то отображается дополнительный блок данных со статусом каждого узла высокой доступности.



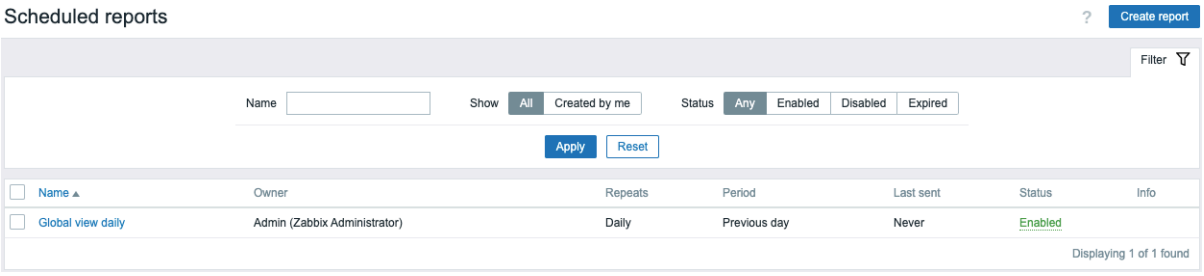
Отображаемые данные:

Column	Description
Name	Имя узла, как определено в конфигурации сервера.
Address	IP-адрес и порт узла.
Last access	Время последнего обращения к узлу. При наведении курсора на ячейку отображается метка времени последнего обращения в полном формате.
Status	Статус узла. Строки таблицы сортируются по этим статусам в следующем порядке приоритета: **Active** - узел работает и доступен **Unavailable** - узел не появлялся 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

2 Отчеты по расписанию

Обзор

В *Отчёты → Запланированные отчёты* пользователи с достаточными правами могут настроить запланированное создание PDF-версий панелей, которые будут отправляться по электронной почте указанным получателям.



На открывающемся экране отображается информация о запланированных отчётах, которую для удобства навигации можно отфильтровать — см. раздел Использование фильтра ниже.

Отображаемые данные:

Column	Description
Name	Название отчёта. Щелчок по нему открывает форму настройки отчёта.
Owner	Пользователь, создавший отчёт.
Repeats	Частота формирования отчёта (ежедневно/еженедельно/ежемесячно/ежегодно).
Period	Период времени, за который подготавливается отчёт (предыдущий день, неделя, месяц или год).
Last sent	Дата и время отправки последнего отчёта.
Status	Текущее состояние отчёта (включён/отключён/истёк). Пользователи с достаточными правами могут изменить состояние, щёлкнув по нему: с ”Enabled” на ”Disabled” (и обратно); с ”Expired” на ”Disabled” (и обратно). Для пользователей с недостаточными правами состояние недоступно для нажатия.

Column	Description
Info	Отображает информационные значки: Красный значок указывает, что формирование отчёта завершилось ошибкой; при наведении отображается всплывающая подсказка с информацией об ошибке. Жёлтый значок указывает, что отчёт был сформирован, но отправка некоторым (или всем) получателям не удалась, либо срок действия отчёта истёк; при наведении 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
- *Показывать* — созданные текущим пользователем или все отчеты
- *Состояние* — выберите между ”Любое” (показать все отчеты), ”Включено”, ”Отключено” или ”Истекло”

Фильтр расположен под названием раздела *Запланированные отчеты*. Его можно открыть и свернуть, щелкнув вкладку *Фильтр* в правом верхнем углу.

Массовое обновление

Иногда может потребоваться удалить или изменить статус сразу нескольких отчетов. Вместо того чтобы открывать каждый отдельный отчет для редактирования, вы можете использовать для этого функцию массового обновления.

Чтобы массово обновить несколько отчетов, выполните следующие действия:

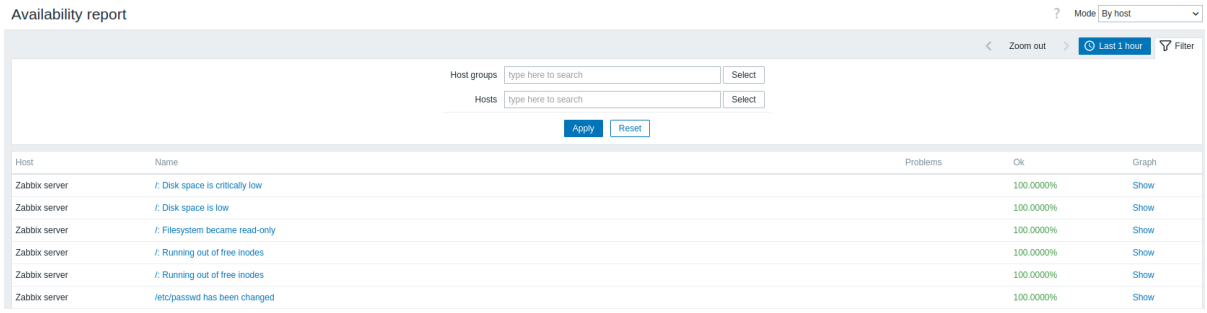
- Установите флажки у отчетов, которые нужно обновить, в списке
- Нажмите нужную кнопку под списком, чтобы внести изменения (*Включить*, *Отключить* или *Удалить*)

3 Отчет о доступности

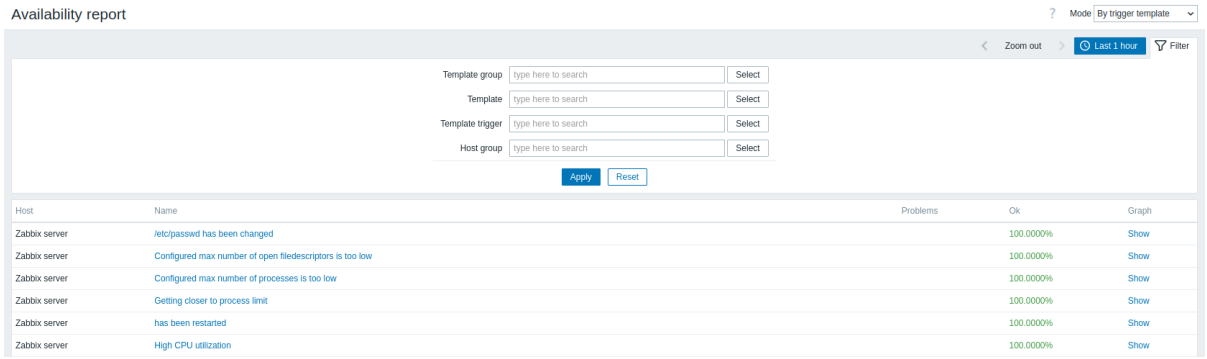
Обзор

В *Отчёты* > *Отчёт о доступности* вы можете 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 выпадающем списке в правом верхнем углу можно выбрать режим отбора — отображать триггеры по узлам сети или по триггерам, принадлежащим шаблону.



Имя триггера является ссылкой на последние события этого триггера.

Использование фильтра

Фильтр помогает сузить количество отображаемых узлов сети и/или триггеров. Для повышения производительности поиска данные ищутся с неразрешёнными макросами.

Фильтр расположен под названием раздела *Отчёт о доступности*. Его можно открыть и свернуть, щёлкнув по вкладке *Фильтр* справа.

Фильтрация по шаблону триггера

В режиме *По шаблону триггера* результаты можно фильтровать по одному или нескольким из перечисленных ниже параметров.

Параметр	Описание
Группа шаблонов	Фильтровать узлы сети по триггерам, унаследованным от шаблонов, принадлежащих выбранной группе шаблонов. Указание родительской группы шаблонов неявно выбирает все вложенные группы шаблонов.
Шаблон	Фильтровать узлы сети по триггерам, унаследованным от выбранного шаблона, включая вложенные шаблоны. Если вложенный шаблон имеет собственные триггеры, эти триггеры отображаться не будут.
Триггер шаблона	Фильтровать узлы сети по выбранному триггеру. Другие триггеры отфильтрованных узлов сети отображаться не будут.
Группа узлов сети	Фильтровать узлы сети, принадлежащие 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 неявно выбираются все вложенные группы узлов сети.

Селектор периода времени

Селектор периода времени позволяет выбирать часто используемые периоды времени одним щелчком. Селектор можно развернуть и свернуть, щелкнув по вкладке *Период времени* рядом с фильтром.

Щелчок по *Показать* в столбце *График* отображает информацию о доступности в виде столбчатого графика, где каждый столбец представляет прошедшую неделю текущего года.

Availability report graph



Зеленый цвет столбца обозначает время OK, а красный — время проблемы.

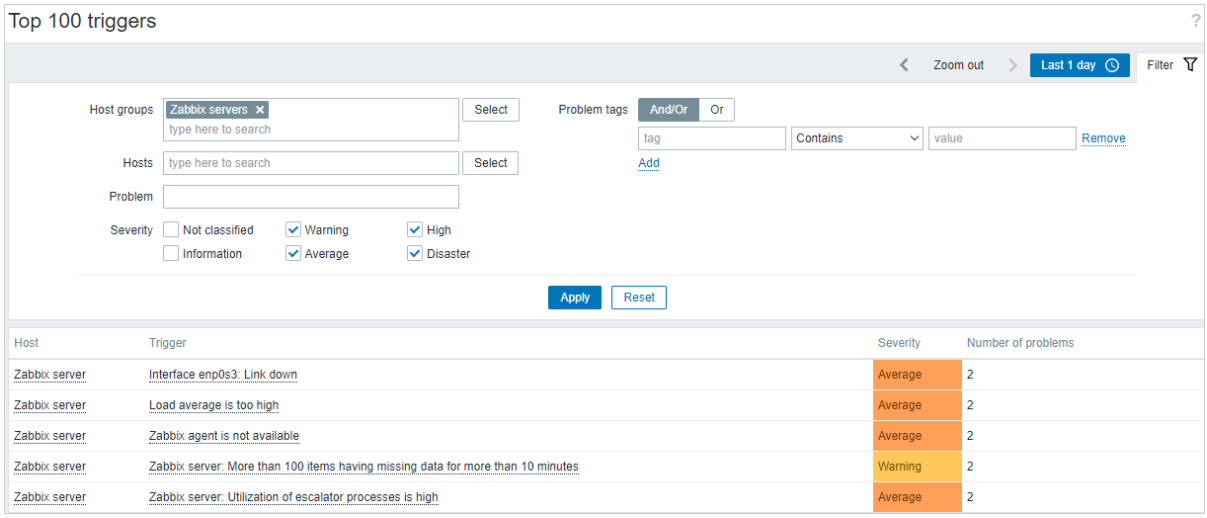
Влияние периодов обслуживания

Обслуживание не исключает время из отчёта о доступности автоматически. Только обслуживание, настроенное как *Без сбора данных*, останавливает опрос (поэтому проблемы не создаются) и, следовательно, оставляет процент доступности без изменений для затронутых триггеров.

4 Топ 100 триггеров

Обзор

В разделе *Отчёты → Топ 100 триггеров* вы можете увидеть триггеры с наибольшим количеством проблем, обнаруженных за выбранный период.



Записи в столбцах узла сети и триггера являются ссылками, которые предоставляют несколько полезных возможностей:

- для узла сети — щелчок по имени узла сети открывает меню узла сети
- для триггера — щелчок по имени триггера открывает ссылки на последние события, простой график для каждого элемента данных триггера, а также формы настройки самого триггера и каждого элемента данных триггера

Использование фильтра

Вы можете использовать фильтр для отображения триггеров по группе узлов сети, узлу сети, имени проблемы, тегам или важности триггера. При указании родительской группы узлов сети неявно выбираются все вложенные группы узлов сети. Для повышения производительности поиска данные ищутся с неразрешёнными макросами.

Фильтр расположен под названием раздела *Топ 100 триггеров*. Его можно развернуть и свернуть, щёлкнув по вкладке *Фильтр* справа.

Селектор периода времени

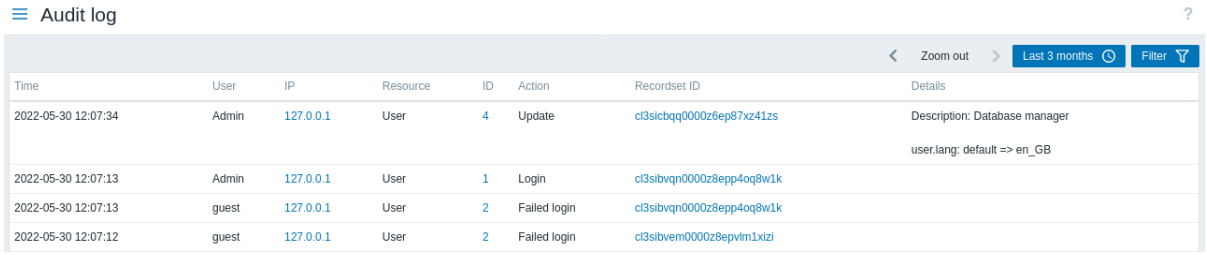
Селектор периода времени рядом с фильтром позволяет выбирать часто используемые периоды одним щелчком мыши. Подробнее см. в разделе Селекторы периода времени и узла сети.

5 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 установлен флажок *Включить журналирование аудита*. Если этот параметр не включен, история действий не будет записываться в базу данных и не будет отображаться в журнале аудита.



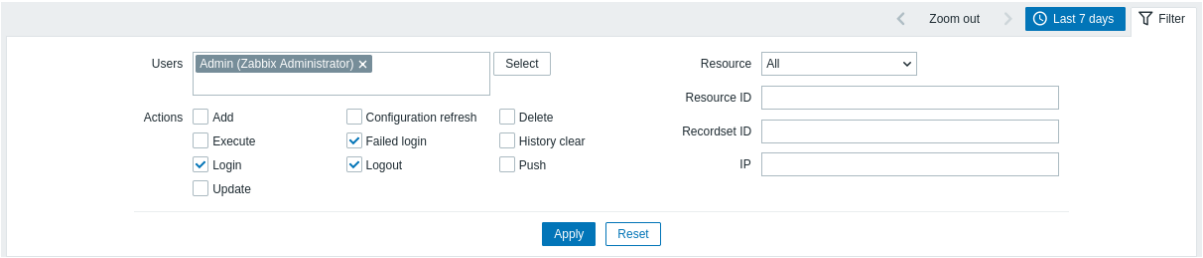
Журнал аудита отображает следующие данные:

Column	Description
Time	Метка времени записи аудита.
User	Пользователь, выполнивший действие.
IP	IP-адрес, с которого было инициировано действие. Щелчок по гиперссылке приведет к фильтрации записей журнала аудита по этому IP-адресу.
Resource	Тип затронутого ресурса (*API token*, *Action*, *Authentication*, *Autoregistration* и т. д.).
ID	ID затронутого ресурса. Щелчок по гиперссылке приведет к фильтрации записей журнала аудита 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 привязке шаблона к узлу сети для каждой унаследованной сущности шаблона (элемент данных, триггер и т. д.) создается отдельная запись журнала аудита — все эти записи будут иметь одинаковый *Recordset ID*. Щелчок по гиперссылке приведет к фильтрации записей журнала аудита по этому *Recordset ID*.
Details	Описание ресурса и подробная информация о выполненном действии. Если запись содержит более двух строк, будет отображена дополнительная ссылка *Details*. Щелкните по этой ссылке, чтобы просмотреть полный список изменений.

Note:
Когда элемент данных trapper или элемент данных HTTP agent (с включенным trapping) получает данные, запись в журнал аудита будет добавлена только в том случае, если данные были отправлены с использованием метода API `history.push`, а не утилиты Zabbix sender.

Использование фильтра

Фильтр расположен под панелью *Журнал аудита*. Его можно развернуть и свернуть, щелкнув вкладку *Фильтр* в правом верхнем углу.



Вы можете использовать фильтр, чтобы 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 одно или несколько конкретных действий.

Для повышения производительности поиска все данные ищутся с неразрешенными макросами.

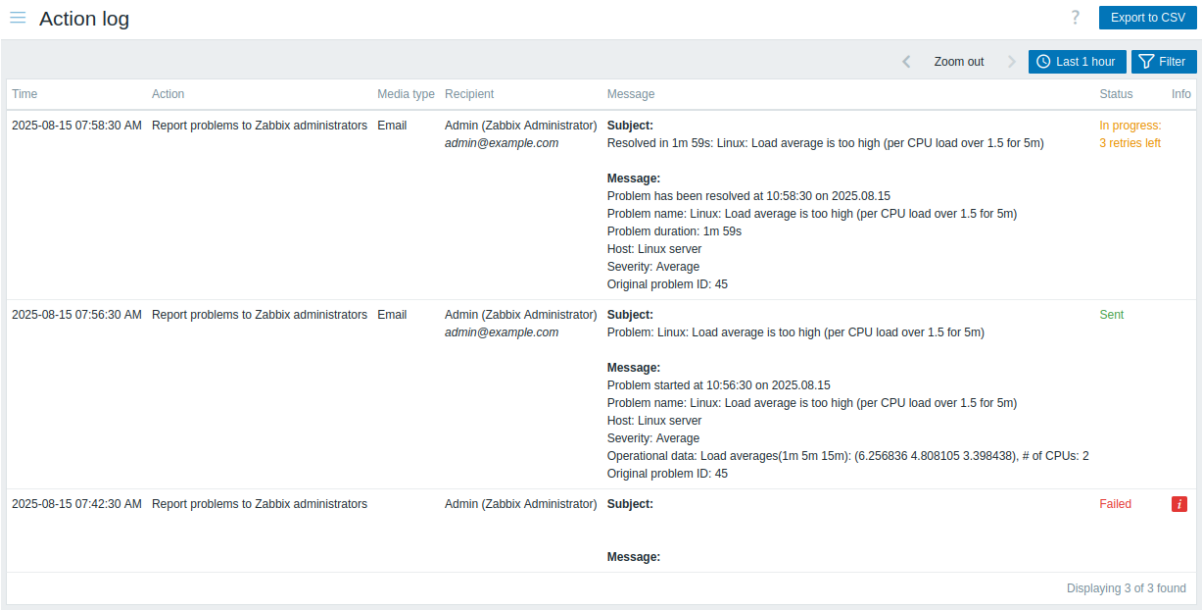
Селектор периода времени

Селектор периода времени рядом с фильтром позволяет выбрать часто используемые периоды одним щелчком мыши. Для получения дополнительной информации см. Селекторы периода времени и узла сети.

6 Журнал действий

Обзор

В разделе *Отчёты* → *Журнал действий* вы можете просматривать подробную информацию об операциях (уведомлениях, удалённых командах), выполненных в рамках действия.



Отображаемые данные:

Column	Description
Time	Метка времени операции.
Action	Имя действия, вызвавшего выполнение операций.
Media type	Тип средства оповещения (например, Email, Jira и т. д.), используемый для отправки уведомления. Для операций, в которых выполнялись удалённые 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 средства оповещения (email, имя пользователя и т. д.). Для операций, в которых выполнялись удалённые команды, этот столбец будет пустым.
Message	Содержимое сообщения, удалённой команды или имя глобального скрипта. Удалённая команда отделяется от целевого узла сети двоеточием: `<host>:<command>`. Например, если удалённая команда была выполнена на сервере Zabbix, информация будет иметь следующий формат: `Zabbix server:<command>`.
Status	Статус операции: *In progress* — операция отправки уведомления выполняется (также отображается оставшееся количество попыток, которое сервер предпримет для отправки уведомления) *Sent* — уведомление отправлено *Executed* — удалённая команда выполнена *Failed* — операция не была завершена
Info	Информация об ошибке (если имеется), связанной с выполнением операции.

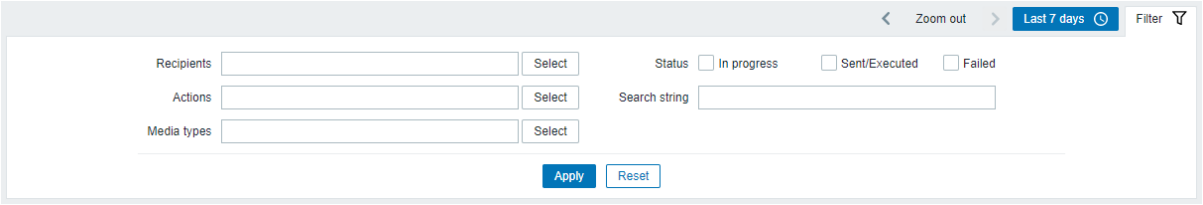
Кнопки

Кнопка в правом верхнем углу страницы предоставляет следующую возможность:

Export to CSV	Экспортировать записи журнала действий с текущей страницы в CSV-файл. Если применён фильтр, будут экспортированы только отфильтрованные записи. В экспортированном CSV-файле столбцы «Получатель» и «Сообщение» разделены на несколько столбцов: «Имя пользователя Zabbix получателя», «Имя получателя», «Фамилия получателя», «Получатель», а также «Тема», «Сообщение», «Команда».

Использование фильтра

Фильтр расположен под панелью *Журнал действий*. Его можно развернуть и свернуть, щелкнув по вкладке *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.

Селектор периода времени

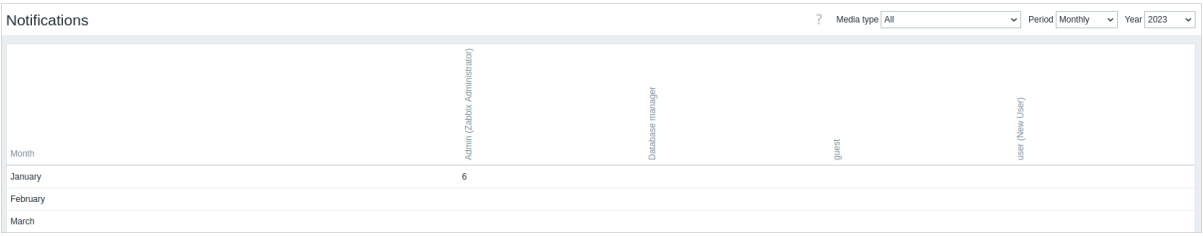
Селектор периода времени рядом с фильтром позволяет выбрать часто используемые периоды одним щелчком мыши. Для получения дополнительной информации см. Селектор периода времени и узла сети.

7 Уведомления

Обзор

В разделе *Отчёты* → *Уведомления* отображается отчёт о количестве уведомлений, отправленных каждому пользователю.

В раскрывающихся списках в правом верхнем углу можно выбрать тип средства оповещения (или все), период (данные за каждый день/неделю/месяц/год) и год для отправленных уведомлений.



В каждом столбце отображаются итоговые значения для одного системного пользователя.

6 Администрирование

Обзор

Меню Администрирование используется в Zabbix для административных функций. Это меню доступно только пользователям с типом Super Administrators.

1 Общее

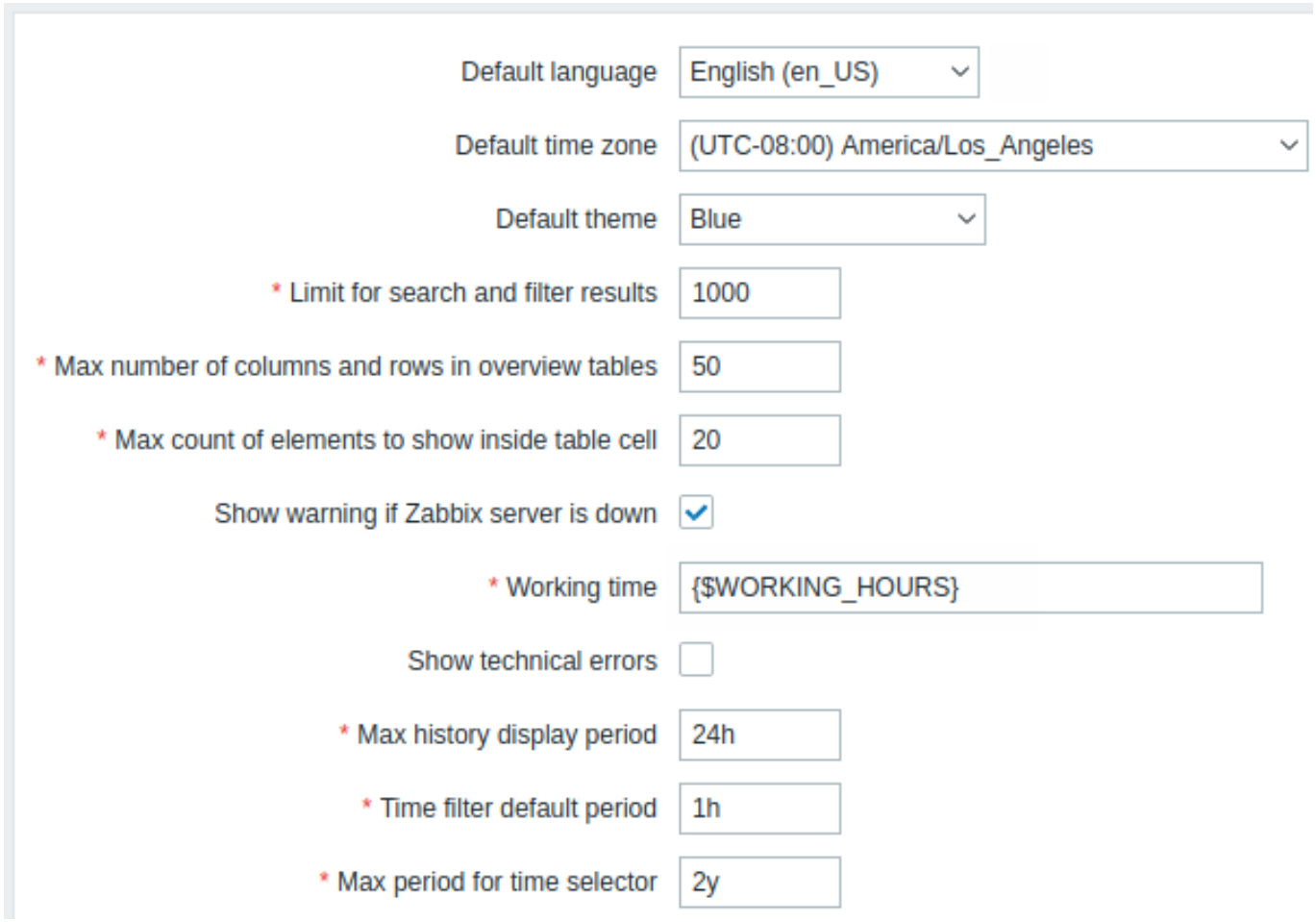
Обзор

Раздел *Администрирование* → *Общие* содержит ряд подразделов для настройки значений по умолчанию, связанных с веб-интерфейсом, и адаптации Zabbix.

Список доступных подразделов отображается при нажатии на *Общие* в разделе меню *Администрирование*. Также можно переключаться между подразделами с помощью выпадающего списка заголовка в левом верхнем углу.

GUI

В этом разделе можно настроить несколько значений по умолчанию, связанных с веб-интерфейсом.



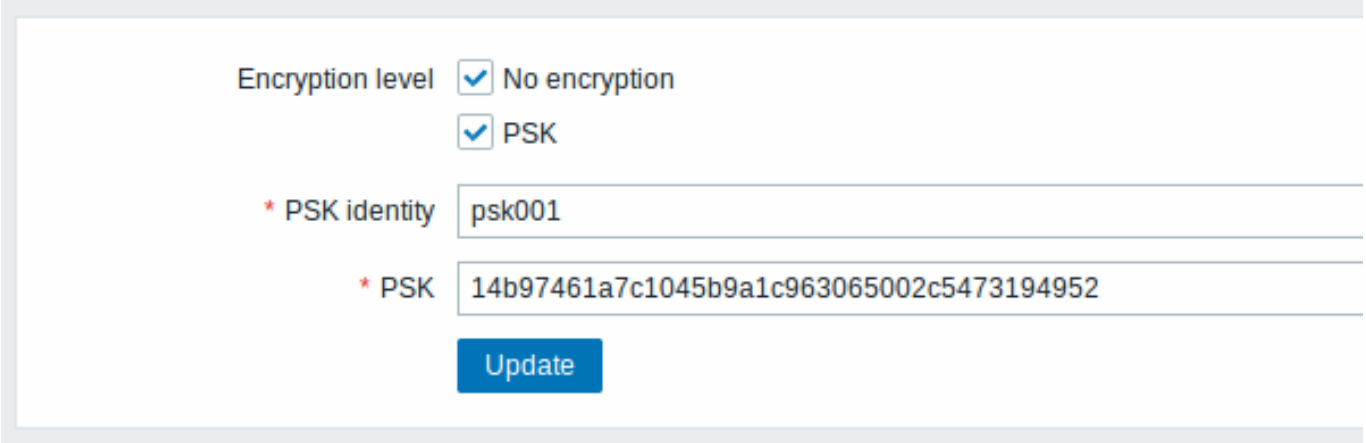
Параметры конфигурации:

Параметр	Описание
Язык по умолчанию	Язык по умолчанию для пользователей, которые не указали язык в своих профилях, а также для гостевых пользователей. Подробнее см. в разделе Установка дополнительных языков веб-интерфейса.
Часовой пояс по умолчанию	Часовой пояс по умолчанию для пользователей, которые не указали часовой пояс в своих профилях, а также для гостевых пользователей. Подробнее см. в разделе Часовой пояс.
Тема по умолчанию	Тема по умолчанию для пользователей, которые не указали тему в своих профилях, а также для гостевых пользователей.
Ограничение для результатов поиска и фильтрации	Максимальное количество элементов (строк), отображаемых в списке веб-интерфейса, например, в *Сбор данных* > *Узлы сети*. *Примечание*: если установить, например, значение '50', во всех затронутых списках веб-интерфейса будут отображаться только первые 50 элементов. Если какой-либо список содержит более пятидесяти элементов, это будет обозначено знаком '+' в строке *"Displaying 1 to 50 of **50+** found"*. Кроме того, если используется фильтрация и совпадений всё равно больше 50, будут показаны только первые 50. Обратите внимание, что увеличение значения этого параметра может привести к снижению производительности и увеличению потребления памяти на стороне веб-сервера.
Максимальное количество столбцов и строк в обзорных таблицах	Максимальное количество столбцов и строк, отображаемых в виджете панели мониторинга *Обзор триггеров*. Одно и то же ограничение применяется как к столбцам, так и к строкам. Если существует больше строк и/или столбцов, чем отображается, система выведет предупреждение внизу таблицы: "Not all results are displayed. Please provide more specific search criteria."
Максимальное количество элементов для отображения внутри ячейки таблицы	Для записей, отображаемых в одной ячейке таблицы, будет показано не больше количества, заданного здесь.

Параметр	Описание
Показывать предупреждение, если сервер Zabbix недоступен	Этот параметр включает отображение предупреждающего сообщения в окне браузера, если сервер Zabbix недоступен (возможно, не работает). Сообщение остаётся видимым, даже если пользователь прокручивает страницу вниз. При наведении курсора сообщение временно скрывается, чтобы показать содержимое под ним.
Рабочее время	Этот общесистемный параметр определяет рабочие часы. На графиках рабочее время отображается белым фоном, а нерабочее — серым. Описание формата времени см. на странице Спецификация периода времени. Поддерживаются пользовательские макросы.
Показывать технические ошибки	Если флажок установлен, все зарегистрированные пользователи смогут видеть технические ошибки (PHP/SQL). Если флажок снят, эта информация доступна только супер-администраторам Zabbix и пользователям, входящим в группы пользователей с включённым режимом отладки.
Максимальный период отображения истории	Максимальный период времени, за который отображаются исторические данные в *Мониторинг* > *Последние данные*, информация о веб-сценариях узла сети в *Мониторинг* > *Узлы сети*, а также в виджете панели мониторинга *Top items*. Допустимый диапазон: от 24 часов (по умолчанию) до 1 недели. Поддерживаются суффиксы времени, например 1w (одна неделя), 36h (36 часов).
Период по умолчанию для фильтра времени	Период времени, используемый по умолчанию в графиках и панелях мониторинга. Допустимый диапазон: от 1 минуты до 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 пользовательская настройка, заменяя глобальное значение по умолчанию или предыдущий выбор пользователя.
Максимальный период для селектора времени	Максимально доступный период времени для графиков и панелей мониторинга. Пользователи не смогут просматривать более старые данные. Допустимый диапазон: от 1 года до 10 лет (по умолчанию: 2 года). Поддерживаются суффиксы времени, например 1y (один год), 365w (365 недель).

Авторегистрация

В этом разделе вы можете настроить уровень шифрования для авторегистрации активного агента.



Параметры, отмеченные звездочкой, являются обязательными.

Параметры настройки:

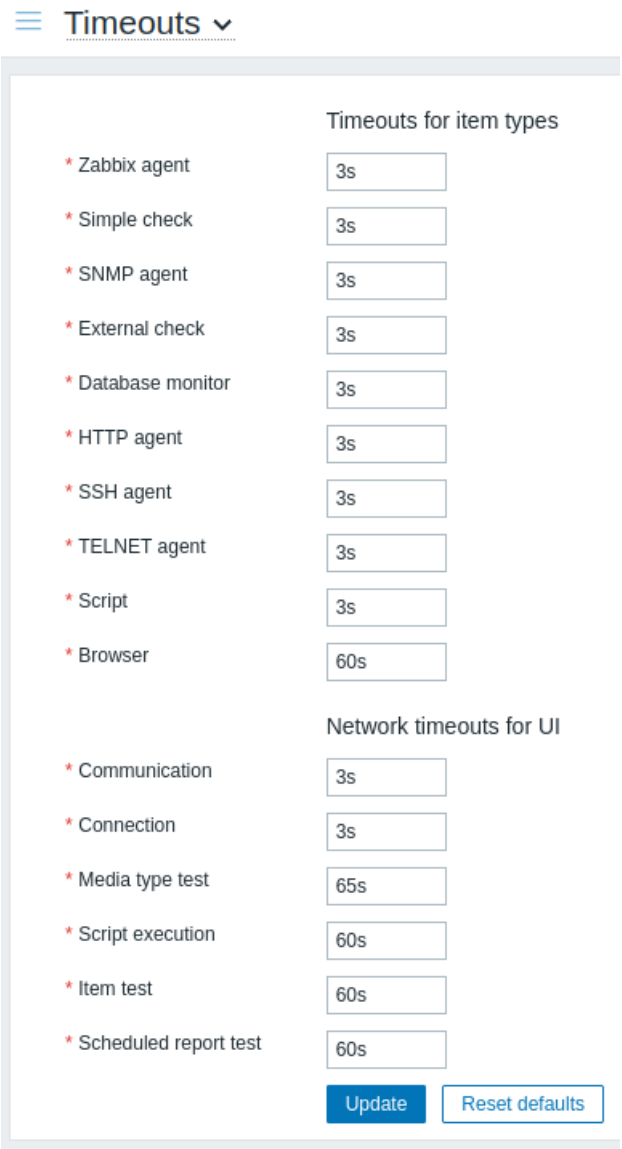
Параметр	Описание
Уровень шифрования	Выберите один или оба варианта уровня шифрования: **Без шифрования** — разрешены незашифрованные соединения **PSK** — разрешены соединения с TLS-шифрованием с использованием предварительно общего ключа
PSK identity	Введите строку идентификатора предварительно общего ключа. Это поле доступно только если в качестве *Уровень шифрования* выбран 'PSK'. Не указывайте конфиденциальную информацию в идентификаторе PSK, так как он передается по сети в незашифрованном виде, чтобы сообщить получателю, какой PSK следует использовать.

Параметр	Описание
PSK	Введите предварительно общий ключ (четное количество шестнадцатеричных символов). Максимальная длина: 512 шестнадцатеричных цифр (256-байтный PSK), если Zabbix использует библиотеку GnuTLS или OpenSSL, 64 шестнадцатеричные цифры (32-байтный PSK), если Zabbix использует библиотеку mbed TLS (PolarSSL). Пример: 1f87b595725ac58dd977beef14b97461a7c1045b9a1c963065002c5473194952 Это поле доступно только если в качестве *Уровень шифрования* выбран 'PSK'.

См. также: Безопасная авторегистрация

Тайм-ауты

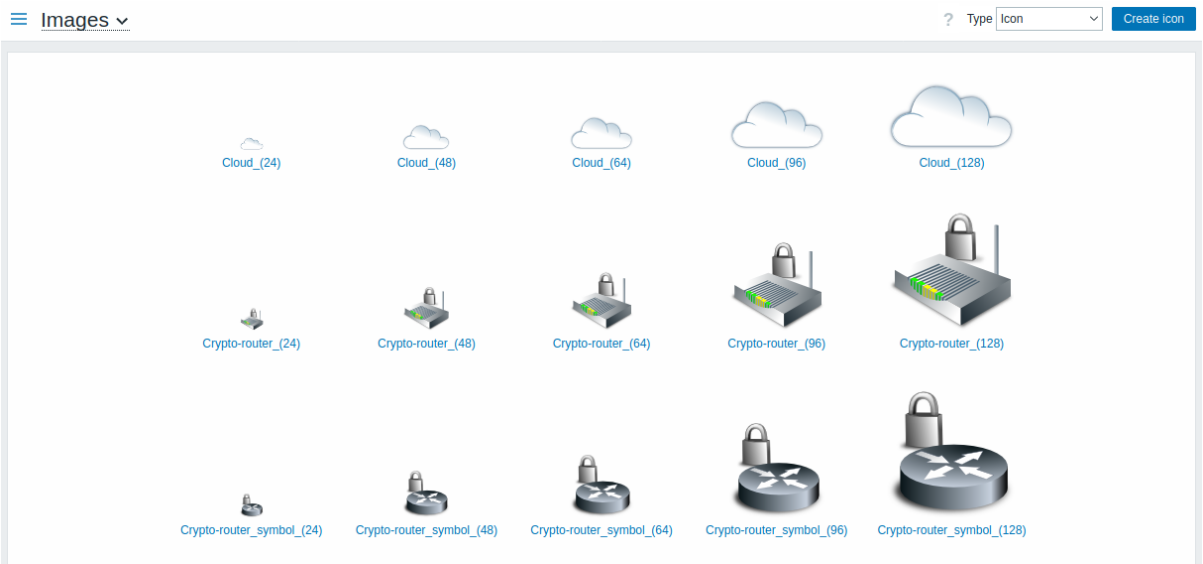
В этом разделе можно задать глобальные тайм-ауты для типов элементов данных и сетевые тайм-ауты. Все поля в этой форме являются обязательными.



Параметр		Описание
Тайм-ауты для типов элементов данных		Сколько секунд Zabbix должен ожидать перед прерыванием проверки (в зависимости от ее типа). Дополнительные сведения и ограничения см. в разделе тайм-аут элемента данных. Допустимый диапазон: 1 - 600s (по умолчанию: 3s; по умолчанию для элементов данных browser: 60s). Поддерживаются суффиксы времени, например 30s, 1m, а также пользовательские макросы. Обратите внимание, что тайм-ауты на уровне прокси, если они настроены, переопределяют эти глобальные параметры. Кроме того, если настроены тайм-ауты для конкретных элементов данных, они переопределяют как глобальные параметры, так и параметры прокси.
Сетевые тайм-ауты для UI		
	Обмен данными	Сколько секунд ожидать перед закрытием неактивного сокета (если соединение с сервером Zabbix было установлено ранее, но веб-интерфейс не может завершить операцию чтения/отправки данных в течение этого времени, соединение будет разорвано). Допустимый диапазон: 1 - 300s (по умолчанию: 3s).
	Соединение	Сколько секунд ожидать перед прекращением попытки подключения к серверу Zabbix. Допустимый диапазон: 1 - 30s (по умолчанию: 3s).
	Тест типа медиа	Сколько секунд ожидать ответа при тестировании типа медиа. Допустимый диапазон: 1 - 300s (по умолчанию: 65s).
	Выполнение скрипта	Сколько секунд ожидать ответа при выполнении скрипта. Допустимый диапазон: 1 - 300s (по умолчанию: 60s). Этот тайм-аут применяется ко всей цепочке выполнения скрипта, длина которой может быть различной. Например, если скрипт выполняется на агенте, это путь туда и обратно через сервер (возможно, также через прокси) к агенту и обратно. Для скриптов, выполняемых на агенте, применяется тайм-аут server или proxy. Для скриптов, выполняемых только на активном агенте, вероятно, потребуется увеличить стандартный тайм-аут сервера/прокси. Тайм-аут сервера/прокси должен быть больше частоты обновления активных проверок, иначе тайм-аут сервера/прокси может быть превышен еще до того, как активный агент вообще получит скрипт. См. также: Тайм-аут скрипта
	Тест элемента данных	Сколько секунд ожидать возвращаемые данные при тестировании элемента данных. Допустимый диапазон: 1 - 600s (по умолчанию: 60s).
	Тест запланированного отчета	Сколько секунд ожидать возвращаемые данные при тестировании запланированного отчета. Допустимый диапазон: 1 - 300s (по умолчанию: 60s).

Изображения

Раздел «Изображения» отображает все изображения, доступные в Zabbix. Изображения хранятся в базе данных.

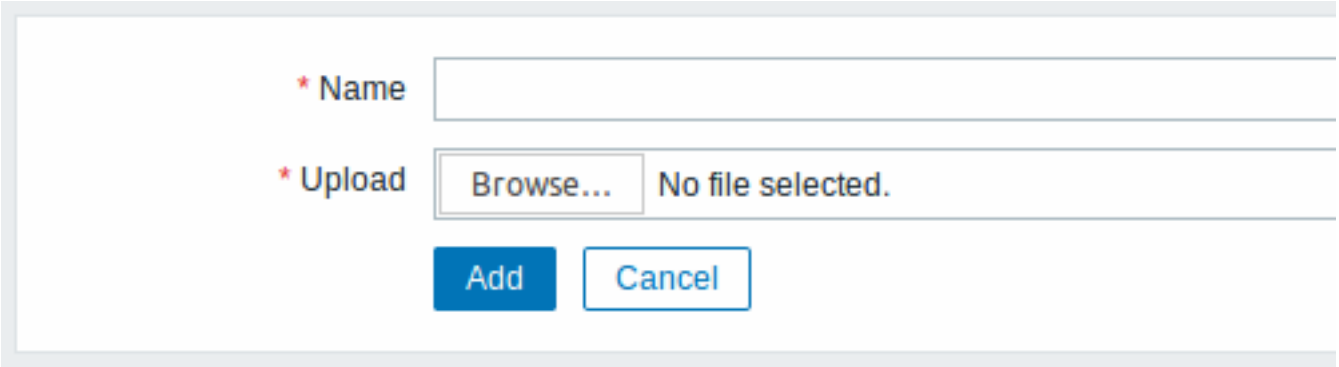


Выпадающий список *Тип* позволяет переключаться между изображениями значков и фоновыми изображениями:

- Значки используются для отображения элементов карты сети
- Фоны используются в качестве фоновых изображений карт сети

Добавление изображения

Вы можете добавить собственное изображение, нажав кнопку *Создать значок* или *Создать фон* в правом верхнем углу.



Атрибуты изображения:

Параметр	Описание
Имя	Уникальное имя изображения.
Загрузить	Выберите файл (PNG, JPEG, GIF или WebP) в локальной системе для загрузки в Zabbix. Обратите внимание, что может быть возможно загрузить и другие форматы, которые во время загрузки будут преобразованы в PNG. Для обработки изображений используется библиотека GD, поэтому поддерживаемые форматы зависят от используемой версии библиотеки (Zabbix требует версию 2.0.28 или выше).

Note:
Максимальный размер загружаемого файла ограничен значением ZBX_MAX_IMAGE_SIZE, которое составляет 1024x1024 байт или 1 МБ.

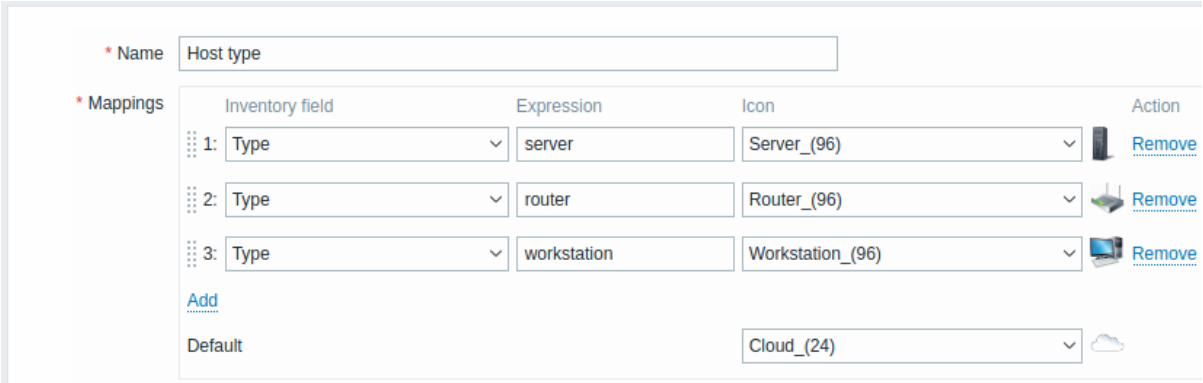
Загрузка изображения может завершиться ошибкой, если размер изображения близок к 1 МБ, а параметр конфигурации MySQL `max_allowed_packet` имеет значение по умолчанию 1MB. В этом случае увеличьте значение параметра max_allowed_packet.

Сопоставление иконок

Этот раздел позволяет создавать сопоставление определённых узлов сети с определёнными иконками. Для создания сопоставления используется информация из поля инвентарных данных узла сети.

Затем эти сопоставления можно использовать в настройке карты сети, чтобы автоматически назначать подходящие иконки соответствующим узлам сети.

Чтобы создать новое сопоставление иконок, нажмите *Создать сопоставление иконок* в правом верхнем углу.



Параметры настройки:

Параметр	Описание
Имя	Уникальное имя сопоставления иконок.
Сопоставления	Список сопоставлений. Порядок сопоставлений определяет, какое из них будет иметь приоритет. Вы можете перемещать сопоставления вверх и вниз по списку с помощью перетаскивания.
Поле инвентарных данных	Поле инвентарных данных узла сети, в котором будет выполняться поиск совпадения.
Выражение	Регулярное выражение, описывающее совпадение.
Иконка	Иконка, которая будет использоваться, если найдено совпадение с выражением.
По умолчанию	Иконка, используемая 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

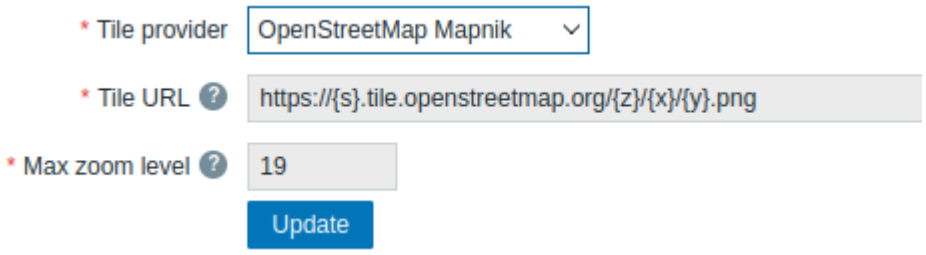
В этом разделе можно настроить, как состояние триггера отображается в веб-интерфейсе, а также имена и цвета важности триггера.



Параметр	Описание
Использовать пользовательские цвета статусов событий	Установка этого параметра включает настройку цветов 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 предложенной 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 статуса 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*, *Катастрофа*	Пользовательские имена уровней важности и/или цвета для отображения вместо системных значений по умолчанию. Введите новый код цвета или щелкните по цвету, чтобы выбрать новый из предложенной палитры. Обратите внимание, что пользовательские имена уровней важности, введенные здесь, будут использоваться во всех локалях. Если вам нужно перевести их на другие языки для определенных пользователей, смотрите страницу Настройка важностей триггера.

Географические карты

В этом разделе можно выбрать поставщика сервиса тайлов географических карт и настроить параметры поставщика сервиса для виджета Geomap панели мониторинга. Для визуализации с использованием географических карт Zabbix использует Leaflet — интерактивную JavaScript-библиотеку карт с открытым исходным кодом. Обратите внимание, что Zabbix не контролирует качество изображений, предоставляемых сторонними поставщиками тайлов, включая предопределённых поставщиков тайлов.



Параметр	Описание
Поставщик тайлов	Выберите одного из доступных поставщиков сервиса тайлов или выберите *Other*, чтобы добавить другого поставщика тайлов или самостоятельно размещаемые тайлы (см. Использование пользовательского поставщика сервиса тайлов).
URL тайлов	Шаблон URL (до 2048 символов) для загрузки и отображения слоя тайлов на географических картах. Это поле можно редактировать, только если для *Tile provider* установлено значение *Other*. Поддерживаются следующие заполнители: *{s}* обозначает один из доступных поддоменов; *{z}* обозначает параметр уровня масштабирования в URL; *{x}* и *{y}* обозначают координаты тайла; *{r}* можно использовать для добавления "@2x" в URL для загрузки тайлов retina. Пример: `https://{s}.example.com/{z}/{x}/{y}{r}.png`
Текст атрибуции	Текст атрибуции поставщика тайлов, отображаемый в небольшом текстовом поле на карте. Это поле отображается, только если для *Tile provider* установлено значение *Other*.

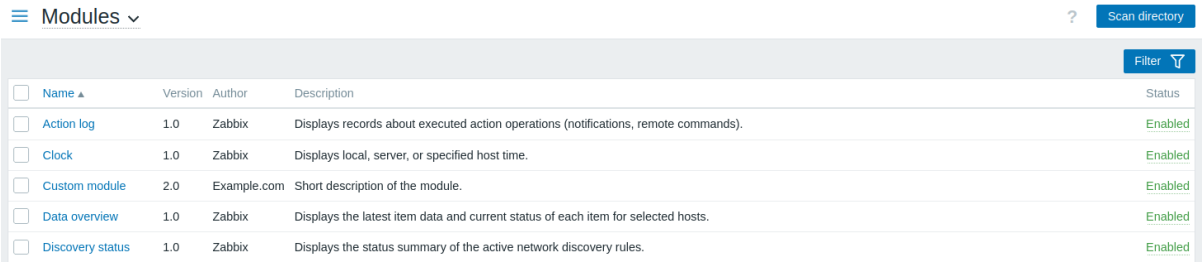
Параметр	Описание
Максимальный уровень масштабирования	Максимальный уровень масштабирования карты. Это поле можно редактировать, только если для *Tile provider* установлено значение *Other*.

Использование пользовательского поставщика сервиса тайлов

Виджет Geomap может загружать растровые изображения тайлов из пользовательского самостоятельно размещенного или стороннего сервиса поставщика тайлов. Чтобы использовать пользовательский сторонний сервис поставщика тайлов или самостоятельно размещенную папку тайлов либо сервер, выберите *Other* в поле *Tile provider* и укажите пользовательский URL в поле *Tile URL*, используя соответствующие заполнители.

Модули

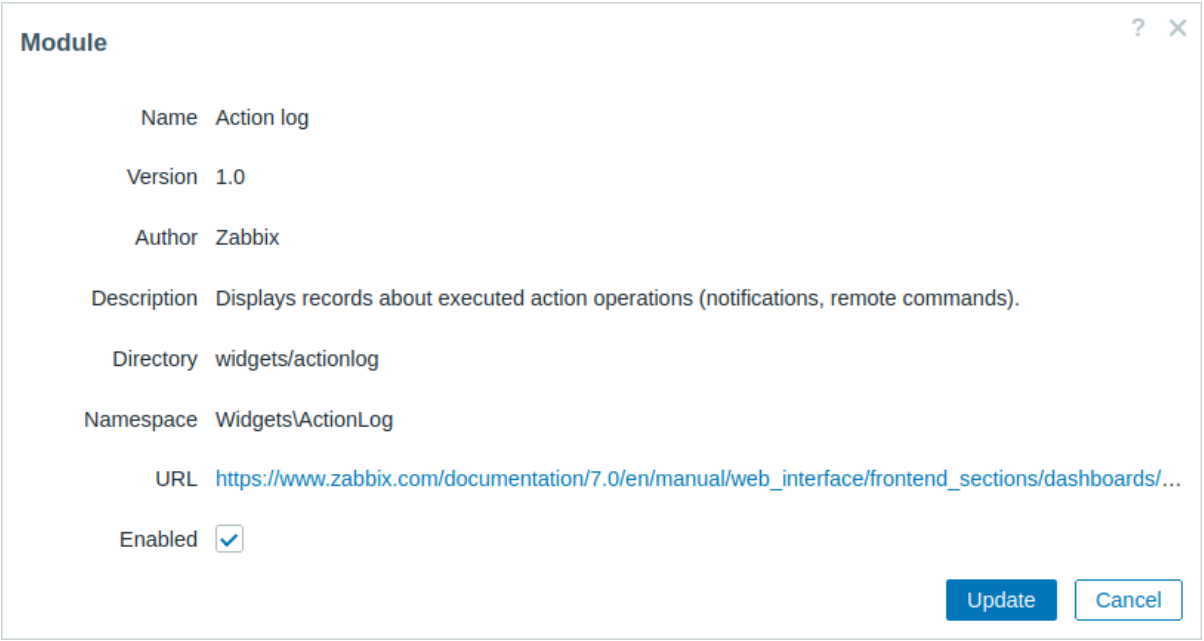
Этот раздел позволяет администрировать пользовательские, а также встроенные модули веб-интерфейса.



Нажмите *Сканировать каталог*, чтобы зарегистрировать/отменить регистрацию любых пользовательских модулей. Зарегистрированные модули появятся в списке; незарегистрированные модули будут удалены из списка.

Нажмите на статус модуля в списке, чтобы включить/отключить модуль. Вы также можете массово включать/отключать модули, выбрав их в списке, а затем нажав кнопки *Включить/Отключить* под списком.

Нажмите на имя модуля в списке, чтобы просмотреть его сведения во всплывающем окне.

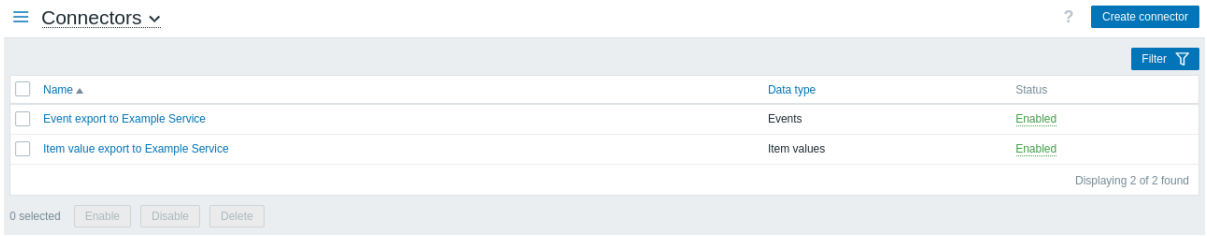


Статус модуля также можно изменить во всплывающем окне сведений о модуле; для этого установите/снимите флажок *Включено*, а затем нажмите *Обновить*.

Вы можете фильтровать модули по имени или статусу (включен/отключен).

Коннекторы

В этом разделе можно настроить коннекторы для потоковой передачи данных Zabbix во внешние системы по HTTP.



Нажмите *Создать коннектор*, чтобы настроить новый коннектор.

Вы можете фильтровать коннекторы по имени или статусу (включен/отключен). Нажмите на статус коннектора в списке, чтобы включить/отключить коннектор.
Вы также можете массово включать/отключать коннекторы, выбрав их в списке, а затем нажав кнопки *Включить/Отключить* под списком.

Прочее

В этом разделе можно настроить различные прочие параметры веб-интерфейса.



Параметр	Описание
URL веб-интерфейса	URL (до 2048 символов) веб-интерфейса Zabbix. Этот параметр используется веб-службой Zabbix для связи с веб-интерфейсом и должен быть указан для включения запланированных отчетов.

Параметр	Описание
Группа для обнаруженных узлов сети	Узлы сети, обнаруженные с помощью обнаружения сети и авторегистрации агента, будут автоматически помещены в группу узлов сети, выбранную здесь.
Режим инвентаризации узла сети по умолчанию	Режим инвентаризации по умолчанию для узла сети. Он будет использоваться каждый раз, когда новый узел сети или прототип узла сети создается сервером или веб-интерфейсом, если только он не переопределен во время обнаружения узла сети/авторегистрации операцией *Установить режим инвентаризации узла сети*.
Группа пользователей для сообщения о недоступности базы данных	Группа пользователей для отправки тревожного сообщения или значение 'None'. 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 отправлены в указанную здесь группу пользователей с использованием включенных записей способов оповещения пользователей. Уведомления передаются с помощью следующих типов медиа (если включены): email; SMS; пользовательские скрипты оповещений. Даже если запись медиа типа вебхук настроена и включена, она не будет инициировать уведомления. Сервер Zabbix не остановится; он будет ждать, пока база данных снова не станет доступной, 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 он определен как 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 оповещений, но не чаще чем раз в 15 минут.
Журналировать несопоставленные SNMP trap	Журналировать SNMP trap, если не найдено соответствующих SNMP-интерфейсов.

Авторизация

Параметр	Описание
Попытки входа	Количество неудачных попыток входа, после которого возможность входа будет заблокирована.
Интервал блокировки входа	Период времени, в течение которого вход будет запрещён при превышении лимита *Попытки входа*. Допустимый диапазон: 0 - 3600 секунд. Поддерживаются суффиксы времени, например 90s, 5m, 1h.

Хранение секретов

Parameter	Description
Поставщик хранилища секретов	Выберите программное обеспечение для управления секретами для хранения значений пользовательских макросов — *HashiCorp Vault* (по умолчанию) или *CyberArk Vault*.
Разрешать макросы хранилища секретов с помощью	Разрешать макросы хранилища секретов с помощью: **сервера Zabbix** — секреты извлекаются сервером Zabbix и при необходимости передаются прокси (по умолчанию); **сервера Zabbix и прокси** — секреты извлекаются как сервером Zabbix, так и прокси, что позволяет им разрешать макросы независимо.

См. также: Хранение секретов.

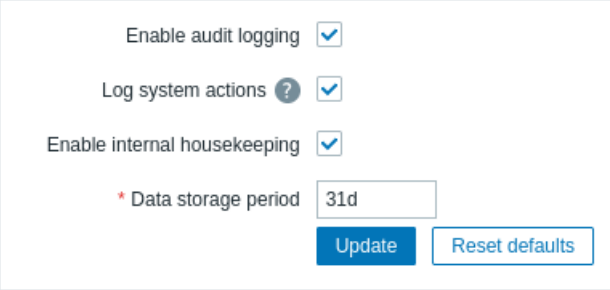
Безопасность

Parameter	Description
Проверять схемы URI	Снимите этот флажок, чтобы отключить проверку схем URI (по умолчанию включена). Если флажок установлен, вы можете указать разделённый запятыми список разрешённых схем URI (по умолчанию: http,https,ftp,file,mailto,tel,ssh). Применяется ко всем полям в веб-интерфейсе, где используются URI (например, URL элементов карты).
Использовать HTTP-заголовок X-Frame-Options	Снимите этот флажок, чтобы отключить HTTP-заголовок X-Frame-options (не рекомендуется). Если флажок установлен, вы можете указать значение HTTP-заголовка X-Frame-options. Поддерживаемые значения: **SAMEORIGIN** (по умолчанию) или **'self'** (должно быть заключено в одинарные кавычки) — страница может отображаться во фрейме только с тем же источником, что и сама страница; **DENY** или **'none'** (должно быть заключено в одинарные кавычки) — страница не может отображаться во фрейме независимо от сайта, пытающегося это сделать; **строка имён узлов сети, разделённых пробелами**; добавление **'self'** (должно быть заключено в одинарные кавычки) в список позволяет отображать страницу во фрейме с тем же источником, что и сама страница. Обратите внимание, что использование **'self'** или **'none'** без одинарных кавычек приведёт к тому, что они будут рассматриваться как имена узлов сети.
Использовать sandboxing iframe	Снимите этот флажок, чтобы отключить помещение содержимого полученного URL в sandbox (не рекомендуется). Если флажок установлен, вы можете указать исключения sandboxing для iframe; неуказанные ограничения всё равно будут применяться. Если это поле пустое, применяются все ограничения атрибута sandbox. Для получения дополнительной информации см. описание атрибута sandbox.

2 Журнал аудита

Обзор

В этом разделе можно настроить параметры журнала аудита.



Доступны следующие параметры:

Параметр	Описание
Включить журналирование аудита	Включить (по умолчанию) или отключить журналирование аудита.
Журналировать системные действия	Включить (по умолчанию) или отключить журналирование аудита действий по низкоуровневому обнаружению, сетевому обнаружению и авторегистрации, выполняемых сервером (пользователь System).
Включить внутреннюю очистку	Включить (по умолчанию) или отключить внутреннюю очистку записей журнала аудита.
Период хранения данных	Количество дней, в течение которых записи журнала аудита должны храниться до их удаления процессом очистки. Обязательно, если очистка включена. По умолчанию: 31 день.

Параметры сжатия журнала аудита доступны в разделе *Администрирование* > *Очистка данных*, в блоке *Сжатие истории, трендов и журнала аудита*, который становится видимым при использовании TimescaleDB.

3 Очистка истории

Обзор

Раздел *Administration* > *Housekeeping* позволяет настроить внутреннюю очистку Zabbix, которая автоматически удаляет из базы данных устаревшие или удаленные пользователем данные. Это предотвращает неограниченный рост базы данных, который привел бы к ненужному расходу дискового пространства и снижению производительности базы данных.

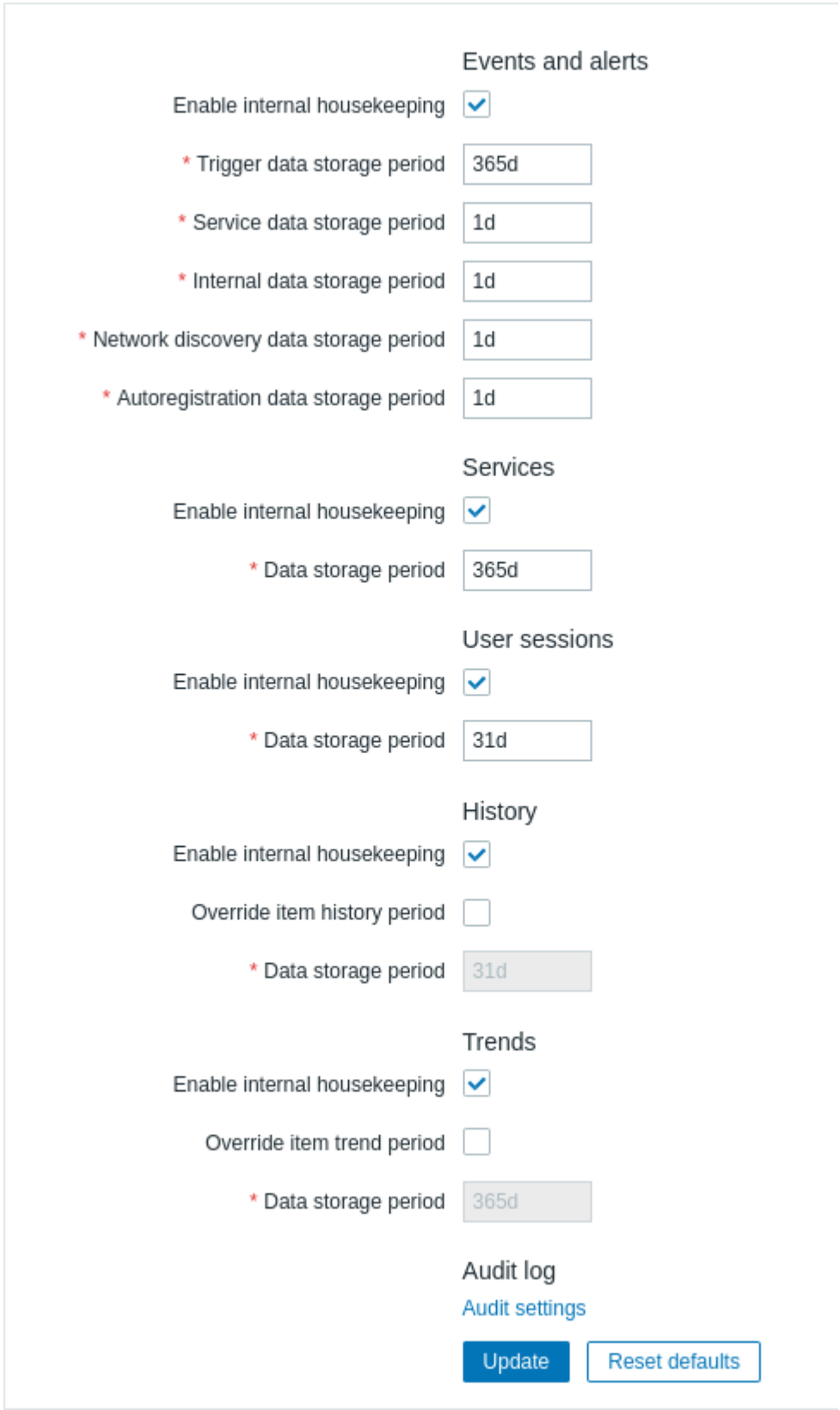
Очистка может быть включена и настроена для следующих типов данных:

- События и оповещения от триггеров, сервисов, внутренних данных, обнаружения сети и авторегистрации
- Сервисы
- Пользовательские сессии
- История и тренды для элементов данных

Note:
Очистка журнала аудита настраивается отдельно.

Конфигурация

Доступны следующие параметры:



Параметр	Описание
Включить внутреннюю очистку	Включить или отключить внутреннюю очистку (по умолчанию включена). Если включено, процедура очистки автоматически удаляет из базы данных данные, превышающие *Период хранения данных*.

Параметр	Описание
Период хранения данных	Укажите, как долго хранить данные до их удаления очисткой. Обязательно, если внутренняя очистка включена. Диапазон: от 1 дня (1 часа для истории) до 25 лет; или ”0”. Поддерживаются суффиксы времени (например, 1d, 1w). Для *Событий и оповещений* период хранения данных задается отдельно для триггеров, сервисов, внутренних данных, обнаружения сети и авторегистрации. Очистка удаляет только те события, которые не связаны с проблемами. Например, событие проблема/восстановление старше *Периода хранения данных* не будет удалено, если оно связано с нерешенной проблемой. Когда очистка удаляет устаревшие сущности, сначала удаляются проблемы, затем события. Для *Истории* и *Трендов* периоды хранения данных также определяют, как долго данные остаются видимыми в *Мониторинг* > *Последние данные*, даже если внутренняя очистка отключена. Обратите внимание, что при удалении элемента данных или триггера также удаляются его проблемы и связанные события.
Переопределить период хранения истории элемента данных	Если включено, период хранения истории/трендов, указанный в конфигурации элемента данных, переопределяется настройкой *Период хранения данных* (кроме элементов данных, для которых включена опция *Не хранить*). Эту опцию можно использовать даже при отключенной внутренней очистке и использовании внешней очистки.
Переопределить период хранения трендов элемента данных	

При использовании **TimescaleDB** становится доступен раздел *Сжатие истории, трендов и журнала аудита*.

Note:
Для TimescaleDB включите *Переопределить период хранения истории элемента данных*, *Переопределить период хранения трендов элемента данных* и *Включить внутреннюю очистку* для истории и трендов, чтобы в полной мере воспользоваться автоматическим разбиением на разделы. Если эти параметры отключены, данные, хранящиеся в таблицах истории и трендов, все равно будут разбиты на разделы, но очистка не будет удалять устаревшие разделы, и появятся предупреждения конфигурации. Когда включено удаление устаревших разделов, сервер Zabbix и веб-интерфейс больше не будут отслеживать удаленные элементы данных, а история для этих элементов данных будет очищаться при удалении устаревшего раздела.

Кнопка *Сбросить по умолчанию* позволяет отменить все внесенные изменения.

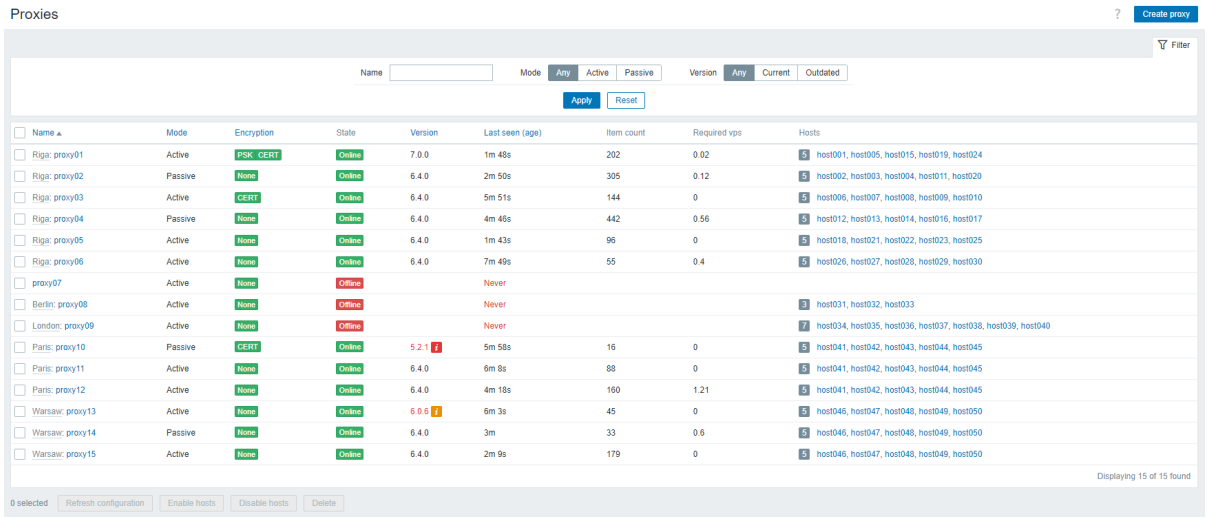
4 Прокси

Обзор

В разделе *Администрирование* → *Прокси* располагаются настройки относящиеся к распределенному мониторингу и которые можно настроить в веб-интерфейсе Zabbix.

Прокси

Отображается список существующих прокси с подробной информацией о них.



Отображаемые данные:

Column	Description
Name	Имя прокси. Щелчок по имени прокси открывает форму настройки прокси. Если прокси принадлежит группе прокси, имя группы отображается перед именем прокси в виде серой ссылки. Щелчок по имени группы открывает форму настройки группы прокси.
Mode	Режим прокси — *Активный* или *Пассивный*.
Encryption	Состояние шифрования для соединений от прокси: **None** — без шифрования; **PSK** — с использованием предварительно общего ключа; **Cert** — с использованием сертификата.
State	Состояние прокси: **Unknown** — прокси был создан, когда сервер Zabbix был недоступен, или сервер еще не обновил состояние; **Online** — прокси обменивался данными с сервером Zabbix (пассивный прокси ответил на запрос сервера; активный прокси отправил запрос) в течение периода отказоустойчивости; **Offline** — прокси не обменивался данными с сервером Zabbix в течение периода отказоустойчивости.
Version	Версия прокси (трехзначный номер версии). Если прокси устарел или не поддерживается, номер версии выделяется красным цветом, а также отображается значок информационного статуса (желтый или красный). Наведите курсор на значок, чтобы увидеть подробности.
Last seen (age)	Время, когда прокси в последний раз был замечен сервером.
Item count	Количество включенных элементов данных на включенных узлах сети, назначенных прокси.
Required vps	Требуемая производительность прокси (количество значений, которые необходимо собирать в секунду).
Hosts	Количество включенных узлов сети, назначенных прокси, и список узлов сети, которые мониторятся этим прокси. Щелчок по имени узла сети открывает форму настройки узла сети.

Чтобы настроить новый прокси, нажмите кнопку *Create proxy* в правом верхнем углу.

Параметры массового редактирования

Кнопки под списком предлагают несколько вариантов массового редактирования:

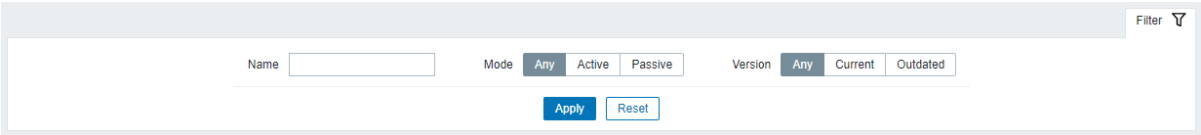
- *Refresh configuration* — обновить конфигурацию прокси;
- *Enable hosts* — изменить статус узлов сети, отслеживаемых прокси, на *Monitored*;
- *Disable hosts* — изменить статус узлов сети, отслеживаемых прокси, на *Not monitored*;
- *Delete* — удалить прокси.

Чтобы использовать эти параметры, установите флажки напротив соответствующих прокси, затем нажмите нужную кнопку.

Использование фильтра

Вы можете использовать фильтр, чтобы отображать только те прокси, которые вас интересуют. Для повышения производительности поиска данные ищутся с неразрешёнными макросами.

Ссылка *Фильтр* доступна над списком прокси. Если вы нажмёте на неё, станет доступен фильтр, в котором можно отфильтровать прокси по имени, режиму и версии. Обратите внимание, что параметр фильтра *Устаревшие* отображает как устаревшие (частично поддерживаемые), так и неподдерживаемые прокси.



5 Группы прокси

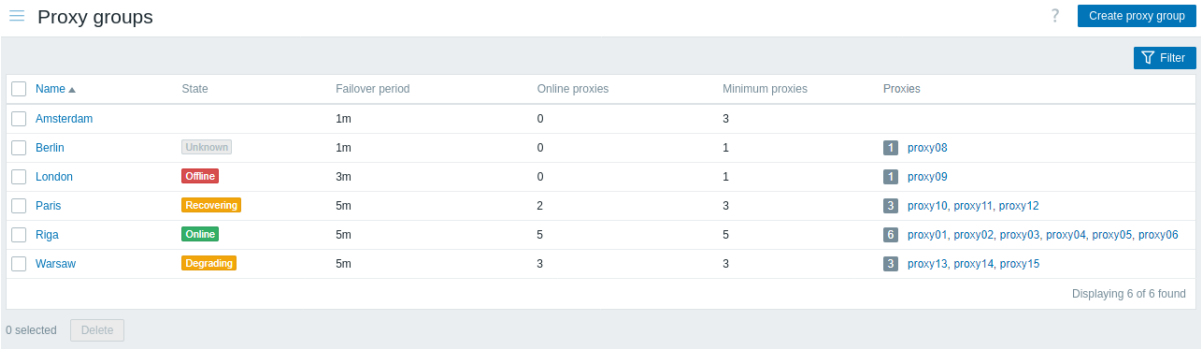
Обзор

В *Administration* → *Proxy groups* можно настроить группы прокси.

Группы прокси используются в балансировке нагрузки прокси с автоматическим распределением узлов сети между прокси и высокой доступностью между прокси.

Группы прокси

Отображается список существующих групп прокси с подробной информацией о них.



Отображаемые данные:

Column	Description
Name	Имя группы прокси. Нажатие на имя группы прокси открывает форму настройки группы прокси.
State	Состояние группы прокси (отображается, если группа содержит хотя бы один прокси): **Unknown** — группа была создана, когда сервер Zabbix был недоступен, или сервер еще не обновил состояние; **Online** — минимальное количество прокси обменивалось данными с сервером Zabbix в течение периода переключения при отказе; **Degrading** — группа вскоре перейдет в состояние офлайн, так как количество прокси в сети стало ниже порога *Minimum proxies*; **Offline** — с сервером Zabbix обменивалось данными меньше минимального количества прокси в течение периода переключения при отказе; **Recovering** — группа вскоре снова перейдет в состояние онлайн, так как количество прокси в сети достигло порога *Minimum proxies*. Состояние группы прокси можно отслеживать с помощью элементов данных zabbix[proxy group,<name>,state] и zabbix[proxy group,discovery].
Failover period	Период, в течение которого прокси в группе прокси должен обмениваться данными с сервером Zabbix, чтобы считаться в сети.
Online proxies	Количество прокси в сети (отображается красным цветом, если оно ниже группового минимума).
Minimum proxies	Минимальное количество прокси в сети, необходимое для поддержания группы прокси в состоянии онлайн.

Column	Description
Proxies	Количество прокси в группе и список прокси в группе со ссылками на форму настройки прокси. Максимальное количество отображаемых прокси ограничено значением *Max count of elements to show inside table cell*.

Чтобы настроить новую группу прокси, нажмите кнопку *Create proxy groups* в правом верхнем углу.

Параметры массового редактирования

Кнопки под списком предлагают несколько вариантов массового редактирования:

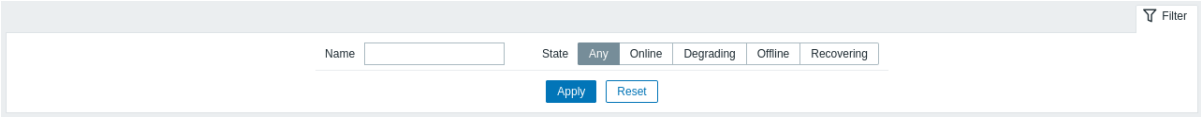
- *Удалить* — удалить группы прокси.

Чтобы использовать эти параметры, установите флажки напротив соответствующих групп прокси, затем нажмите нужную кнопку.

Использование фильтра

Вы можете использовать фильтр, чтобы отображать только те группы прокси, которые вас интересуют. Для повышения производительности поиска данные ищутся с неразрешёнными макросами.

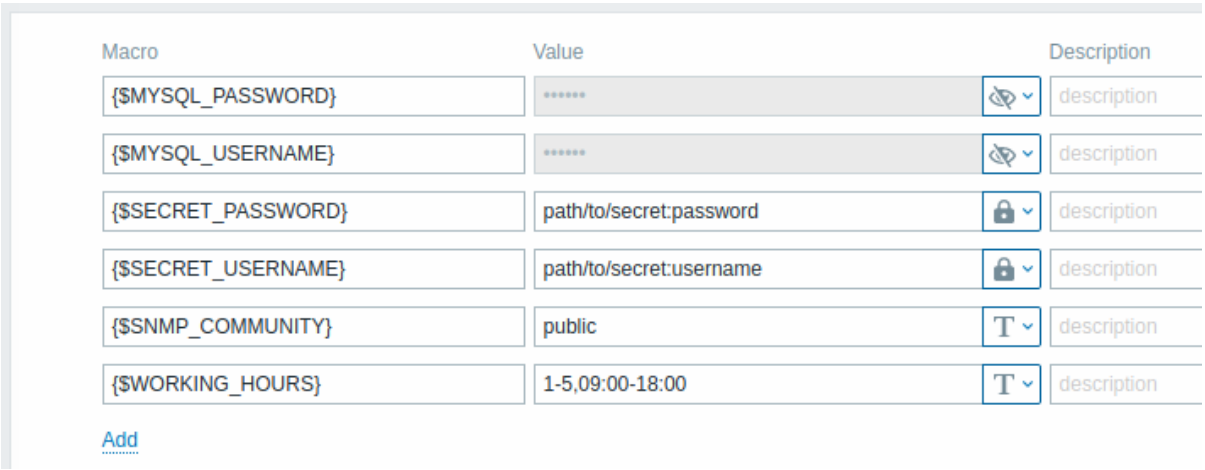
Ссылка *Фильтр* доступна над списком групп прокси. Если щёлкнуть по ней, станет доступен фильтр, в котором можно отфильтровать группы прокси по имени и состоянию.



6 Макросы

Обзор

Этот раздел позволяет определять общесистемные пользовательские макросы как пары «имя-значение». Обратите внимание, что значения макросов могут храниться как обычный текст, секретный текст или секрет Vault. Также поддерживается добавление описания.



7 Очередь

Обзор

В разделе *Администрирование* → *Очередь* отображаются элементы данных, ожидающие обновления.

В идеале при открытии этого раздела всё должно быть «зелёным», то есть в очереди не должно быть элементов данных. Если все элементы данных обновляются без задержки, то ожидающих нет. Однако из-за недостаточной производительности сервера некоторые элементы данных могут задерживаться, и эта информация отображается в данном разделе. Подробнее см. в разделе Очередь.

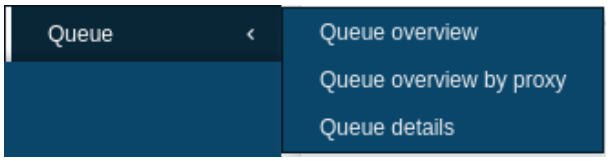
Note:
Прокси Zabbix не учитывает периоды обслуживания; подробности см. в разделе Расчёт очередей во время обслуживания.

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 если агент работает некорректно.

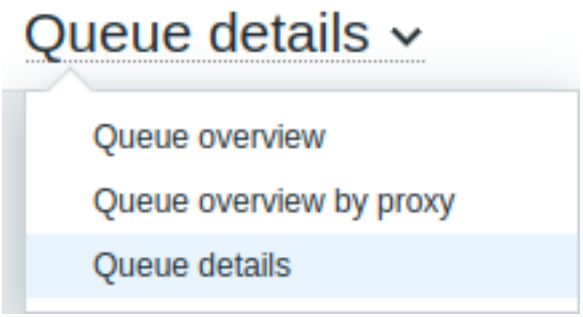
Раздел *Администрирование* → *Очередь* содержит следующие страницы:

- Обзор очереди — отображает очередь по типу элементов данных;
- Обзор очереди по прокси — отображает очередь по прокси;
- Подробности очереди — отображает список задержанных элементов данных.

Список доступных страниц появляется при нажатии на *Очередь* в разделе меню *Администрирование*. Также можно переключаться между страницами с помощью выпадающего списка заголовков в левом верхнем углу.



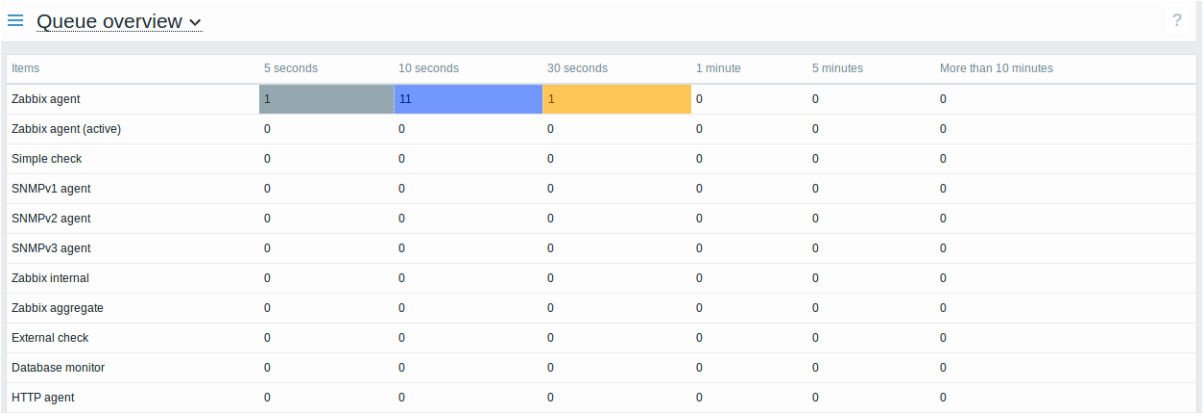
Меню третьего уровня.



Выпадающий список заголовков.

Обзор по типу элемента данных

На этом экране легко определить, связана ли проблема с одним или несколькими типами элементов данных.

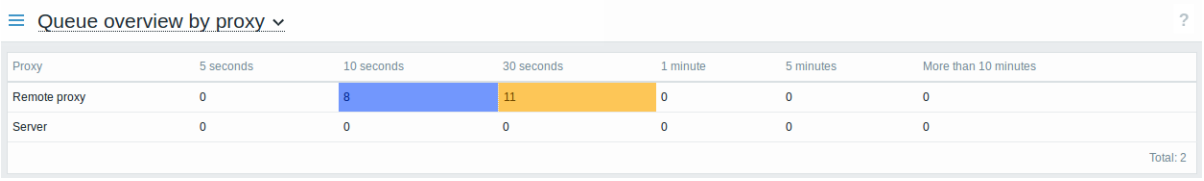

Queue overview

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

Каждая строка содержит тип элемента данных. Каждый столбец показывает количество ожидающих элементов данных — ожидающих 5–10 секунд/10–30 секунд/30–60 секунд/1–5 минут/5–10 минут или более 10 минут соответственно.

Обзор по прокси

На этом экране легко определить, связана ли проблема с одним из прокси или сервером.


Queue overview by proxy

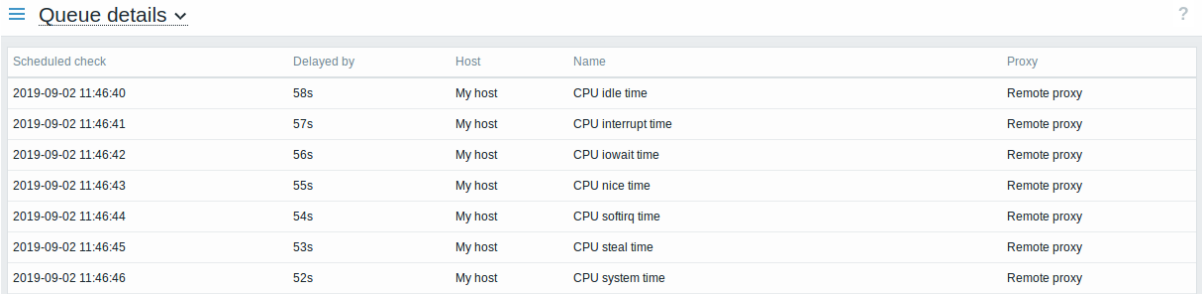
Proxy	5 seconds	10 seconds	30 seconds	1 minute	5 minutes	More than 10 minutes
Remote proxy	0	8	11	0	0	0
Server	0	0	0	0	0	0

Total: 2

Каждая строка содержит прокси, при этом сервер находится последним в списке. Каждый столбец показывает количество ожидающих элементов данных — ожидающих 5–10 секунд/10–30 секунд/30–60 секунд/1–5 минут/5–10 минут или более 10 минут соответственно.

Список ожидающих элементов данных

На этом экране перечислены все ожидающие элементы данных.



Отображаемые данные:

Столбец	Описание
Запланированная проверка	Отображается время, когда должна была быть выполнена проверка.
Задержка	Отображается длительность задержки.
Узел сети	Отображается узел сети элемента данных.
Имя	Отображается имя ожидающего элемента данных.
Прокси	Отображается имя прокси, если узел сети мониторится прокси.

Возможные сообщения об ошибках

Вы можете столкнуться с ситуацией, когда данные не отображаются и появляется следующее сообщение об ошибке:



Сообщение об ошибке в этом случае следующее:

`Cannot display item queue. Permission denied`

Это происходит, когда параметры конфигурации PHP в файле *zabbix.conf.php* — `$ZBX_SERVER` или оба `$ZBX_SERVER` и `$ZBX_SERVER_PORT` — указывают на существующий сервер Zabbix, использующий другую базу данных.

6 Сбор данных

Обзор

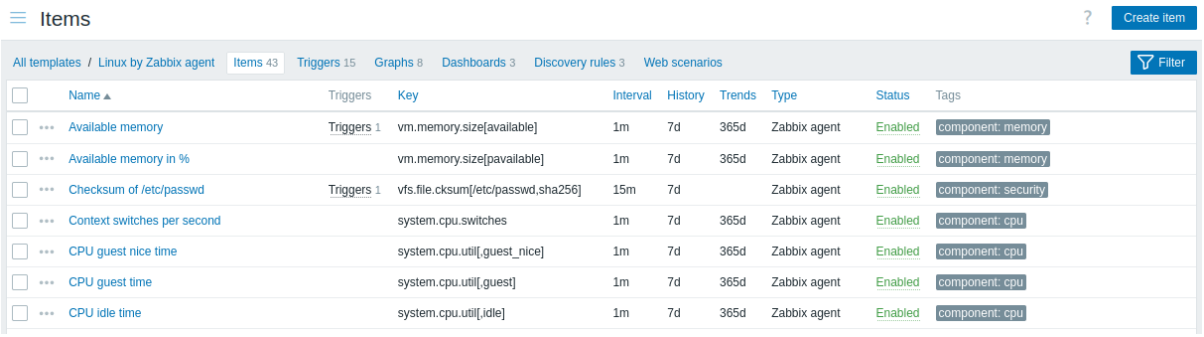
Это меню содержит разделы, связанные с настройкой сбора данных.

1 Элементы данных

Обзор

Список элементов данных для шаблона можно открыть в разделе *Сбор данных* → *Шаблоны*, щелкнув по *Элементы данных* у соответствующего шаблона.

Отображается список существующих элементов данных.



Отображаемые данные:

Столбец	Описание
Меню элемента данных	Щелкните по значку с тремя точками, чтобы открыть меню элемента данных.
Шаблон	Шаблон, которому принадлежит элемент данных. Щелчок по имени шаблона открывает форму 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 отображается как синяя ссылка на сведения об элементе данных. Щелчок по ссылке с именем элемента данных открывает форму настройки элемента данных. Если элемент данных унаследован от другого шаблона, перед именем элемента данных отображается имя шаблона в виде серой ссылки. Щелчок по ссылке шаблона открывает список элементов данных на уровне этого шаблона.
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.
История	Отображается, сколько дней будет храниться история данных элемента данных.
Тренды	Отображается, сколько дней будет храниться история трендов элемента данных.
Тип	Отображается тип элемента данных (агент Zabbix, SNMP-агент, простая проверка и т. д.).
Состояние	Отображается состояние элемента данных — *Включено* или *Отключено*. Щелчком по состоянию его можно изменить — с Включено на Отключено (и обратно).
Теги	Отображаются теги элемента данных. Может отображаться до трех тегов (пары имя:значение). Если тегов больше, отображается ссылка "..." , позволяющая увидеть все теги при наведении указателя мыши.

Чтобы настроить новый элемент данных, нажмите кнопку *Создать элемент данных* в правом верхнем углу.

Параметры массового редактирования

Кнопки под списком предоставляют несколько вариантов массового редактирования:

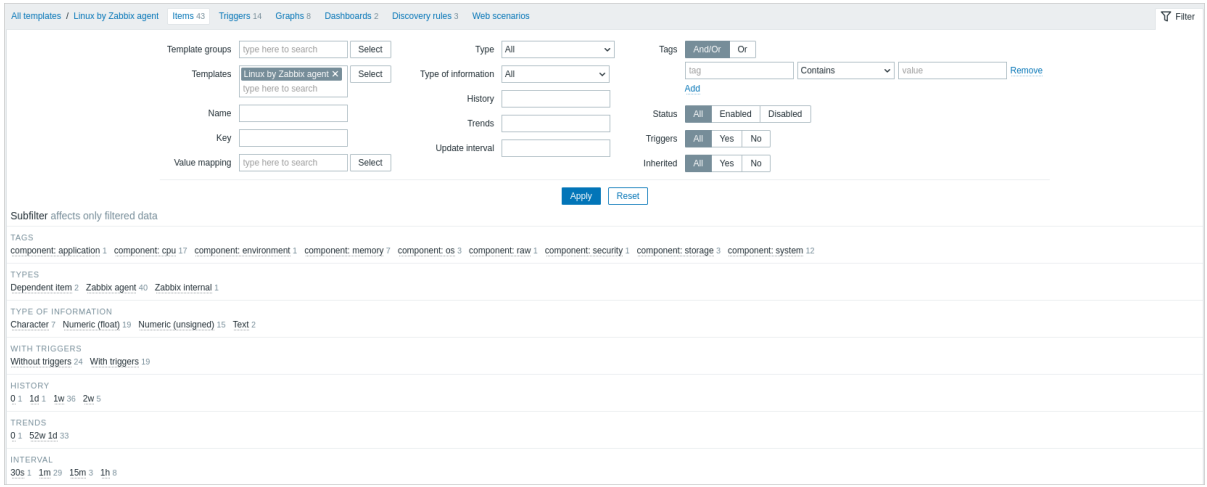
- *Включить* — изменить статус элемента данных на *Включено*.
- *Отключить* — изменить статус элемента данных на *Отключено*.
- *Копировать* — скопировать элементы данных на другие узлы сети или шаблоны.
- *Массовое обновление* — обновить несколько свойств сразу для нескольких элементов данных.
- *Удалить* — удалить элементы данных.

Чтобы использовать эти параметры, установите флажки напротив соответствующих элементов данных, затем нажмите нужную кнопку.

Использование фильтра

Список элементов данных может содержать большое количество элементов данных. Используя фильтр, вы можете отфильтровать некоторые из них, чтобы быстро найти нужные элементы данных. Для повышения производительности поиска данные ищутся с неразрешёнными макросами.

Значок *Фильтр* доступен в правом верхнем углу. Щелчок по нему откроет фильтр, где вы можете указать нужные критерии фильтрации.

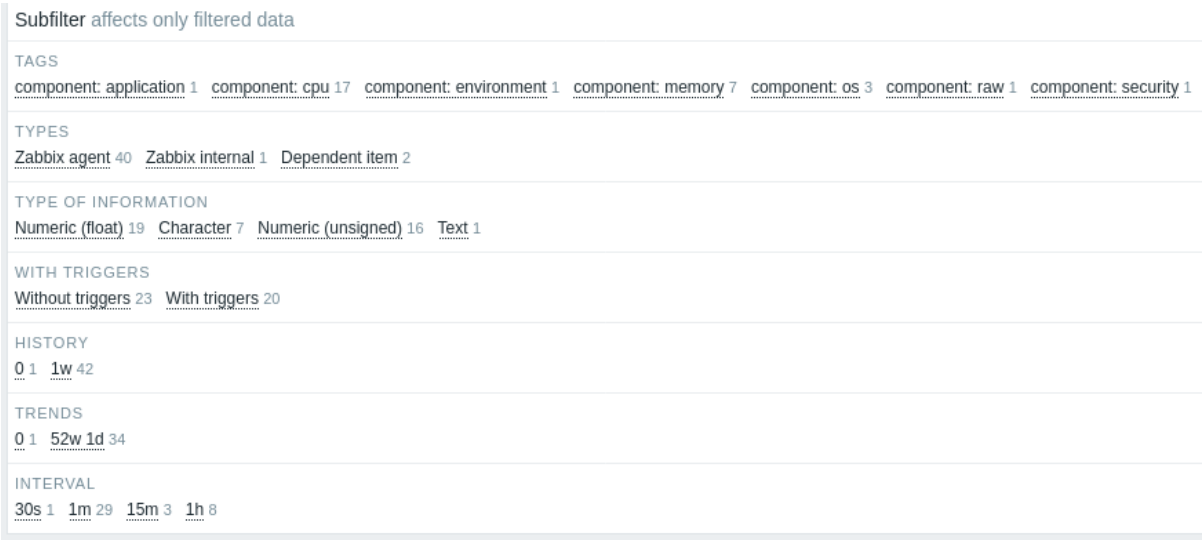


Параметр	Описание
Группы шаблонов	Фильтрация по одной или нескольким группам шаблонов. Указание родительской группы шаблонов неявно выбирает все вложенные группы.
Шаблоны	Фильтрация по одному или нескольким шаблонам.
Имя	Фильтрация по имени элемента данных.
Ключ	Фильтрация по ключу элемента данных.
Преобразование значений	Фильтрация по используемой карте значений. Этот параметр не отображается, если опция *Шаблоны* пуста.
Тип	Фильтрация по типу элемента данных (агент Zabbix, SNMP-агент и т. д.).
Тип информации	Фильтрация по типу информации (Numeric unsigned, float и т. д.).
История	Фильтрация по тому, как долго хранится история элемента данных.
Тренды	Фильтрация по тому, как долго хранятся тренды элемента данных.
Интервал обновления	Фильтрация по интервалу обновления элемента данных.
Теги	Укажите теги, чтобы ограничить количество отображаемых элементов данных. 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 доступно несколько операторов: **Существует** — включить указанные имена тегов **Равно** — включить указанные имена и значения тегов (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 и значения тегов (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** — все условия должны быть выполнены, условия с одинаковым именем тега будут сгруппированы по условию Или **Или** — достаточно выполнения одного условия
Состояние	Фильтрация по состоянию элемента данных — *Включено* или *Отключено*.
Триггеры	Фильтрация элементов данных с триггерами (или без них).
Унаследовано	Фильтрация элементов данных, унаследованных (или не унаследованных) из связанных шаблонов.

Использование подфильтра

Подфильтр позволяет дополнительно изменять фильтрацию из основного фильтра.

Он содержит кликабельные ссылки для быстрого доступа к связанным элементам данных. Элементы данных связаны общей сущностью — тегом, типом элемента данных, интервалом обновления элемента данных и т. д. При нажатии на сущность она выделяется серым фоном, а элементы данных немедленно фильтруются (нажимать *Применить* в основном фильтре не нужно). Нажатие на другую сущность добавляет её в отфильтрованные результаты. Повторное нажатие на сущность убирает фильтрацию.



Подфильтры формируются на основе отфильтрованных данных, количество которых ограничено 1000 записями. Если вы хотите видеть больше записей в подфильтре, необходимо увеличить значение параметра *Limit for search and filter results* (в *Administration -> General -> GUI*).

В отличие от основного фильтра, подфильтр обновляется при каждом запросе на обновление таблицы, чтобы всегда содержать актуальную информацию о доступных вариантах фильтрации и количестве элементов в них.

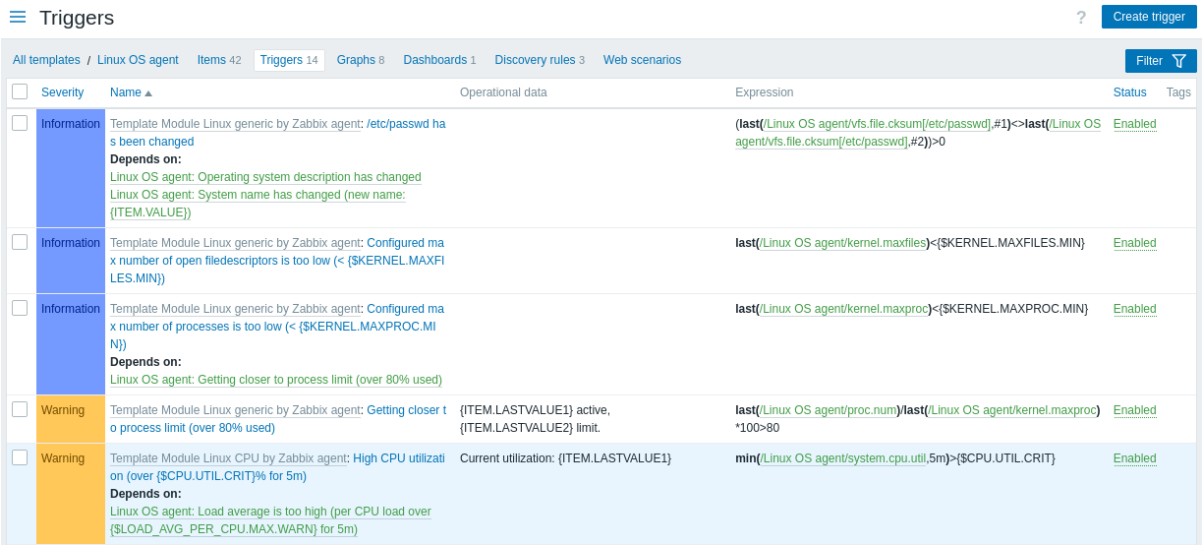
Количество отображаемых сущностей по горизонтали ограничено 100. Если их больше, в конце отображается значок с тремя точками; он не является кликабельным.

Число рядом с каждой кликабельной сущностью показывает количество элементов данных, сгруппированных в ней (на основе результатов основного фильтра). При нажатии на сущность числа у других доступных сущностей отображаются со знаком плюс, указывая, сколько элементов данных может быть добавлено к текущему выбору. Сущности без элементов данных не отображаются, если только они не были ранее выбраны в подфильтре.

2 Триггеры

Обзор

Список триггеров для шаблона можно открыть в разделе *Сбор данных* → *Шаблоны*, щелкнув по *Триггеры* для соответствующего шаблона.



Отображаемые данные:

Column	Description
Severity	Важность триггера отображается как по названию, так и по цвету фона ячейки.

Column	Description
Template	Шаблон, к которому относится триггер. Щелчок по имени шаблона открывает форму настройки шаблона. Этот столбец отображается только в том случае, если в фильтре выбрано несколько шаблонов или не выбран ни один шаблон.
Name	Имя триггера отображается как синяя ссылка на сведения о триггере. Щелчок по ссылке с именем триггера открывает форму настройки триггера. Если триггер унаследован от другого шаблона, имя шаблона отображается перед именем триггера в виде серой ссылки. Щелчок по ссылке шаблона открывает список триггеров на уровне этого шаблона.
Operational data	Определение оперативных данных триггера, содержащее произвольные строки и макросы, которые будут динамически разрешаться в разделе *Мониторинг → Проблемы*.
Expression	Отображается выражение триггера. Часть выражения шаблон-элемент данных отображается как ссылка, ведущая к форме настройки элемента данных.
Status	Отображается состояние триггера — *Включен* или *Отключен*. Щелчком по состоянию его можно изменить — с Включен на Отключен (и обратно).
Tags	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:

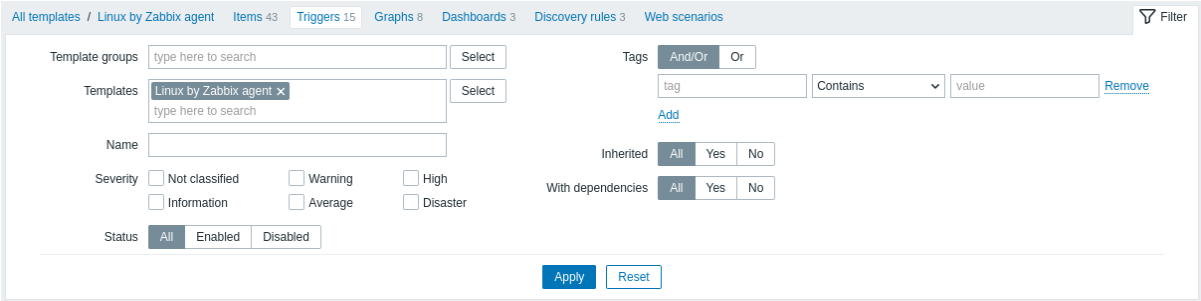
- *Enable* — изменить статус триггера на *Enabled*
- *Disable* — изменить статус триггера на *Disabled*
- *Copy* — скопировать триггеры на другие узлы сети или шаблоны
- *Mass update* — обновить несколько свойств для нескольких триггеров одновременно
- *Delete* — удалить триггеры

Чтобы использовать эти параметры, установите флажки напротив соответствующих триггеров, 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 неразрешёнными макросами.

Значок *Фильтр* доступен в правом верхнем углу. Щелчок по нему откроет фильтр, в котором можно указать нужные критерии фильтрации.



Параметр	Описание
Группы шаблонов	Фильтрация по одной или нескольким группам шаблонов. Указание родительской группы шаблонов неявно выбирает 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.
Важность	Выберите, чтобы фильтровать по одной или нескольким важностям триггера.
Состояние	Фильтрация по состоянию активации триггера (Включено/Отключено).

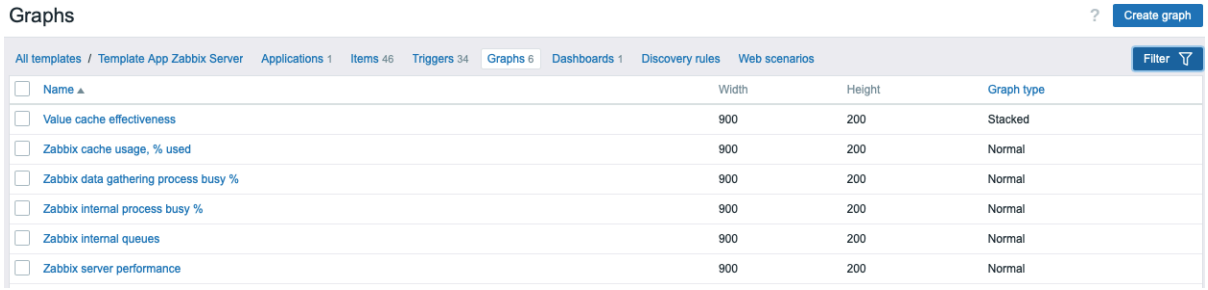
Параметр	Описание
Теги	Фильтрация по имени и значению тега триггера. Можно как включать, так и исключать определённые теги и значения тегов. Можно задать несколько условий. Сопоставление имени тега всегда чувствительно к регистру. Для каждого условия доступны несколько операторов: **Существует** — включить указанные имена тегов **Равно** — включить указанные имена и значения тегов (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 и значения тегов (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, условия с одинаковым именем тега будут сгруппированы по условию Или **Или** — достаточно выполнения одного условия В полях имени тега и значения тега поддерживаются макросы и функции макросов.
Унаследованные	Фильтрация триггеров, унаследованных (или не унаследованных) из связанных шаблонов.
С зависимостями	Фильтрация триггеров с зависимостями (или без зависимостей).

3 Графики

Обзор

К списку пользовательских графиков для шаблона можно перейти из *Сбор данных* → *Шаблоны*, щелкнув *Графики* у соответствующего шаблона.

Отображается список существующих графиков.



Отображаемые данные:

Столбец	Описание
Шаблон	Шаблон, которому принадлежит график. Щелчок по имени шаблона открывает форму 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 как синяя ссылка на сведения о графике. Щелчок по ссылке с именем графика открывает форму настройки графика. Если график унаследован от другого шаблона, перед именем графика отображается имя шаблона в виде серой ссылки. Щелчок по ссылке шаблона открывает список графиков на уровне этого шаблона.
Ширина	Отображается ширина графика.
Высота	Отображается высота графика.
Тип графика	Отображается тип графика — *Обычный*, *С накоплением*, *Круговой* или *Разделённый*.

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, установите флажки напротив соответствующих графиков, затем нажмите нужную кнопку.

Использование фильтра

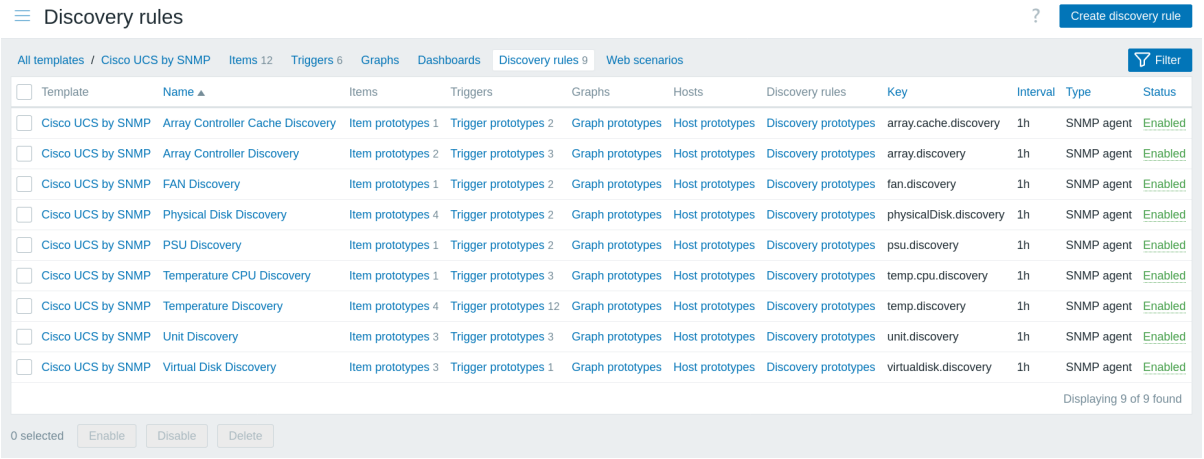
Вы можете фильтровать графики по группе шаблонов и шаблону. Для повышения производительности поиска данные ищутся с неразрешёнными макросами.

4 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 определённой группы шаблонов, изменив настройки фильтра.



Отображаемые данные:

Столбец	Описание
Шаблон	Шаблон, к которому относится правило обнаружения. Нажатие на имя шаблона открывает форму настройки шаблона.
Имя	Имя правила, отображаемое как синяя ссылка. Нажатие на имя правила открывает форму настройки правила низкоуровневого обнаружения. Если правило обнаружения унаследовано от другого шаблона, имя шаблона отображается перед именем правила как серая ссылка. Нажатие на ссылку шаблона открывает 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 частота выполнения обнаружения.
Тип	Отображается тип элемента данных, используемого для обнаружения (агент Zabbix, SNMP-агент и т. д.).
Состояние	Отображается состояние правила обнаружения — *Включено* или *Отключено*. Нажав на состояние, вы можете изменить его — с *Включено* на *Отключено* (и обратно).

Чтобы настроить новое правило низкоуровневого обнаружения, нажмите кнопку *Создать правило обнаружения* в правом верхнем углу.

Параметры массового редактирования

Кнопки под списком предоставляют несколько вариантов массового редактирования:

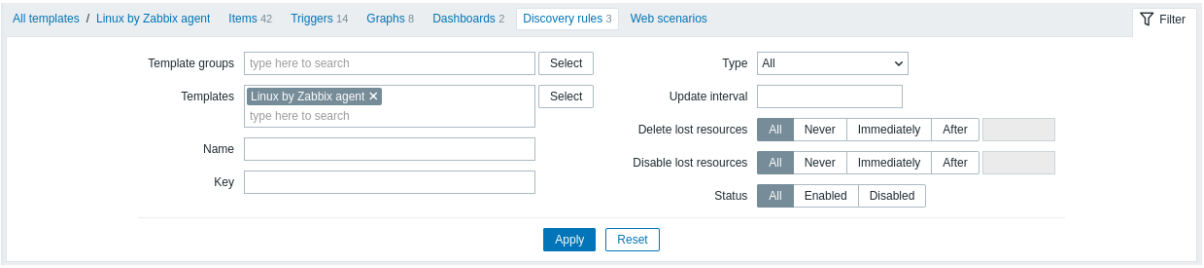
- *Enable* — изменить состояние правила обнаружения низкого уровня на *Enabled*
- *Disable* — изменить состояние правила обнаружения низкого уровня на *Disabled*
- *Delete* — удалить правила обнаружения низкого уровня

Чтобы использовать эти параметры, установите флажки напротив соответствующих правил 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 неразрешёнными макросами.

Значок *Фильтр* доступен в правом верхнем углу. Щелчок по нему откроет фильтр, где вы можете указать нужные критерии фильтрации, такие как шаблон, имя правила обнаружения, ключ элемента данных, тип элемента данных и т. д.



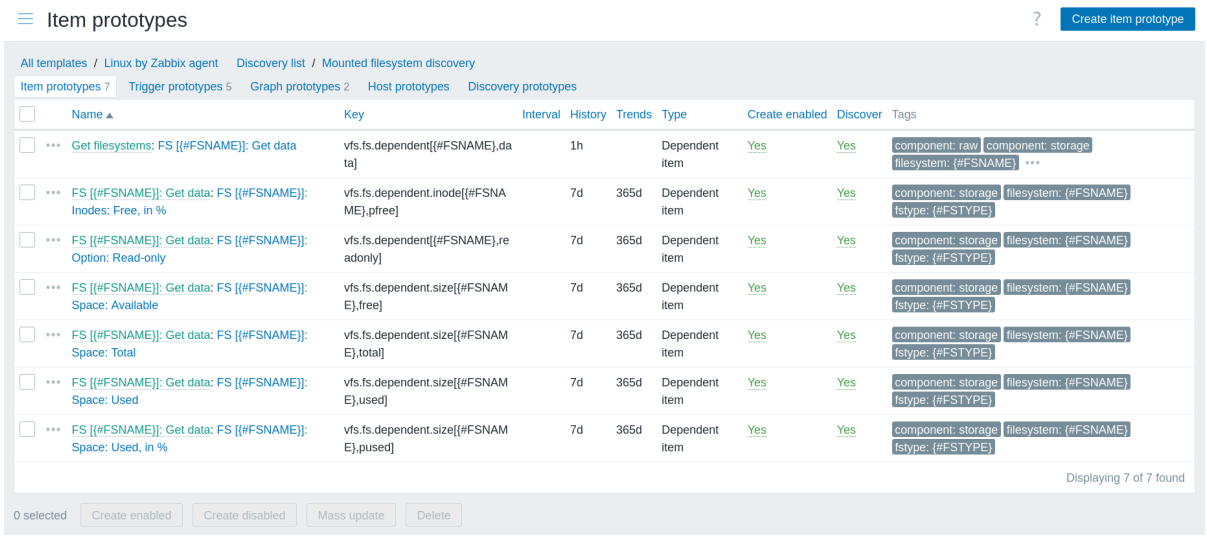
Параметр	Описание
Группы шаблонов	Фильтрация по одной или нескольким группам шаблонов. Указание родительской группы шаблонов неявно выбирает все вложенные группы.
Шаблоны	Фильтрация по одному или нескольким шаблонам.
Имя	Фильтрация по имени правила обнаружения.
Ключ	Фильтрация по ключу элемента данных обнаружения.
Тип	Фильтрация по типу элемента данных обнаружения.
Интервал обновления	Фильтрация по интервалу обновления. Недоступно для Zabbix trapper и зависимых элементов данных.
Удалять потерянные ресурсы	Фильтрация по периоду *Удалять потерянные ресурсы*.
Отключать потерянные ресурсы	Фильтрация по периоду *Отключать потерянные ресурсы*.
Состояние	Фильтрация по состоянию правила обнаружения (Все/Включено/Отключено).

1 Прототипы элементов данных

Обзор

В этом разделе отображаются настроенные прототипы элементов данных правила низкоуровневого обнаружения в шаблоне.

Если шаблон связан с узлом сети, прототипы элементов данных станут основой для создания реальных элементов данных узла сети в ходе низкоуровневого обнаружения.



Отображаемые данные:

Столбец	Описание
Имя	Отображается имя прототипа элемента данных в виде синей ссылки. Нажатие на имя открывает форму настройки прототипа элемента данных. Если прототип элемента данных принадлежит связанному шаблону, перед именем прототипа элемента данных отображается имя шаблона в виде серой ссылки. Нажатие на ссылку шаблона открывает список прототипов элементов данных на уровне связанного шаблона. Если прототип элемента данных имеет тип ”Зависимый элемент данных”, перед именем прототипа элемента данных зелёным цветом отображается имя мастер-элемента данных (как и в случае с прототипами элементов данных на уровне узла сети).
Ключ	Отображается ключ прототипа элемента данных.
Интервал	Отображается частота проверки.
История	Отображается, сколько дней хранить историю данных элемента данных.
Тренды	Отображается, сколько дней хранить историю трендов элемента данных.
Тип	Отображается тип прототипа элемента данных (агент Zabbix, SNMP-агент, простая проверка и т. д.).
Создавать включённым	Создавать элемент данных на основе этого прототипа как: **Да** — включённый **Нет** — отключённый. Вы можете переключаться между значениями ”Да” и ”Нет”, нажимая на них.
Обнаруживать	Обнаруживать элемент данных на основе этого прототипа: **Да** — обнаруживать **Нет** — не обнаруживать. Вы можете переключаться между значениями ”Да” и ”Нет”, 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 предоставляют несколько вариантов массового редактирования:

- *Create enabled* — создать эти элементы данных как *Enabled*
- *Create disabled* — создать эти элементы данных как *Disabled*
- *Mass update* — массово обновить эти прототипы элементов данных
- *Delete* — удалить эти прототипы элементов данных

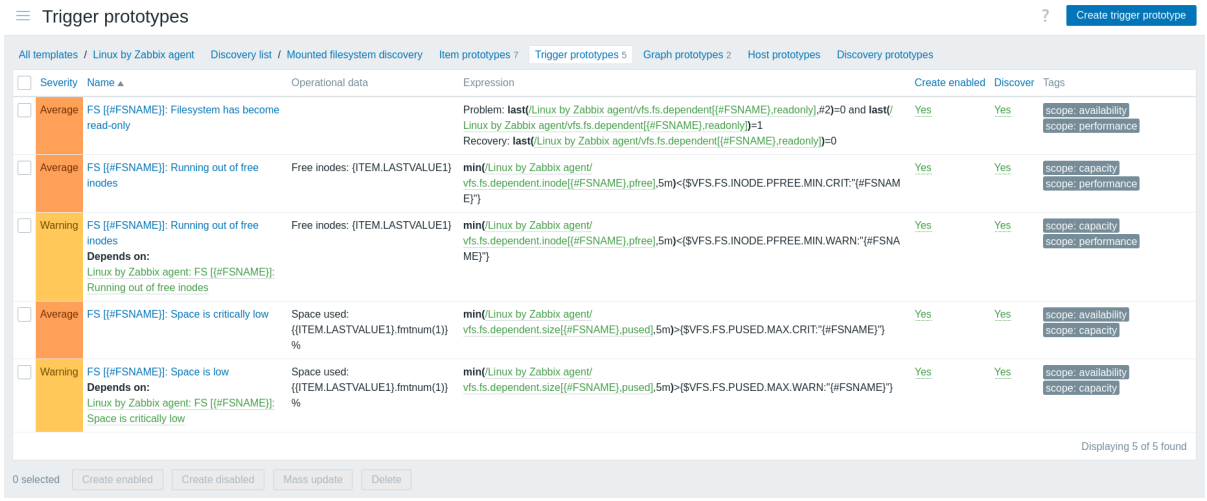
Чтобы использовать эти параметры, отметьте флажки напротив соответствующих прототипов элементов данных, затем нажмите нужную кнопку.

2 Прототипы триггеров

Обзор

В этом разделе отображаются настроенные прототипы триггеров правила 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	Важность триггера отображается как по названию, так и по цвету фона ячейки.
Name	Имя прототипа триггера отображается в виде синей ссылки. Нажатие на имя открывает форму настройки прототипа триггера. Если прототип триггера принадлежит связанному шаблону, имя шаблона отображается перед именем триггера в виде серой ссылки. Нажатие на ссылку шаблона открывает список прототипов триггеров на уровне связанного шаблона.
Expression	Отображается выражение триггера. Часть выражения шаблон-элемент данных отображается как ссылка, ведущая к форме настройки элемента данных.
Operational data	Отображается формат эксплуатационных данных триггера, содержащий произвольные строки и макросы, которые будут динамически разрешаться в *Monitoring* → *Problems*.
Create enabled	Создавать триггер на основе этого прототипа как: **Yes** - включённый **No** - отключённый. Вы можете переключаться между ’Yes’ и ’No’, нажимая на них.
Discover	Обнаруживать триггер на основе этого прототипа: **Yes** - обнаруживать **No** - не обнаруживать. Вы можете переключаться между ’Yes’ и ’No’, 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 предоставляют несколько вариантов массового редактирования:

- *Create enabled* — создать эти триггеры как *Включено*
- *Create disabled* — создать эти триггеры как *Отключено*
- *Mass update* — массово обновить эти прототипы триггеров
- *Delete* — удалить эти прототипы триггеров

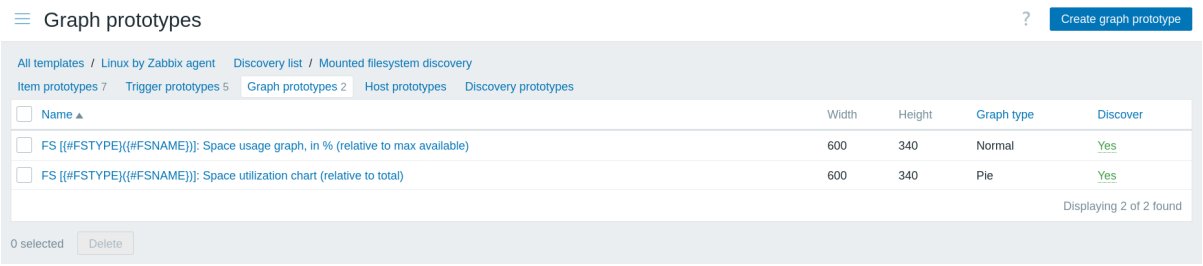
Чтобы использовать эти параметры, установите флажки напротив соответствующих прототипов триггеров, затем нажмите нужную кнопку.

3 Прототипы графиков

Обзор

В этом разделе отображаются настроенные прототипы графиков правила низкоуровневого обнаружения в шаблоне.

Если шаблон связан с узлом сети, прототипы графиков станут основой для создания реальных графиков узла сети во время низкоуровневого обнаружения.



Отображаемые данные:

Column	Description
Name	Отображается имя прототипа графика в виде синей ссылки. Щелчок по имени открывает форму настройки прототипа графика. Если прототип графика принадлежит связанному шаблону, перед именем графика отображается имя шаблона в виде серой ссылки. Щелчок по ссылке шаблона открывает 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. Вы можете переключаться между 'Yes' и 'No', щелкая по ним.

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

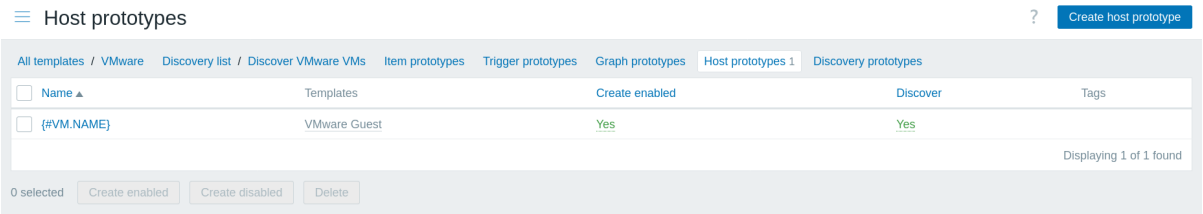
Чтобы использовать эти параметры, установите флажки напротив соответствующих прототипов графиков, затем нажмите требуемую кнопку.

4 Прототипы узлов сети

Обзор

В этом разделе отображаются настроенные прототипы узлов сети правила низкоуровневого обнаружения в шаблоне.

Если шаблон связан с узлом сети, прототипы узлов сети станут основой для создания реальных узлов сети в ходе низкоуровневого обнаружения.



Отображаемые данные:

Столбец	Описание
Имя	Имя прототипа узла сети, отображаемое как синяя ссылка. Щелчок по имени открывает форму настройки прототипа узла сети. Если прототип узла сети принадлежит связанному шаблону, имя шаблона отображается перед именем узла сети как серая ссылка. Щелчок по ссылке шаблона открывает список прототипов узлов сети на уровне связанного шаблона.
Шаблоны	Отображаются шаблоны прототипа узла сети.

Столбец	Описание
Создавать включённым	Создавать узел сети на основе этого прототипа в состоянии: **Да** — включено **Нет** — отключено. Вы можете переключаться между значениями «Да» и «Нет», щёлкая по ним.
Обнаруживать	Обнаруживать узел сети на основе этого прототипа: **Да** — обнаруживать **Нет** — не обнаруживать. Вы можете переключаться между значениями «Да» и «Нет», щёлкая по ним.
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 предоставляют несколько вариантов массового редактирования:

- *Create enabled* — создать эти узлы сети как *Enabled*
- *Create disabled* — создать эти узлы сети как *Disabled*
- *Delete* — удалить эти прототипы узлов сети

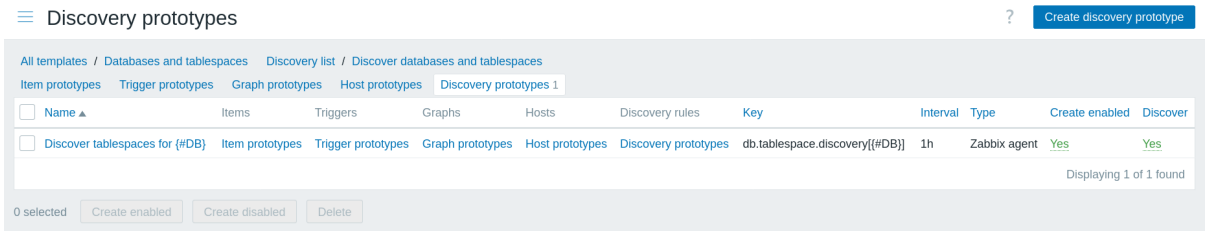
Чтобы использовать эти параметры, установите флажки напротив соответствующих прототипов узлов сети, затем нажмите нужную кнопку.

5 Прототипы обнаружения

Обзор

В этом разделе отображаются настроенные прототипы обнаружения правила низкоуровневого обнаружения в шаблоне.

Если шаблон связан с узлом сети, прототипы обнаружения станут основой для создания реальных прототипов обнаружения родительского правила обнаружения.



Отображаемые данные:

Column	Description
Name	Имя прототипа обнаружения, отображаемое как синяя ссылка. Щелчок по имени открывает форму настройки прототипа обнаружения. Если прототип обнаружения принадлежит связанному шаблону, имя шаблона отображается перед именем прототипа обнаружения как серая ссылка. Щелчок по ссылке шаблона открывает 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.
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, используемого для обнаружения (агент Zabbix, SNMP агент и т. д.).

Column	Description
Create enabled	Отображается состояние создания прототипа обнаружения — создание включено (*Yes*) или создание отключено (*No*). Щелчком по состоянию его можно изменить — с Yes на No (и обратно).
Discover	Отображается состояние обнаружения прототипа обнаружения — обнаруживать (*Yes*) или не обнаруживать (*No*). Щелчком по состоянию 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 предоставляют несколько вариантов массового редактирования:

- *Create enabled* — создать эти прототипы обнаружения в состоянии *Включено*
- *Create disabled* — создать эти прототипы обнаружения в состоянии *Отключено*
- *Delete* — удалить эти прототипы обнаружения

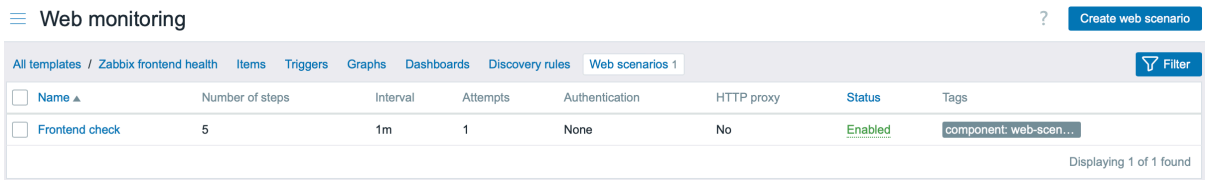
Чтобы использовать эти параметры, установите флажки напротив соответствующих прототипов обнаружения, затем нажмите нужную кнопку.

5 Веб-сценарии

Обзор

К списку веб-сценариев для шаблона можно перейти из *Сбор данных* → *Шаблоны*, щёлкнув по *Веб* для соответствующего шаблона.

Отображается список существующих веб-сценариев.



Отображаемые данные:

Column	Description
Name	Имя веб-сценария. Щелчок по имени веб-сценария открывает форму настройки веб-сценария. Если веб-сценарий унаследован от другого шаблона, перед именем веб-сценария отображается имя шаблона в виде серой ссылки. Щелчок по ссылке шаблона открывает список веб-сценариев на уровне этого шаблона.
Number of steps	Количество шагов, содержащихся в сценарии.
Update interval	Как часто выполняется сценарий.
Attempts	Сколько попыток выполняется для выполнения шагов веб-сценария.
Authentication	Отображается метод аутентификации — Basic, NTLM, Kerberos, Digest или None.
HTTP proxy	Отображает HTTP-прокси или 'No', если он не используется.
Status	Отображается состояние веб-сценария — *Включено* или *Отключено*. Щелчком по состоянию его можно изменить.
Tags	Отображаются теги веб-сценария. Может отображаться до трёх тегов (пары имя:значение). Если тегов больше, отображается ссылка "..." , позволяющая увидеть все теги при наведении указателя мыши.

Чтобы настроить новый веб-сценарий, нажмите кнопку *Создать веб-сценарий* в правом верхнем углу.

Параметры массового редактирования

Кнопки под списком предоставляют несколько вариантов массового редактирования:

- *Enable* — изменить статус сценария на *Enabled*

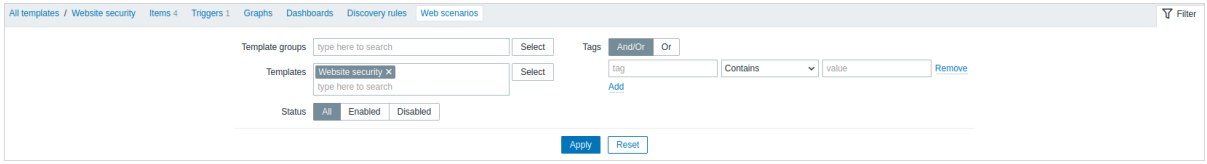
- *Disable* — изменить статус сценария на *Disabled*
- *Delete* — удалить веб-сценарии

Чтобы использовать эти параметры, установите флажки напротив соответствующих веб-сценариев, 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 неразрешёнными макросами.

Ссылка *Фильтр* доступна над списком веб-сценариев. Если щёлкнуть по ней, станет доступен фильтр, в котором можно отфильтровать сценарии по группе шаблонов, шаблону, состоянию и тегам.

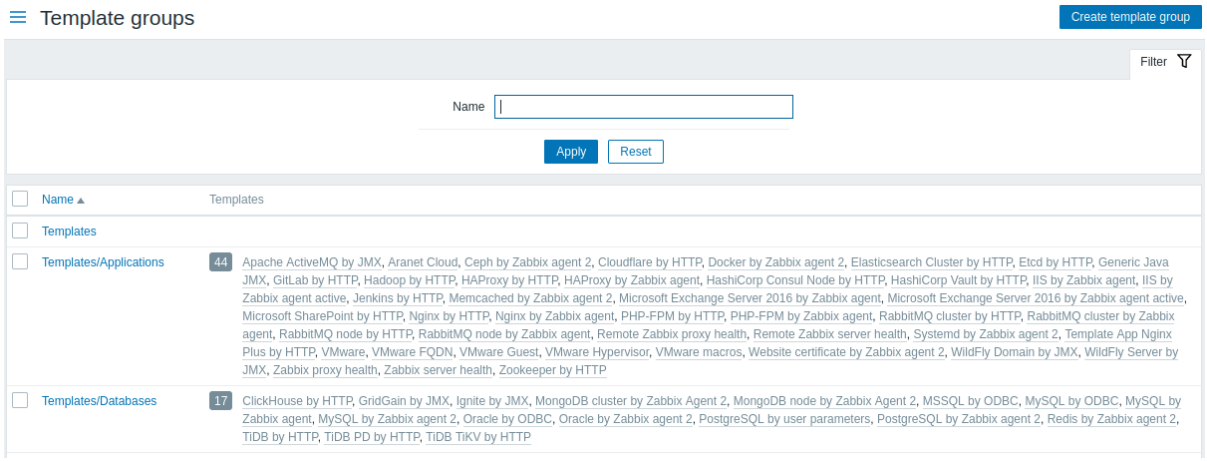


1 Группы шаблонов

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 фильтрацию групп шаблонов по имени.



Отображаемые данные:

Столбец	Описание
Имя	Имя группы шаблонов. Щелчок по имени группы открывает форму настройки группы.
Шаблоны	Количество шаблонов в группе (отображается серым цветом), после которого следует список участников группы. Щелчок по имени шаблона открывает форму настройки шаблона. Щелчок по числу открывает список шаблонов в этой группе.

Параметры массового редактирования

Чтобы удалить сразу несколько групп шаблонов, установите флажки напротив соответствующих групп, затем нажмите кнопку Delete под списком.

Использование фильтра

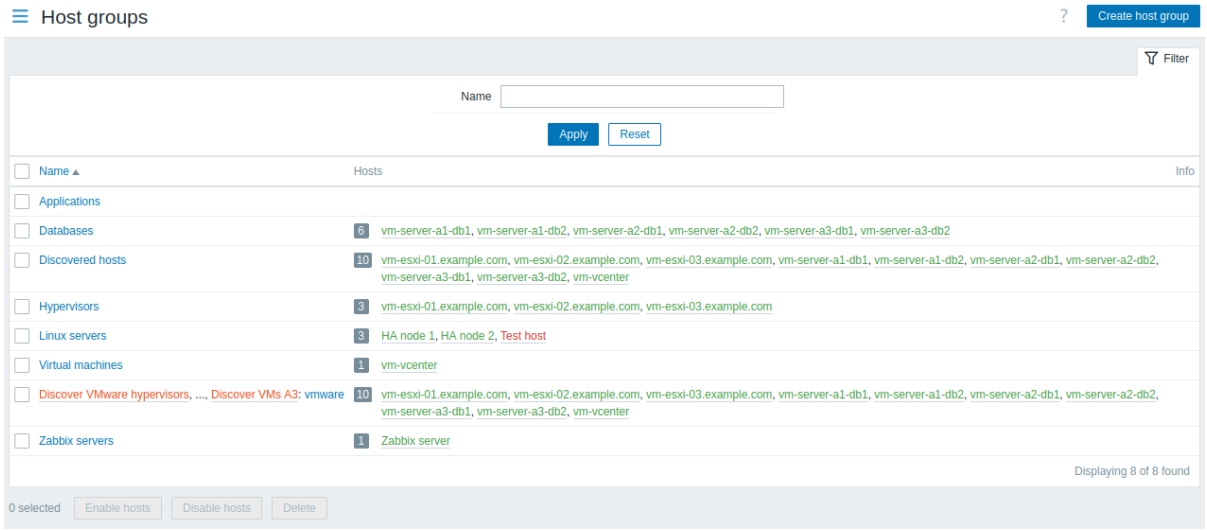
Вы можете использовать фильтр, чтобы отображать только те группы шаблонов, которые вас интересуют. Для повышения производительности поиска данные ищутся с неразрешёнными макросами.

2 Группы узлов сети

Обзор

В разделе *Сбор данных* → *Группы узлов сети* пользователи могут настраивать и поддерживать 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:

Column	Description
Name	Имя группы узлов сети. Щелчок по имени группы открывает форму настройки группы. Обнаруженные группы узлов сети отображаются с именами правил обнаружения низкого уровня в качестве префикса. Щелчок по имени правила LLD открывает форму настройки прототипа узла сети. Обратите внимание, что обнаруженные группы узлов сети удаляются, когда удаляются все правила LLD, которые их обнаружили.
Hosts	Количество узлов сети в группе (отображается серым цветом), за которым следует список участников группы. Щелчок по имени узла сети открывает форму настройки узла сети. Щелчок по числу выполнит фильтрацию в полном списке узлов сети, оставив только те, которые принадлежат группе.
Info	Отображается информация об ошибках (если есть), относящихся к группе узлов сети.

Параметры массового редактирования

Кнопки под списком предоставляют несколько вариантов массового редактирования:

- *Включить узлы сети* — изменить статус всех узлов сети в группе на «Наблюдается»
- *Отключить узлы сети* — изменить статус всех узлов сети в группе на «Не наблюдается»
- *Удалить* — удалить группы узлов сети

Чтобы использовать эти параметры, установите флажки напротив соответствующих групп узлов сети, затем нажмите нужную кнопку.

Использование фильтра

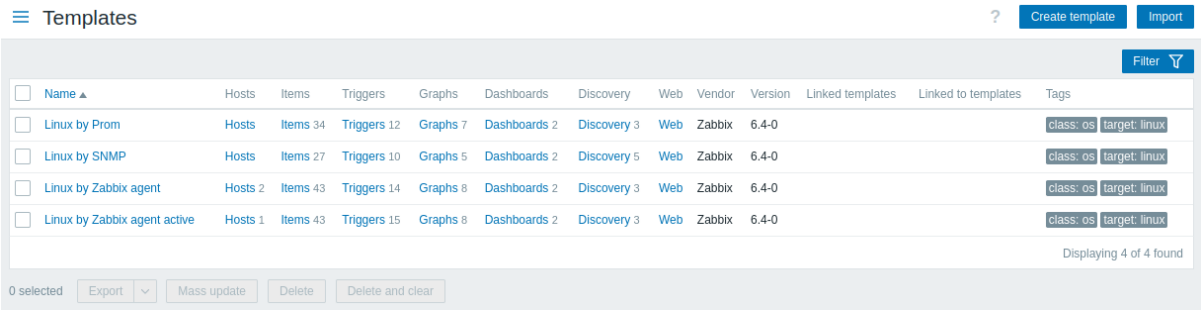
Вы можете использовать фильтр, чтобы отображать только те группы узлов сети, которые вас интересуют. Для повышения производительности поиска данные ищутся с неразрешёнными макросами.

3 Шаблоны

Обзор

В разделе *Сбор данных* → *Шаблоны* пользователи могут настраивать и поддерживать шаблоны.

Отображается список существующих шаблонов с подробной информацией о них.



Отображаемые данные:

Столбец	Описание
Имя	Имя шаблона. Щелчок по имени шаблона открывает форму настройки шаблона.
Узлы сети	Количество редактируемых узлов сети, с которыми связан шаблон; узлы сети только для чтения не включаются. Щелчок по *Узлам сети* откроет список узлов сети, отфильтрованный так, чтобы отображались только те узлы сети, которые связаны с шаблоном.
Сущности (Элементы данных, Триггеры, Графики, Панели, Обнаружение, Веб)	Количество соответствующих сущностей в шаблоне (отображается серым цветом). Щелчок по имени сущности отфильтрует в полном списке этой сущности те, которые принадлежат шаблону.
Связанные шаблоны	Шаблоны, которые связаны с данным шаблоном.
Связан с шаблонами	Шаблоны, с которыми данный шаблон связан.
Поставщик, *Версия*	Поставщик и версия шаблона; отображаются, если конфигурация шаблона содержит такую информацию, и только для готовых шаблонов, импортированных шаблонов или шаблонов, изменённых через Template API. Для готовых шаблонов версия отображается следующим образом: основная версия Zabbix, разделитель (”-”), номер редакции (увеличивается с каждой новой версией шаблона и сбрасывается с каждой новой основной версией Zabbix). Например, 7.0-0, 7.0-3, 8.0-0, 8.0-3.
Теги	Теги шаблона, с неразрешёнными макросами.

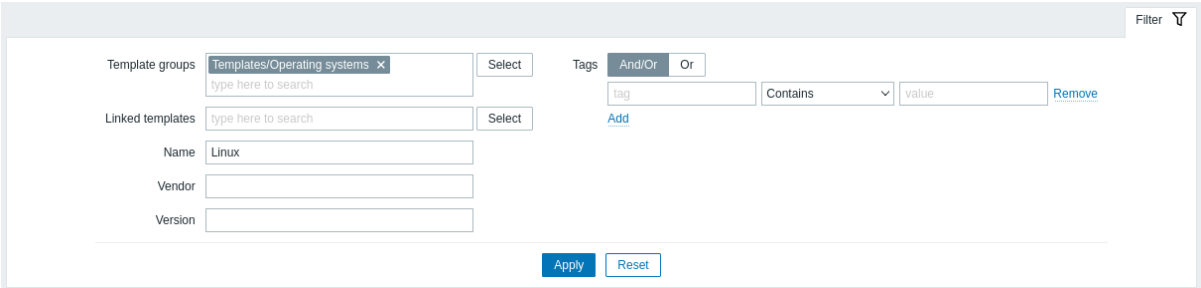
Чтобы настроить новый шаблон, нажмите кнопку *Создать шаблон* в правом верхнем углу.

Чтобы импортировать шаблон из файла YAML, XML или JSON, нажмите кнопку *Импорт* в правом верхнем углу.

Использование фильтра

Вы можете использовать фильтр, чтобы отображать только те шаблоны, которые вас интересуют. Для повышения производительности поиска данные ищутся с неразрешёнными макросами.

Ссылка *Фильтр* доступна под кнопками *Создать шаблон* и *Импорт*. Если вы нажмёте на неё, станет доступен фильтр, в котором можно фильтровать шаблоны по группе шаблонов, связанным шаблонам, имени и тегам.



Параметр	Описание
Группы шаблонов	Фильтрация по одной или нескольким группам шаблонов. Указание родительской группы шаблонов неявно выбирает все вложенные группы.
Связанные шаблоны	Фильтрация по напрямую связанным шаблонам.
Имя	Фильтрация по имени шаблона.
Поставщик	Фильтрация по поставщику шаблона.
Версия	Фильтрация по версии шаблона.

Параметр	Описание
Теги	Фильтрация по имени и значению тега шаблона. Фильтрация возможна только по тегам уровня шаблона (не унаследованным). 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 доступно несколько операторов: **Существует** — включить указанные имена тегов; **Равно** — включить указанные имена и значения тегов (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 и значения тегов (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.

Параметры массового редактирования

Кнопки под списком предоставляют несколько вариантов массового редактирования:

- *Экспорт* — экспортировать шаблон в файл YAML, XML или JSON;
- *Массовое обновление* — обновить несколько свойств сразу для нескольких шаблонов;
- *Удалить* — удалить шаблон, оставив его связанные сущности (элементы данных, триггеры и т. д.) на узлах сети;
- *Удалить и очистить* — удалить шаблон и его связанные сущности с узлов сети.

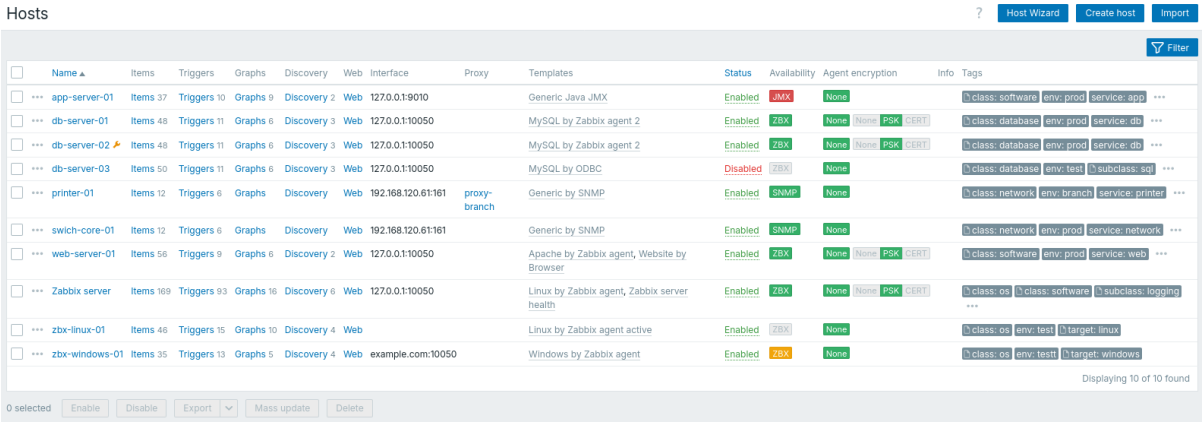
Чтобы использовать эти параметры, установите флажки напротив соответствующих шаблонов, затем нажмите нужную кнопку.

4 Узлы сети

Обзор

В разделе *Сбор данных* → *Узлы сети* пользователи могут настраивать и поддерживать узлы сети.

Отображается список существующих узлов сети с их подробной информацией.



Отображаемые данные:

Столбец	Описание
Меню узла сети	Щелкните по значку с тремя точками, чтобы открыть меню узла сети.

Столбец	Описание
Имя	Имя узла сети. Щелчок по имени узла сети открывает его форму настройки. Оранжевый значок гаечного ключа после имени узла сети указывает, что этот узел сети находится на обслуживании. Подробности обслуживания отображаются при наведении указателя мыши на значок.
Сущности (Элементы данных, Триггеры, Графики, Обнаружение, Веб)	Щелчок по имени сущности отобразит 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: <Proxy name> - узел сети мониторится автономным прокси (даже если прокси является частью группы прокси); <Proxy group name: proxy name> - узел сети мониторится группой прокси, и сервер Zabbix назначил прокси для мониторинга узла сети; <Proxy group name> - узел сети мониторится группой прокси без каких-либо прокси, либо если сервер Zabbix не назначил прокси для мониторинга узла сети; Ничего - узел сети не мониторится ни прокси, ни группой прокси. Этот столбец отображается только если параметр фильтра *Мониторится* установлен в 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. Щелчок по имени шаблона откроет его форму настройки.
Состояние	Отображается состояние узла сети — *Включен* или *Отключен*. Щелчком по состоянию его можно изменить вручную. Обнаруженные узлы сети, которые были потеряны, помечаются информационным значком. Текст всплывающей подсказки содержит сведения об их состоянии.

Столбец	Описание
Доступность	Отображается доступность узла сети для каждого настроенного интерфейса. Значки доступности отражают текущее состояние интерфейса узла сети на сервере Zabbix. Поэтому, если вы отключите узел сети в 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 на значок отображается всплывающее окно со списком всех интерфейсов того же типа с подробностями, состоянием и ошибками. Для интерфейса Agent во всплывающем окне отображаются интерфейсы (пассивные) и активные проверки. Если у узла сети есть только активные проверки, значок интерфейса Agent отображается, даже если у узла сети не настроен интерфейс Agent. Для узлов сети без интерфейсов столбец пуст. 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 было успешным; **Недоступен** - соединение с узлом сети было неуспешным (тайм-аут, проблемы с межсетевым экраном и т. д.); **Неизвестно** - попытка соединения с узлом сети не выполнялась или результат неизвестен. Дополнительные сведения о том, как сервер Zabbix определяет состояние интерфейса, см. в разделах Неизвестное состояние интерфейса и Настройки недостижимого/недоступного интерфейса узла сети. Состояние всех интерфейсов узла сети одного типа (Agent, SNMP, IPMI, JMX) определяется по тем интерфейсам, которые используются хотя бы одним включенным элементом данных. Состояние обозначается цветом значка: **Зеленый** - все интерфейсы доступны; **Желтый** - как минимум один интерфейс недоступен, и как минимум один доступен или имеет состояние «неизвестно»; **Красный** - все интерфейсы недоступны; **Серый** - как минимум один интерфейс имеет состояние «неизвестно», 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 в потоке активных проверок агента отправляются heartbeat-сообщения. Частота heartbeat-сообщений управляется параметром `HeartbeatFrequency` в конфигурации Zabbix агент или агент 2 (по умолчанию 60 секунд, диапазон 0-3600). Активные проверки считаются недоступными, если heartbeat активной проверки старше 2 x HeartbeatFrequency секунд. **Примечание:** агенты Zabbix старше версии 6.2.x не отправляют heartbeat активных проверок, поэтому доступность их узлов сети остается неизвестной.
Шифрование агента	Отображается состояние шифрования для соединений к узлу сети и от узла сети: **Нет** - без шифрования; **PSK** - используется предварительно разделяемый ключ; **Cert** - используется сертификат.
Информация	Отображается информация об ошибках (если есть), относящихся к узлу сети.
Теги	Теги узла сети с неразрешенными макросами.

Чтобы настроить новый узел сети, нажмите кнопку *Host Wizard* или *Create host* в правом верхнем углу. Чтобы импортировать узел сети из файла YAML, XML или JSON, нажмите кнопку *Import* в правом верхнем углу.

Параметры массового редактирования

Кнопки под списком предоставляют несколько параметров массового редактирования:

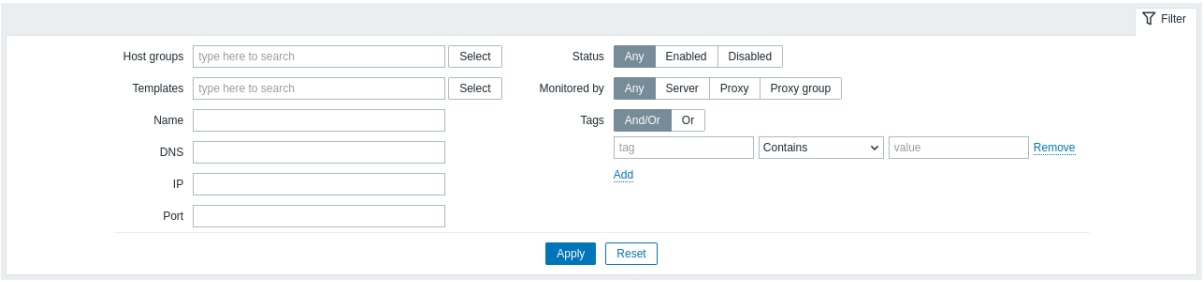
- *Включить* — изменить состояние узла сети на *Наблюдаемый*;
- *Отключить* — изменить состояние узла сети на *Не наблюдаемый*;
- *Экспорт* — экспортировать узлы сети в файл YAML, XML или JSON;
- *Массовое обновление* — обновить несколько свойств сразу для нескольких узлов сети;
- *Удалить* — удалить узлы сети.

Чтобы использовать эти параметры, установите флажки напротив соответствующих узлов сети, затем нажмите нужную кнопку.

Использование фильтра

Вы можете использовать фильтр, чтобы отображать только те узлы сети, которые вас интересуют. Для повышения производительности поиска данные ищутся с неразрешёнными макросами.

Значок *Фильтр* доступен в правом верхнем углу. При нажатии на него откроется фильтр, в котором можно указать нужные критерии фильтрации.



Параметр	Описание
Группы узлов сети	Фильтрация по одной или нескольким группам узлов сети. Указание родительской группы узлов сети неявно выбирает все вложенные группы узлов сети.
Шаблоны	Фильтрация по связанным шаблонам.
Имя	Фильтрация по видимому имени узла сети.
DNS	Фильтрация по имени DNS.
IP	Фильтрация по IP-адресу.
Порт	Фильтрация по номеру порта.
Состояние	Фильтрация по состоянию узла сети.
Кем мониторится	Фильтрация узлов сети, которые мониторятся сервером Zabbix, прокси или группой прокси.
Прокси	Фильтрация узлов сети, которые мониторятся указанными здесь прокси. Это поле доступно только если в поле *Кем мониторится* выбрано ”Прокси”.
Группы прокси	Фильтрация узлов сети, которые мониторятся указанными здесь группами прокси. Это поле доступно только если в поле *Кем мониторится* выбрано ”Группа прокси”.
Теги	Фильтрация по имени и значению тега узла сети. 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 и значения тегов (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 и значения тегов (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.

Неизвестный статус интерфейса

Сервер Zabbix отображает статус ”Unknown” для интерфейса узла сети (агент, SNMP, IPMI, JMX) в следующих случаях:

- Узел сети отключен.
- Для узла сети задан мониторинг через прокси, другой прокси или сервер, если ранее он мониторился через прокси.
- Узел сети мониторится прокси, который, по-видимому, недоступен (от прокси не поступали обновления в течение максимального интервала heartbeat — 1 час).
- Все элементы данных узла сети с этим типом интерфейса отключены.
- Для этого типа интерфейса не настроены poller-процессы (например, параметр конфигурации сервера `StartAgentPollers` или `StartSNMPPollers` установлен в 0).

Доступность интерфейса переводится в состояние ”Unknown” после синхронизации кэша конфигурации сервера Zabbix.

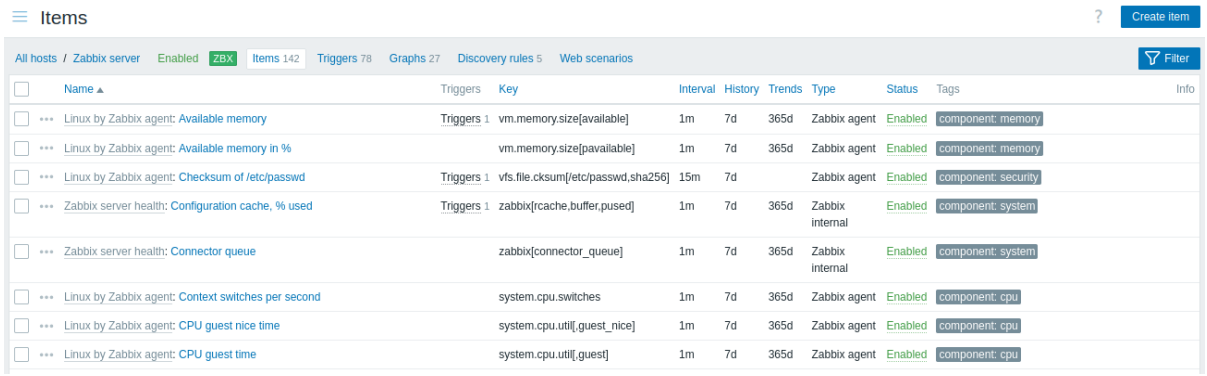
Доступность интерфейса (Available/Not available) на узлах сети, мониторинг которых выполняется прокси, восстанавливается после синхронизации кэша конфигурации прокси.

1 Элементы данных

Обзор

К списку элементов данных для узла сети можно перейти из *Сбор данных* → *Узлы сети*, нажав *Элементы данных* для соответствующего узла сети.

Отображается список существующих элементов данных.



Отображаемые данные:

Column	Description
Item menu	Нажмите на значок с тремя точками, чтобы открыть меню элемента данных.
Host	Узел сети элемента данных. Нажатие на имя узла сети открывает форму настройки узла сети. Этот столбец отображается только в том случае, если в фильтре выбрано несколько узлов сети или не выбран ни один.
Name	Имя элемента данных отображается как синяя ссылка на сведения об элементе данных. Нажатие на ссылку с именем элемента данных открывает форму настройки элемента данных. Если элемент данных узла сети принадлежит шаблону, имя шаблона отображается перед именем элемента данных в виде серой ссылки. Нажатие на ссылку шаблона откроет список элементов данных на уровне шаблона. Если элемент данных был создан из прототипа элемента данных, перед его именем оранжевым цветом отображается имя правила обнаружения низкого уровня. Нажатие на имя правила обнаружения откроет список прототипов элементов данных.
Triggers	При наведении указателя мыши на Triggers отображается информационный блок со связанными с элементом данных триггерами. Количество триггеров отображается серым цветом.
Key	Отображается ключ элемента данных.
Interval	Отображается частота проверки. Обратите внимание, что пассивные элементы данных также могут быть проверены немедленно нажатием кнопки *Выполнить сейчас* button.
History	Отображается, сколько дней будет храниться история данных элемента данных.
Trends	Отображается, сколько дней будет храниться история трендов элемента данных.
Type	Отображается тип элемента данных (агент Zabbix, SNMP-агент, простая проверка и т. д.).

Column	Description
Status	Отображается статус элемента данных — *Включен*, *Отключен* или *Не поддерживается*. Вы можете изменить статус вручную, нажав на него: с Включен на Отключен (и обратно); с Не поддерживается на Отключен (и обратно). Обнаруженные элементы данных, которые были потеряны, помечаются информационным значком. Текст всплывающей подсказки содержит подробности об их статусе.
Tags	Отображаются теги элемента данных. Может отображаться до трех тегов (пары имя:значение). Если тегов больше, отображается ссылка "..." , позволяющая увидеть все теги при наведении указателя мыши.
Info	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:

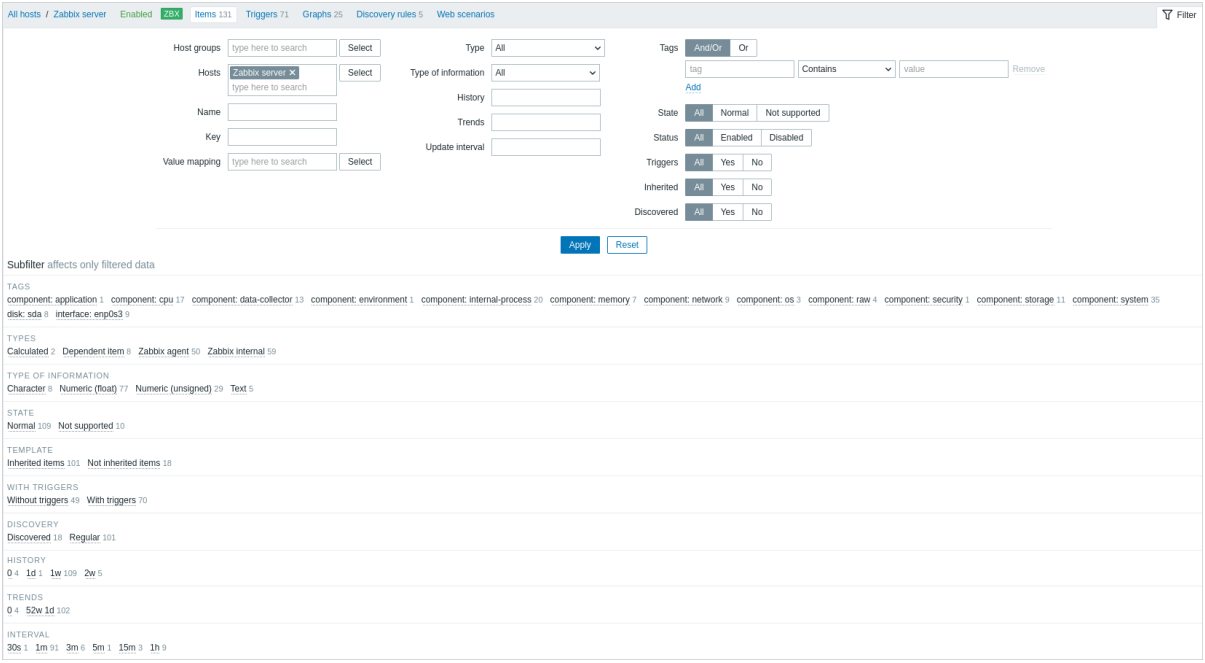
- *Включить* — изменить статус элемента данных на *Включен*
- *Отключить* — изменить статус элемента данных на *Отключен*
- *Выполнить сейчас* — немедленно выполнить проверку для получения новых значений элемента данных. Поддерживается только для **пассивных** проверок (см. подробнее). Обратите внимание, что при немедленной проверке значений кэш конфигурации не обновляется, поэтому значения не будут отражать самые последние изменения в конфигурации элемента данных.
- *Очистить историю и тренды* — удалить данные истории и трендов для элементов данных.
- *Копировать* — скопировать элементы данных на другие узлы сети или шаблоны.
- *Массовое обновление* — обновить несколько свойств сразу для нескольких элементов данных.
- *Удалить* — 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 те элементы данных, которые вас интересуют. Для повышения производительности поиска данные ищутся с неразрешёнными макросами.

Значок *Фильтр* доступен в правом верхнем углу. Щелчок по нему откроет фильтр, в котором можно указать нужные критерии фильтрации.

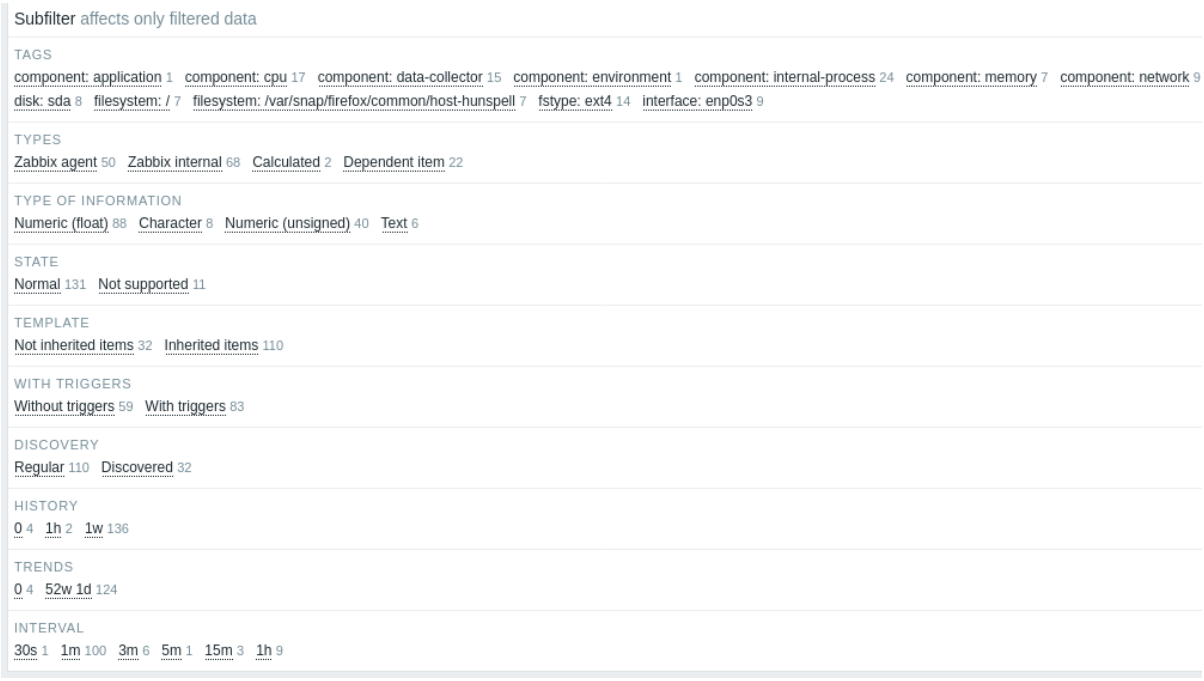


Параметр	Описание
Группы узлов сети	Фильтрация по одной или нескольким группам узлов сети. При указании родительской группы узлов сети неявно выбираются все вложенные группы узлов сети. Группы узлов сети, содержащие только шаблоны, выбрать нельзя.
Узлы сети	Фильтрация по одному или нескольким узлам сети.
Имя	Фильтрация по имени элемента данных.
Ключ	Фильтрация по ключу элемента данных.
Преобразование значений	Фильтрация по используемому преобразованию значений. Этот параметр не отображается, если параметр *Узлы сети* пуст.
Тип	Фильтрация по типу элемента данных (агент Zabbix, SNMP-агент и т. д.).
Тип информации	Фильтрация по типу информации (беззнаковое числовое, число с плавающей точкой и т. д.).
История	Фильтрация по сроку хранения истории элемента данных.
Тренды	Фильтрация по сроку хранения трендов элемента данных.
Интервал обновления	Фильтрация по интервалу обновления элемента данных.
Теги	Укажите теги, чтобы ограничить количество отображаемых элементов данных. 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 доступно несколько операторов: **Существует** — включить указанные имена тегов **Равно** — включить указанные имена и значения тегов (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 и значения тегов (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
Состояние	Фильтрация по состоянию элемента данных — *Нормальное* или *Не поддерживается*.
Статус	Фильтрация по статусу элемента данных — *Включено* или *Отключено*.
Триггеры	Фильтрация элементов данных с триггерами (или без них).
Унаследовано	Фильтрация элементов данных, унаследованных (или не унаследованных) из шаблона.
Обнаружение	Фильтрация элементов данных, обнаруженных (или не обнаруженных) с помощью low-level discovery.

Использование подфильтра

Подфильтр позволяет дополнительно изменять фильтрацию из основного фильтра.

Он содержит кликабельные ссылки 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 нажатии на сущность она выделяется серым фоном, а элементы данных немедленно фильтруются (нажимать *Применить* в основном фильтре не нужно). Нажатие на другую сущность добавляет её к отфильтрованным результатам. Повторное нажатие на сущность убирает фильтрацию.



Подфильтры формируются на основе отфильтрованных данных, которые ограничены 1000 записями. Если вы хотите видеть больше записей в подфильтре, необходимо увеличить значение параметра *Limit for search and filter results* (в *Administration -> General -> GUI*).

В отличие от основного фильтра, подфильтр обновляется при каждом запросе на обновление таблицы, чтобы всегда содержать актуальную информацию о доступных вариантах фильтрации и количестве элементов данных в них.

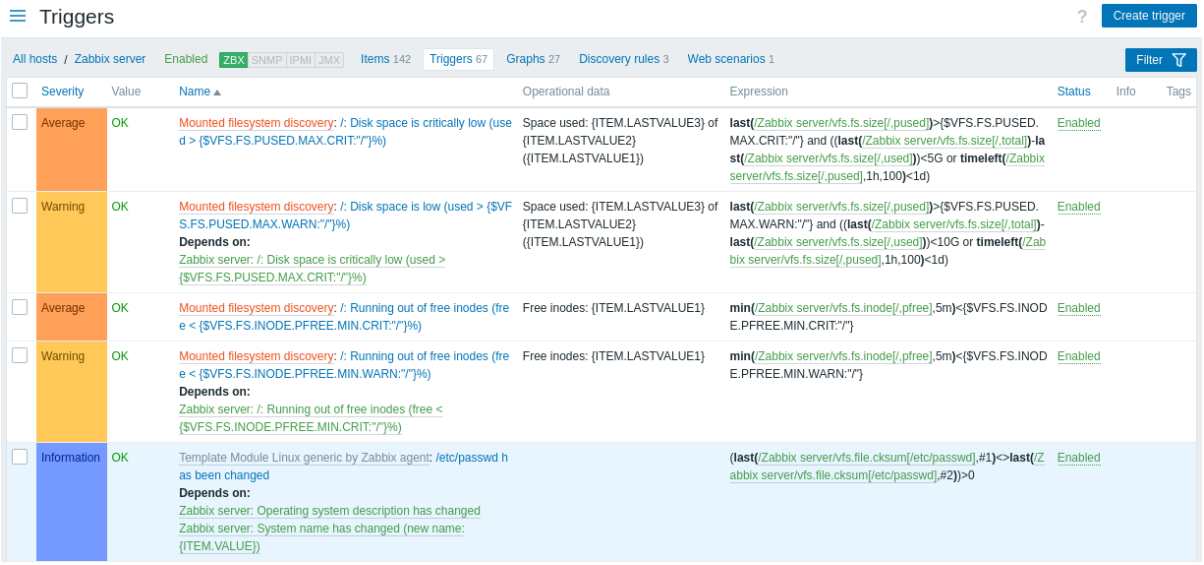
Количество отображаемых сущностей по горизонтали ограничено 100. Если их больше, в конце отображается значок с тремя точками; он не является кликабельным.

Число рядом с каждой кликабельной сущностью показывает количество элементов данных, сгруппированных в ней (на основе результатов основного фильтра). При нажатии на сущность числа у других доступных сущностей отображаются со знаком «плюс», указывая, сколько элементов данных может быть добавлено к текущему выбору. Сущности без элементов данных не отображаются, если только они не были ранее выбраны в подфильтре.

2 Триггеры

Обзор

Список триггеров для узла сети можно открыть из *Сбор данных* → *Узлы сети*, нажав на *Триггеры* для соответствующего узла сети.



Отображаемые данные:

Column	Description
Severity	Важность триггера отображается как по названию, так и по цвету фона ячейки.
Value	Отображается состояние триггера: **OK** - состояние OK **PROBLEM** - состояние проблемы
Host	Узел сети триггера. Нажатие на имя узла сети открывает форму настройки узла сети. Этот столбец отображается только если в фильтре выбрано несколько узлов сети или не выбран ни один узел сети.
Name	Имя триггера отображается как синяя ссылка на сведения о триггере. Нажатие на ссылку с именем триггера открывает форму настройки триггера. Если триггер узла сети принадлежит шаблону, имя шаблона отображается перед именем триггера в виде серой ссылки. Нажатие на ссылку шаблона откроет список триггеров на уровне шаблона. Если триггер был создан из прототипа триггера, перед его именем оранжевым цветом отображается имя правила низкоуровневого обнаружения. Нажатие на имя правила обнаружения открывает список прототипов триггеров.
Operational data	Определение оперативных данных триггера, содержащее произвольные строки и макросы, которые будут динамически разрешаться в *Мониторинг → Проблемы*.
Expression	Отображается выражение триггера. Часть выражения host-item отображается как ссылка, ведущая к форме настройки элемента данных.
Status	Отображается статус триггера — *Включен*, *Отключен* или *Неизвестно*. Нажатием на статус можно вручную изменить его — 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 информационным значком. Текст всплывающей подсказки содержит подробности об их статусе.
Info	Если всё работает корректно, в этом столбце значок не отображается. В случае ошибок отображается квадратный значок с буквой ”i”. Наведите курсор на значок, чтобы увидеть всплывающую подсказку с описанием ошибки.
Tags	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 предоставляют несколько вариантов массового редактирования:

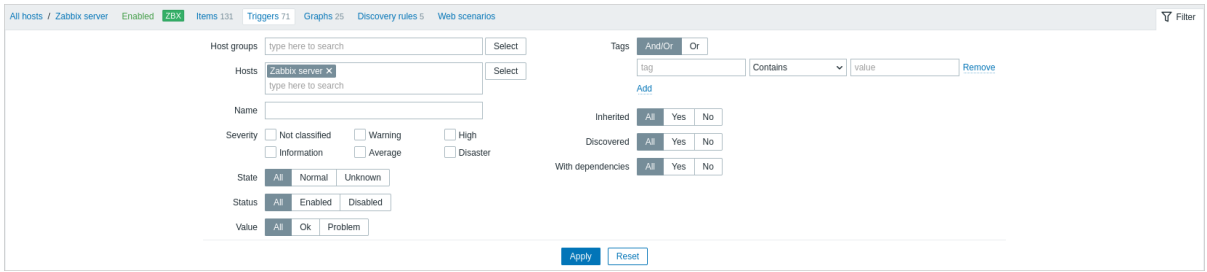
- *Включить* — изменить статус триггера на *Включен*.
- *Отключить* — изменить статус триггера на *Отключен*.
- *Копировать* — скопировать триггеры на другие узлы сети или шаблоны.
- *Массовое обновление* — обновить сразу несколько свойств для нескольких триггеров.
- *Удалить* — удалить триггеры.

Чтобы использовать эти параметры, установите флажки напротив соответствующих триггеров, 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 неразрешёнными макросами.

Значок *Фильтр* доступен в правом верхнем углу. Щелчок по нему откроет фильтр, где можно указать нужные критерии фильтрации.



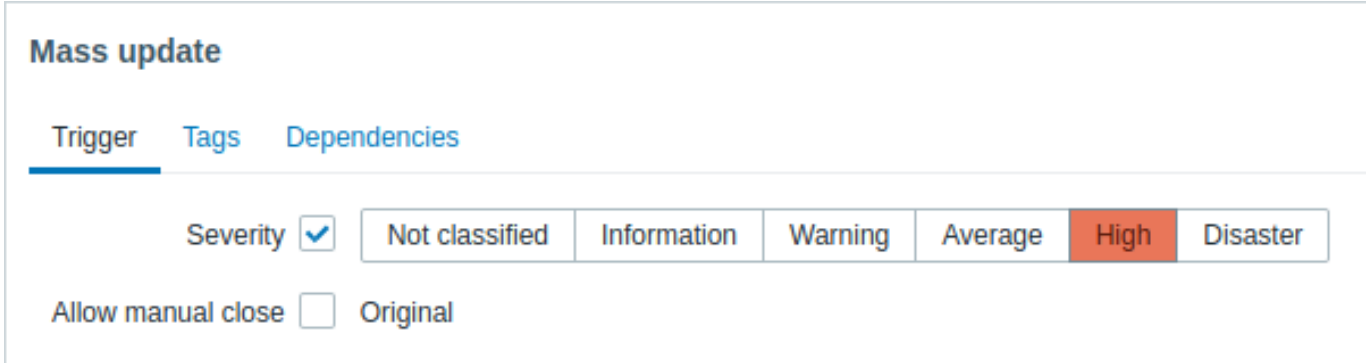
Параметр	Описание
Группы узлов сети	Фильтрация по одной или нескольким группам узлов сети. При указании родительской группы узлов сети автоматически выбираются все вложенные группы узлов сети. Группы узлов сети, содержащие только шаблоны, выбрать нельзя.
Узлы сети	Фильтрация по одному или нескольким узлам сети. Если выше уже выбраны группы узлов сети, выбор узлов сети ограничивается этими группами.
Имя	Фильтрация по имени триггера.
Важность	Выберите, чтобы фильтровать по одной или нескольким важностям триггера.
Состояние	Фильтрация по состоянию выражения триггера (Нормальное/Неизвестное).
Статус	Фильтрация по статусу активации триггера (Включён/Отключён).
Значение	Фильтрация по значению триггера.
Теги	Фильтрация по имени и значению тега триггера. 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 доступно несколько операторов: **Существует** — включить указанные имена тегов **Равно** — включить указанные имена и значения тегов (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 и значения тегов (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 выполняться все условия, при этом условия с одинаковым именем тега будут сгруппированы по условию Или **Или** — достаточно выполнения одного условия Макросы и функции макросов поддерживаются как в поле имени тега, так и в поле значения тега.
Унаследованные	Фильтрация триггеров, унаследованных (или не унаследованных) от шаблона.
Обнаруженные	Фильтрация триггеров, обнаруженных (или не обнаруженных) с помощью low-level discovery.
С зависимостями	Фильтрация триггеров с зависимостями (или без зависимостей).

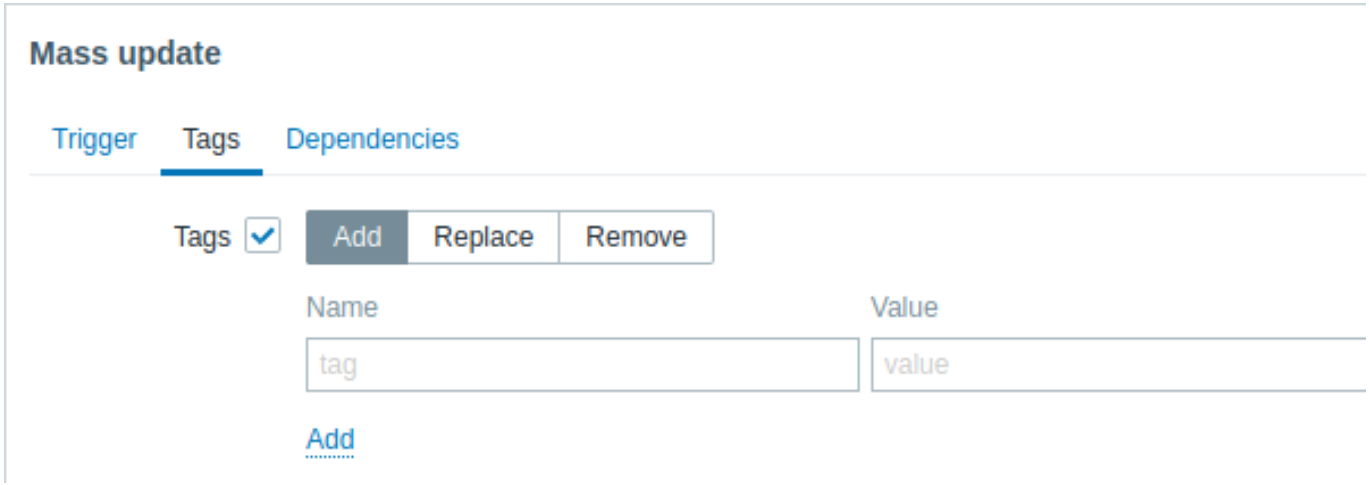
Использование массового обновления

С помощью массового обновления вы можете изменить некоторые атрибуты сразу для нескольких триггеров, что избавляет от необходимости открывать каждый отдельный триггер для редактирования.

Чтобы массово обновить несколько триггеров, выполните следующие действия:

- Установите флажки у триггеров, которые вы хотите обновить, в списке
- Нажмите *Mass update* под списком
- Перейдите на вкладку с необходимыми атрибутами (*Trigger*, *Tags* или *Dependencies*)
- Установите флажки у тех атрибутов, которые нужно обновить

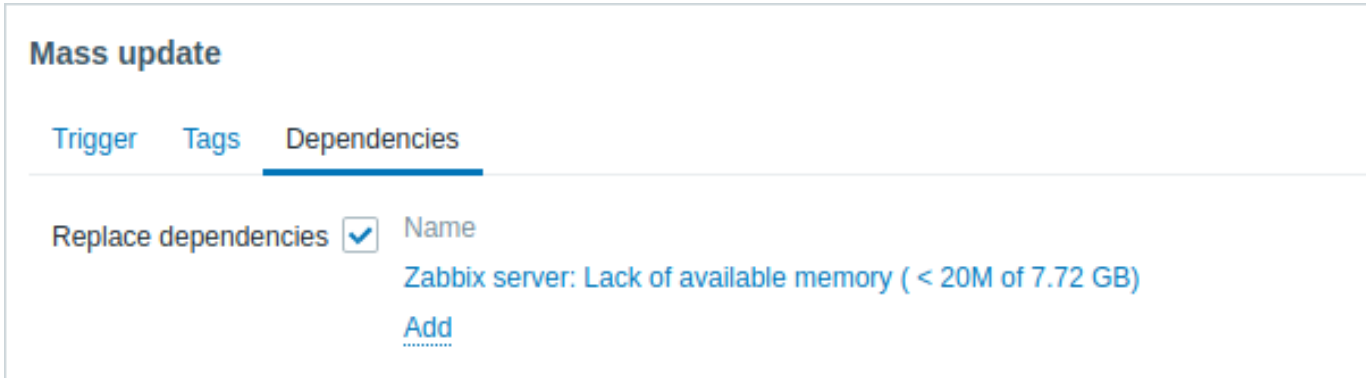




При выборе соответствующей кнопки для обновления тегов доступны следующие параметры:

- *Add* — позволяет добавить новые теги для триггеров;
- *Replace* — удалит все существующие теги из триггера и заменит их указанными ниже;
- *Remove* — удалит 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.



Replace dependencies — удалит все существующие зависимости из триггера и заменит их указанными.

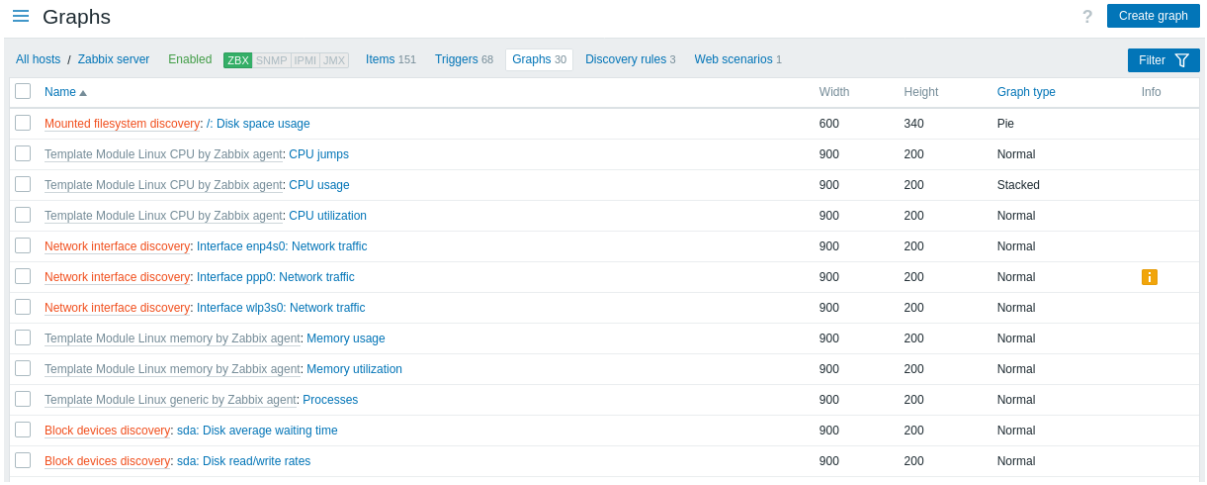
Нажмите *Update*, чтобы применить изменения.

3 Графики

Обзор

К списку пользовательских графиков для узла сети можно перейти из *Сбор данных* → *Узлы сети*, нажав *Графики* для соответствующего узла сети.

Отображается список существующих графиков.



Отображаемые данные:

Column	Description
Host	Узел сети графика. Нажатие на имя узла сети открывает форму настройки узла сети. Этот столбец отображается только в том случае, если в фильтре выбрано несколько узлов сети или не выбран ни один узел сети.
Name	Имя пользовательского графика отображается как синяя ссылка на сведения о графике. Нажатие на ссылку с именем графика открывает форму настройки графика. Если график узла сети принадлежит шаблону, имя шаблона отображается перед именем графика в виде серой ссылки. Нажатие на ссылку шаблона откроет список графиков на уровне шаблона. Если график был создан из прототипа графика, перед его именем оранжевым цветом отображается имя правила обнаружения низкого уровня. Нажатие на имя правила обнаружения открывает список прототипов графиков.
Width	Отображается ширина графика.
Height	Отображается высота графика.
Graph type	Отображается тип графика — *Обычный*, *Накопительный*, *Круговой* или *Разделённый*.
Info	Если график работает корректно, в этом столбце значок не отображается. В случае ошибок отображается квадратный значок с буквой "i". Наведите курсор 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 предоставляют несколько вариантов массового редактирования:

- *Копировать* — копировать графики на другие узлы сети или шаблоны
- *Удалить* — удалить графики

Чтобы использовать эти параметры, установите флажки напротив соответствующих графиков, затем нажмите нужную кнопку.

Использование фильтра

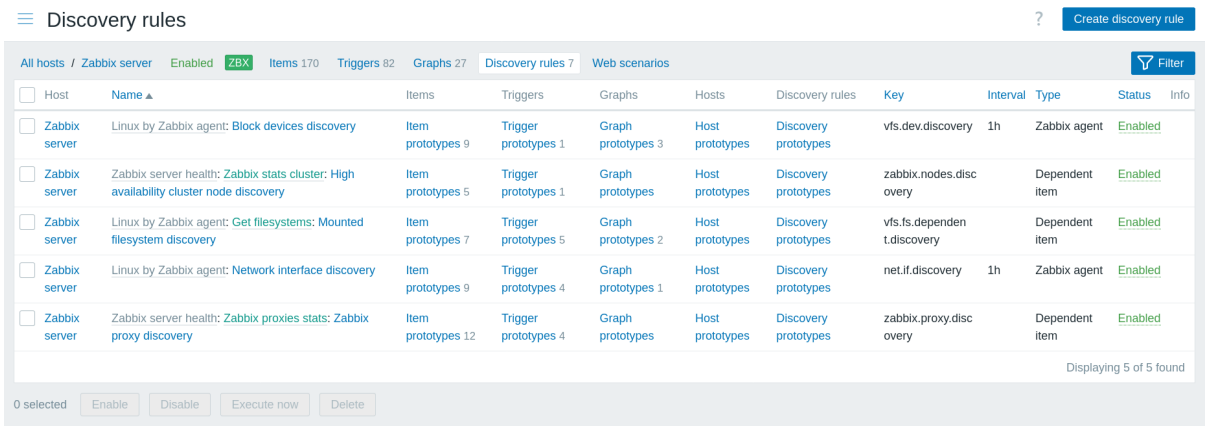
Вы можете фильтровать графики по группе узлов сети и узлу сети. Для повышения производительности поиска данные ищутся с неразрешёнными макросами.

4 Правила обнаружения

Обзор

К списку правил низкоуровневого обнаружения для узла сети можно перейти в *Сбор данных* → *Узлы сети*, щелкнув по *Обнаружение* для соответствующего узла сети.

Отображается список существующих правил низкоуровневого обнаружения. Также можно просматривать все правила обнаружения независимо от узла сети или все правила обнаружения определенной группы узлов сети, изменив настройки фильтра.



Отображаемые данные:

Column	Description
Host	Отображается видимое имя узла сети. Щелчок по имени узла сети открывает форму настройки узла сети. Если видимое имя узла сети отсутствует, отображается техническое имя узла сети.
Name	Имя правила отображается в виде синей ссылки. Щелчок по имени правила открывает форму настройки правила низкоуровневого обнаружения. Если правило обнаружения принадлежит шаблону, перед именем правила отображается имя шаблона в виде серой ссылки. Щелчок по ссылке шаблона открывает список правил на уровне 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.
Key	Отображается ключ элемента данных, используемый для обнаружения.
Interval	Отображается частота выполнения обнаружения. Обратите внимание, что обнаружение также можно выполнить немедленно, нажав кнопку *Выполнить сейчас* под списком.
Type	Отображается тип элемента данных, используемый для обнаружения (агент Zabbix, SNMP-агент и т. д.).
Status	Отображается состояние правила обнаружения — *Включено*, *Отключено* или *Не поддерживается*. Щелчком по состоянию его можно изменить: с Включено на Отключено (и обратно); с Не поддерживается на Отключено (и обратно).
Info	Если все в порядке, в этом столбце значок не отображается. В случае ошибок отображается квадратный значок с буквой "i". Наведите курсор на значок, чтобы увидеть всплывающую подсказку с описанием ошибки.

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:

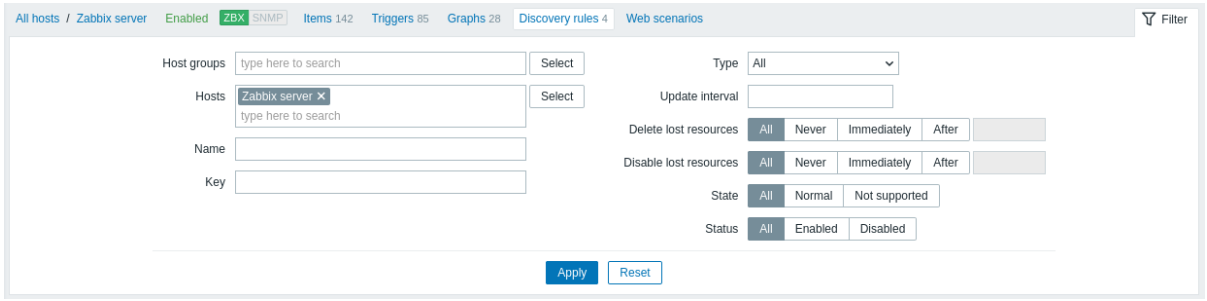
- *Enable* — изменить состояние правила обнаружения низкого уровня на *Enabled*.
- *Disable* — изменить состояние правила обнаружения низкого уровня на *Disabled*.
- *Execute now* — немедленно выполнить обнаружение на основе правил обнаружения. См. подробнее. Обратите внимание, что при немедленном выполнении обнаружения кэш конфигурации не обновляется, поэтому результат не будет отражать самые последние изменения в конфигурации правила обнаружения.
- *Delete* — удалить правила обнаружения низкого уровня.

Чтобы использовать эти параметры, установите флажки напротив соответствующих правил 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 неразрешёнными макросами.

Ссылка *Фильтр* доступна над списком правил обнаружения. Если нажать на неё, станет доступен фильтр, в котором можно фильтровать правила обнаружения по группе узлов сети, узлу сети, имени, ключу элемента данных, типу элемента данных и другим параметрам.

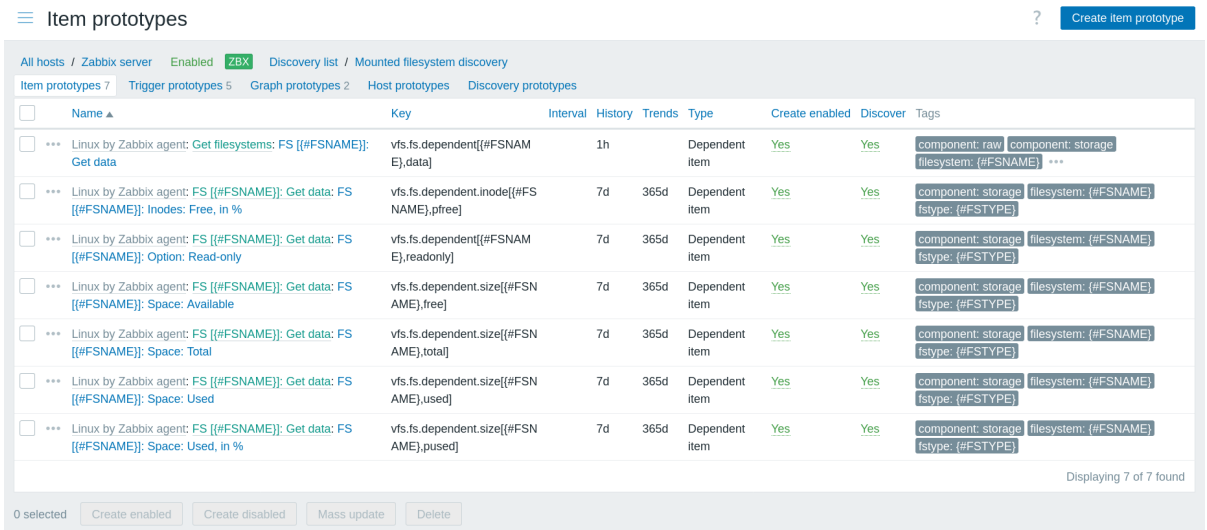


Параметр	Описание
Группы узлов сети	Фильтрация по одной или нескольким группам узлов сети. При указании родительской группы узлов сети автоматически выбираются все вложенные группы узлов сети.
Узлы сети	Фильтрация по одному или нескольким узлам сети.
Имя	Фильтрация по имени правила обнаружения.
Ключ	Фильтрация по ключу элемента данных обнаружения.
Тип	Фильтрация по типу элемента данных обнаружения.
Интервал обновления	Фильтрация по интервалу обновления. Недоступно для Zabbix trapper и зависимых элементов данных.
Удалять потерянные ресурсы	Фильтрация по периоду *Удалять потерянные ресурсы*.
Отключать потерянные ресурсы	Фильтрация по периоду *Отключать потерянные ресурсы*.
SNMP OID	Фильтрация по SNMP OID. Доступно только если в качестве типа выбран *SNMP agent*.
Состояние	Фильтрация по состоянию правила обнаружения (Все/Нормальное/Не поддерживается).
Статус	Фильтрация по статусу правила обнаружения (Все/Включено/Отключено).

1 Прототипы элементов данных

Обзор

В этом разделе отображаются прототипы элементов данных правила низкоуровневого обнаружения на узле сети. Прототипы элементов данных являются основой реальных элементов данных узла сети, которые создаются в процессе низкоуровневого обнаружения.



Отображаемые данные:

Столбец	Описание
Имя	Отображается имя прототипа элемента данных в виде синей ссылки. Щелчок по имени открывает форму настройки прототипа элемента данных. Если прототип элемента данных принадлежит шаблону, имя шаблона отображается перед именем правила в виде серой ссылки. Щелчок по ссылке шаблона открывает список прототипов элементов данных на уровне шаблона. Если прототип элемента данных имеет тип ”Зависимый элемент данных”, имя мастер-элемента данных отображается перед именем прототипа элемента данных зеленым цветом.
Ключ	Отображается ключ прототипа элемента данных.
Интервал	Отображается частота проверки.
История	Отображается, сколько дней хранить историю данных элемента данных.
Тренды	Отображается, сколько дней хранить историю трендов элемента данных.
Тип	Отображается тип прототипа элемента данных (агент Zabbix, SNMP агент, простая проверка и т.д.).
Создавать включенным	Создавать элемент данных на основе этого прототипа как: **Да** - включенный **Нет** - отключенный. Вы можете переключаться между ’Да’ и ’Нет’, щелкая по ним.
Обнаруживать	Обнаруживать элемент данных на основе этого прототипа: **Да** - обнаруживать **Нет** - не обнаруживать. Вы можете переключаться между ’Да’ и ’Нет’, щелкая по ним.
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:

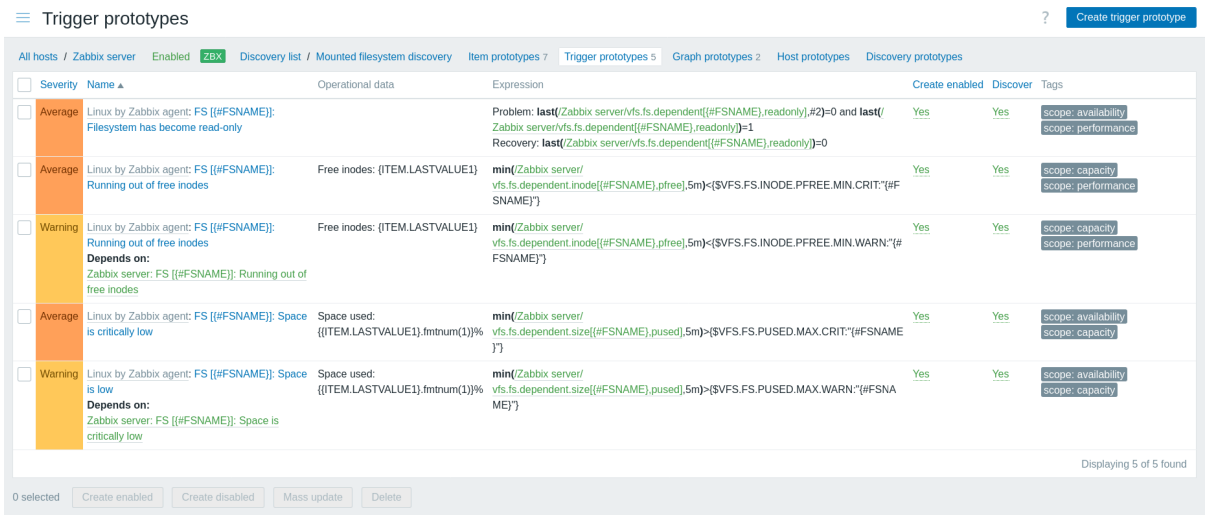
- *Создать включёнными* — создать эти элементы данных как *Включено*
- *Создать отключёнными* — создать эти элементы данных как *Отключено*
- *Массовое обновление* — массово обновить эти прототипы элементов данных
- *Удалить* — удалить эти прототипы элементов данных

Чтобы использовать эти параметры, отметьте флажки напротив соответствующих прототипов элементов данных, затем нажмите нужную кнопку.

2 Прототипы триггеров

Обзор

В этом разделе отображаются прототипы триггеров правила низкоуровневого обнаружения на узле сети. Прототипы триггеров являются основой реальных триггеров узла сети, которые создаются в ходе низкоуровневого обнаружения.



Отображаемые данные:

Столбец	Описание
Имя	Имя прототипа триггера отображается в виде синей ссылки. Щелчок по имени открывает форму настройки прототипа триггера. Если прототип триггера принадлежит связанному шаблону, имя шаблона отображается перед именем триггера в виде серой ссылки. Щелчок по ссылке шаблона открывает список прототипов триггеров на уровне связанного шаблона.
Оперативные данные	Отображается формат оперативных данных триггера, содержащий произвольные строки и макросы, которые будут динамически разрешаться в *Мониторинг → Проблемы*.
Создавать включённым	Создавать триггер на основе этого прототипа как: **Да** - включённый **Нет** - отключённый. Вы можете переключаться между значениями «Да» и «Нет», щёлкая по ним.
Обнаруживать	Обнаруживать триггер на основе этого прототипа: **Да** - обнаруживать **Нет** - не обнаруживать. Вы можете переключаться между значениями «Да» и «Нет», щёлкая по ним.
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 предоставляют несколько вариантов массового редактирования:

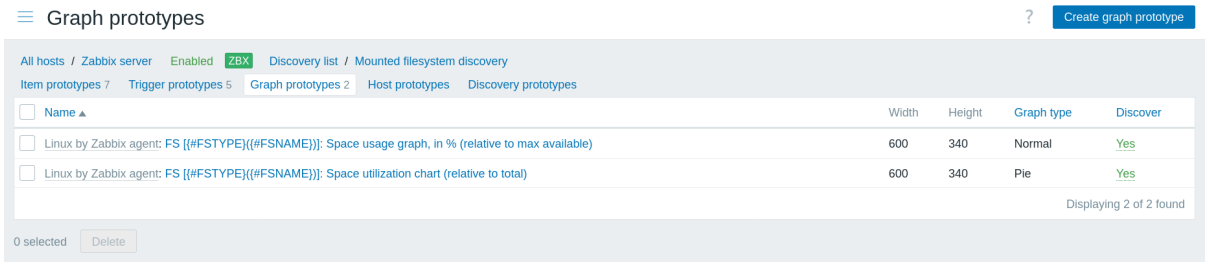
- *Create enabled* — создать эти триггеры как *Enabled*
- *Create disabled* — создать эти триггеры как *Disabled*
- *Mass update* — массово обновить эти прототипы триггеров
- *Delete* — удалить эти прототипы триггеров

Чтобы использовать эти параметры, установите флажки напротив соответствующих прототипов триггеров, затем нажмите нужную кнопку.

3 Прототипы графиков

Обзор

В этом разделе отображаются прототипы графиков правила низкоуровневого обнаружения на узле сети. Прототипы графиков являются основой реальных графиков узла сети, которые создаются в ходе низкоуровневого обнаружения.



Отображаемые данные:

Column	Description
Name	Отображается имя прототипа графика в виде синей ссылки. Нажатие на имя открывает форму настройки прототипа графика. Если прототип графика принадлежит связанному шаблону, перед именем графика отображается имя шаблона в виде серой ссылки. Нажатие на ссылку шаблона открывает 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. Вы можете переключаться между 'Yes' и 'No', 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:

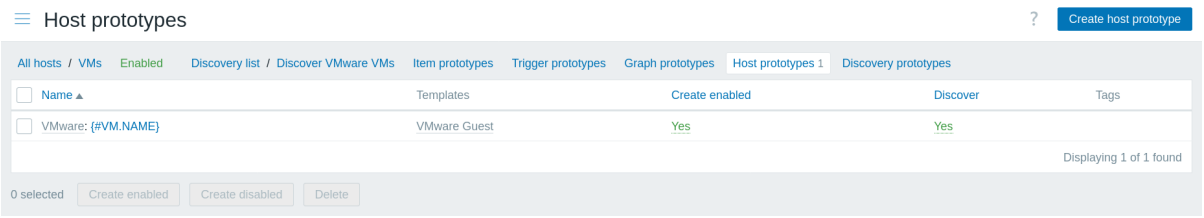
- *Удалить* — удалить эти прототипы графиков

Чтобы использовать эти параметры, установите флажки напротив соответствующих прототипов графиков, затем нажмите требуемую кнопку.

4 Прототипы узлов сети

Обзор

В этом разделе отображаются прототипы узлов сети правила низкоуровневого обнаружения на узле сети. Прототипы узлов сети являются основой для реальных узлов сети, которые создаются в ходе низкоуровневого обнаружения.



Отображаемые данные:

Column	Description
Name	Имя прототипа узла сети, отображаемое как синяя ссылка. Щелчок по имени открывает форму настройки прототипа узла сети. Если прототип узла сети принадлежит связанному шаблону, имя шаблона отображается перед именем узла сети как серая ссылка. Щелчок по ссылке шаблона открывает список прототипов узлов сети на уровне связанного шаблона.
Templates	Отображаются шаблоны прототипа узла сети.
Create enabled	Создавать узел сети на основе этого прототипа как: **Yes** — включено **No** — отключено. Вы можете переключаться между «Yes» и «No», щёлкая по ним.
Discover	Обнаруживать узел сети на основе этого прототипа: **Yes** — обнаруживать **No** — не обнаруживать. Вы можете переключаться между «Yes» и «No», щёлкая по ним.
Tags	Отображаются теги прототипа узла сети.

Чтобы настроить новый прототип узла сети, нажмите кнопку *Create host prototype* в правом верхнем углу.

Параметры массового редактирования

Кнопки под списком предоставляют несколько вариантов массового редактирования:

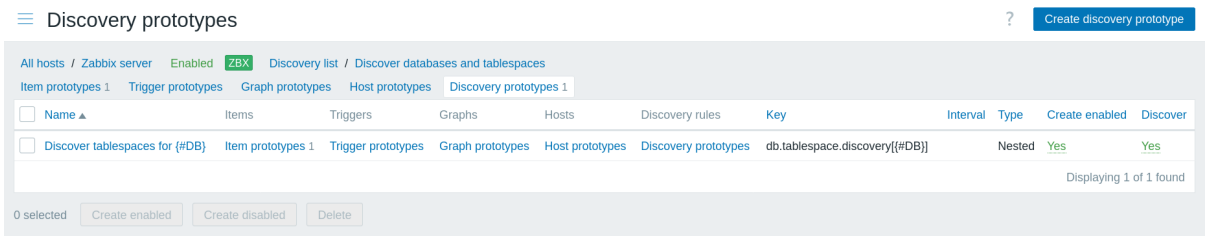
- *Create enabled* — создать эти узлы сети как *Enabled*
- *Create disabled* — создать эти узлы сети как *Disabled*
- *Delete* — удалить эти прототипы узлов сети

Чтобы использовать эти параметры, установите флажки напротив соответствующих прототипов узлов сети, затем нажмите нужную кнопку.

5 Прототипы обнаружения

Обзор

В этом разделе отображаются прототипы обнаружения правила обнаружения низкого уровня 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. Щелчок по имени открывает форму настройки прототипа обнаружения. Если прототип обнаружения принадлежит связанному шаблону, имя шаблона отображается перед именем узла сети в виде серой ссылки. Щелчок по ссылке шаблона открывает 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. Обратите внимание, что обнаружение также можно выполнить немедленно, нажав кнопку *Execute now* 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*). Щелчком по статусу его можно изменить — с Да на Нет (и обратно).
Обнаруживать	Отображается статус обнаружения прототипа обнаружения — обнаруживать (*Да*) или не обнаруживать (*Нет*). Щелчком по статусу его можно изменить — с Да на Нет (и обратно).

Чтобы настроить новый прототип обнаружения, нажмите кнопку *Create discovery prototype* в правом верхнем углу.

Параметры массового редактирования

Кнопки под списком предоставляют несколько вариантов массового редактирования:

- *Создать включёнными* — создать эти прототипы обнаружения как *Включено*
- *Создать отключёнными* — создать эти прототипы обнаружения как *Отключено*
- *Удалить* — удалить эти прототипы обнаружения

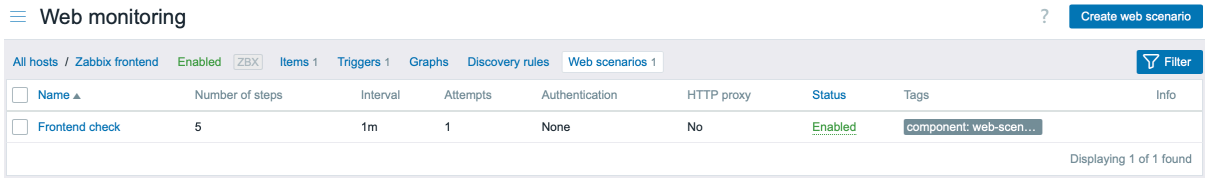
Чтобы использовать эти параметры, установите флажки напротив соответствующих прототипов обнаружения, затем нажмите нужную кнопку.

5 Веб-сценарии

Обзор

К списку веб-сценариев для узла сети можно перейти из *Сбор данных* → *Узлы сети*, нажав *Веб* для соответствующего узла сети.

Отображается список существующих веб-сценариев.



Отображаемые данные:

Column	Description
Name	Имя веб-сценария. Нажатие на имя веб-сценария открывает форму настройки веб-сценария. Если веб-сценарий узла сети принадлежит шаблону, перед именем веб-сценария отображается имя шаблона в виде серой ссылки. Нажатие на ссылку шаблона откроет список веб-сценариев на уровне шаблона.
Number of steps	Количество шагов, содержащихся в сценарии.
Update interval	Как часто выполняется сценарий.
Attempts	Сколько попыток выполняется для выполнения шагов веб-сценария.
Authentication	Отображается метод аутентификации — Basic, NTLM, Kerberos, Digest или None.
HTTP proxy	Отображает HTTP-прокси или 'No', если он не используется.
Status	Отображается состояние веб-сценария — *Включено* или *Отключено*. Нажав на состояние, вы можете изменить его.
Tags	Отображаются теги веб-сценария. Может отображаться до трех тегов (пары имя:значение). Если тегов больше, отображается ссылка "..." , позволяющая увидеть все теги при наведении курсора мыши.
Info	Если все работает корректно, в этом столбце значок не отображается. В случае ошибок отображается квадратный значок с буквой "i". Наведите курсор 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 предоставляют несколько вариантов массового редактирования:

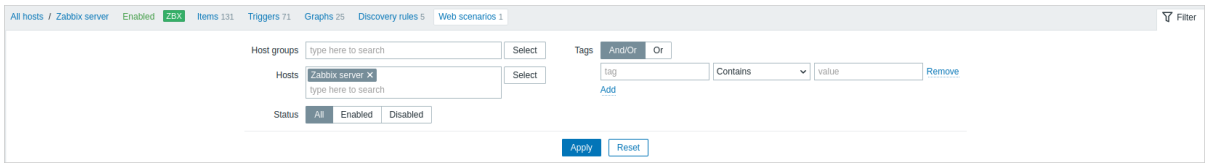
- *Enable* — изменить статус сценария на *Enabled*
- *Disable* — изменить статус сценария на *Disabled*
- *Clear history and trends* — очистить данные истории и трендов для сценариев
- *Delete* — удалить веб-сценарии

Чтобы использовать эти параметры, установите флажки напротив соответствующих веб-сценариев, 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 неразрешёнными макросами.

Ссылка *Фильтр* доступна над списком веб-сценариев. Щелчок по ней откроет фильтр, в котором можно указать нужные критерии фильтрации.



Параметр	Описание
Группы узлов сети	Фильтрация по одной или нескольким группам узлов сети. При указании родительской группы узлов сети автоматически выбираются все вложенные группы узлов сети.
Узлы сети	Фильтрация по одному или нескольким узлам сети. Если выше уже выбраны группы узлов сети, выбор узлов сети ограничивается этими группами.
Состояние	Фильтрация по состоянию веб-сценария.

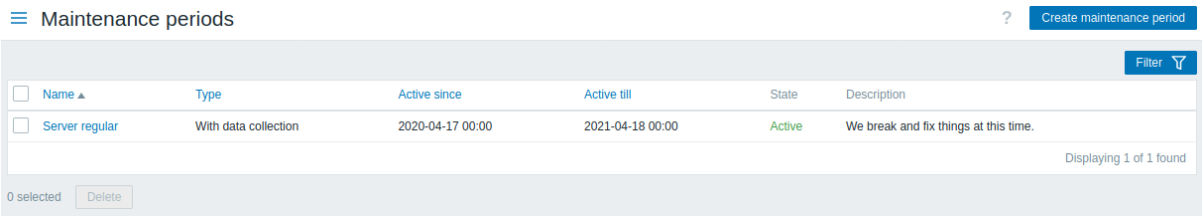
Параметр	Описание
Теги	Фильтрация по имени и значению тега веб-сценария. Можно как включать, так и исключать определённые теги и значения тегов. Можно задать несколько условий. Сопоставление имён тегов всегда чувствительно к регистру. Для каждого условия доступны несколько операторов: **Существует** — включить указанные имена тегов; **Равно** — включить указанные имена и значения тегов (с учётом регистра); **Содержит** — включить указанные имена тегов, у которых значения тегов содержат введённую строку (поиск подстроки, без учёта регистра); **Не существует** — исключить указанные имена тегов; **Не равно** — исключить указанные имена и значения тегов (с учётом регистра); **Не содержит** — исключить указанные имена тегов, у которых 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.

5 Обслуживание

Обзор

В разделе *Сбор данных* → *Обслуживание* пользователи могут настраивать и поддерживать периоды обслуживания для узлов сети.

Отображается список существующих периодов обслуживания с их подробными сведениями.



Отображаемые данные:

Column	Description
Name	Название периода обслуживания. Щелчок по названию периода обслуживания открывает форму настройки периода обслуживания.
Type	Отображается тип обслуживания: *Со сбором данных* или *Без сбора данных*
Active since	Дата и время, начиная с которых выполнение периодов обслуживания становится активным. Примечание: это время не активирует период обслуживания; периоды обслуживания необходимо задавать отдельно.
Active till	Дата и время, когда выполнение периодов обслуживания перестает быть активным.
State	Состояние периода обслуживания: **Скоро начнется** - скоро станет активным **Активен** - активен **Истек** - больше не активен
Description	Отображается описание периода обслуживания.

Чтобы настроить новый период обслуживания, нажмите кнопку *Создать период обслуживания* в правом верхнем углу.

Опции массового редактирования

Кнопка ниже списка предлагает одну опцию массового редактирования:

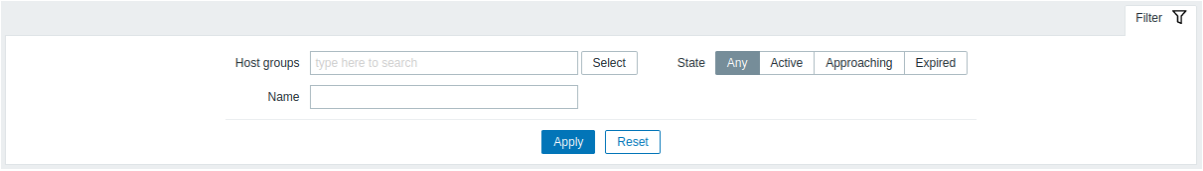
- *Удалить* - удалить периоды обслуживания

Для использования этой опции отметьте флажки перед соответствующими периодами обслуживания, затем нажмите на *Удалить*.

Использование фильтра

Для отображения только интересующих вас периодов обслуживания можно использовать фильтр. Для более эффективного поиска данные ищутся с нераскрытыми макросами.

Ссылка *Фильтр* доступна выше списка периодов обслуживания. При нажатии на неё откроется фильтр, где можно отобрать периоды обслуживания по группе узлов сети, имени и состоянию.



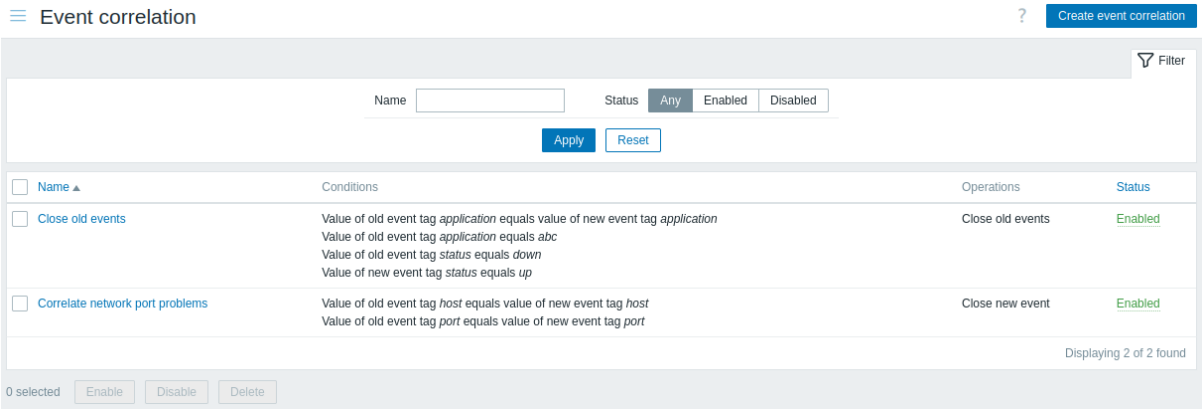
Расчет очередей во время обслуживания

Attention:
Прокси Zabbix не знает о периодах обслуживания; подробности см. в разделе Расчет очередей во время обслуживания.

6 Корреляция событий

Обзор

В разделе *Сбор данных* → *Корреляция событий* пользователи могут настраивать и поддерживать глобальные правила корреляции для событий Zabbix.



Отображаемые данные:

Столбец	Описание
Имя	Имя правила корреляции. Щелчок по имени правила корреляции открывает форму настройки.
Условия	Отображаются условия правила корреляции.
Операции	Отображаются операции правила корреляции.
Статус	Отображается статус правила корреляции — *Включено* или *Отключено*. Щелчком по статусу его можно изменить.

Чтобы настроить новое правило корреляции, нажмите кнопку *Создать корреляцию событий* в правом верхнем углу.

Параметры массового редактирования

Кнопки под списком предоставляют несколько вариантов массового редактирования:

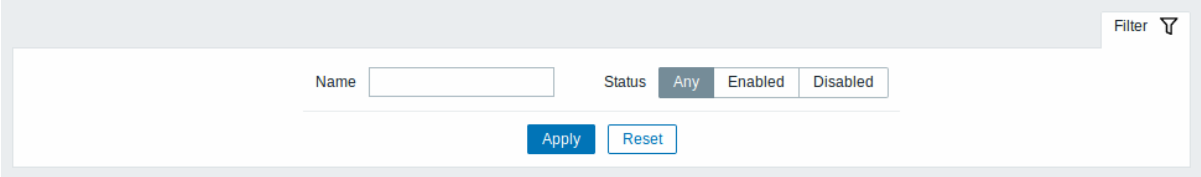
- *Enable* — изменить состояние правила корреляции на *Enabled*
- *Disable* — изменить состояние правила корреляции на *Disabled*
- *Delete* — удалить правила корреляции

Чтобы использовать эти параметры, установите флажки напротив соответствующих правил корреляции, затем нажмите нужную кнопку.

Использование фильтра

Вы можете использовать фильтр, чтобы отображать только те правила корреляции, которые вас интересуют. Для повышения производительности поиска данные ищутся с неразрешёнными макросами.

Ссылка *Фильтр* доступна над списком правил корреляции. Если щёлкнуть по ней, станет доступен фильтр, в котором можно отфильтровать правила корреляции по имени и состоянию.

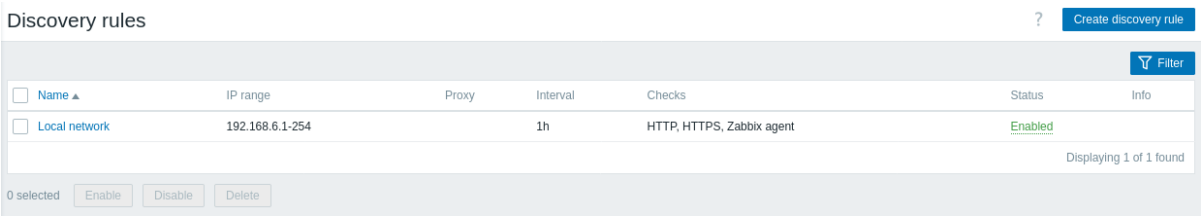


7 Обнаружение

Обзор

В разделе *Сбор данных* → *Обнаружение* пользователи могут настраивать и поддерживать правила обнаружения.

Отображается список существующих правил обнаружения с подробной информацией о них.



Отображаемые данные:

Column	Description
Name	Имя правила обнаружения. Щелчок по имени правила обнаружения открывает форму настройки правила обнаружения.
IP range	Отображается диапазон IP-адресов, используемый для сканирования сети.
Proxy	Отображается имя прокси, если обнаружение выполняется прокси.
Interval	Отображается частота выполнения обнаружения.
Checks	Отображаются типы проверок, используемые для обнаружения.
Status	Отображается состояние правила обнаружения — *Включено* или *Отключено*. Щелчком по состоянию его можно изменить.
Info	Если всё работает корректно, в этом столбце ничего не отображается. В случае ошибок отображается красный значок информации с буквой ”i”. Наведите курсор 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 предоставляют несколько вариантов массового редактирования:

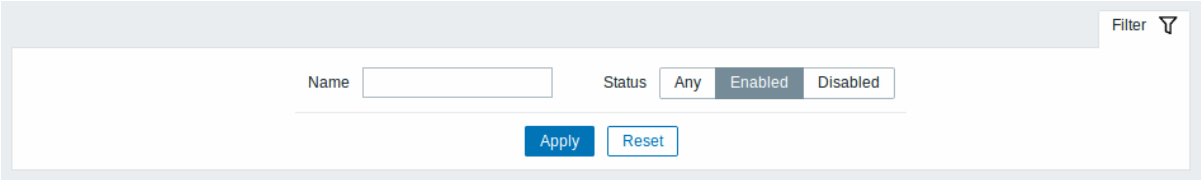
- *Включить* — изменить состояние правила обнаружения на *Включено*
- *Отключить* — изменить состояние правила обнаружения на *Отключено*
- *Удалить* — удалить правила обнаружения

Чтобы использовать эти параметры, установите флажки напротив соответствующих правил 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 неразрешёнными макросами.

Ссылка *Фильтр* доступна над списком правил обнаружения. Если вы нажмёте на неё, станет доступен фильтр, в котором можно отфильтровать правила обнаружения по имени и состоянию.



7 Оповещения

Обзор

В этом меню есть разделы, связанные с настройкой оповещений в Zabbix.

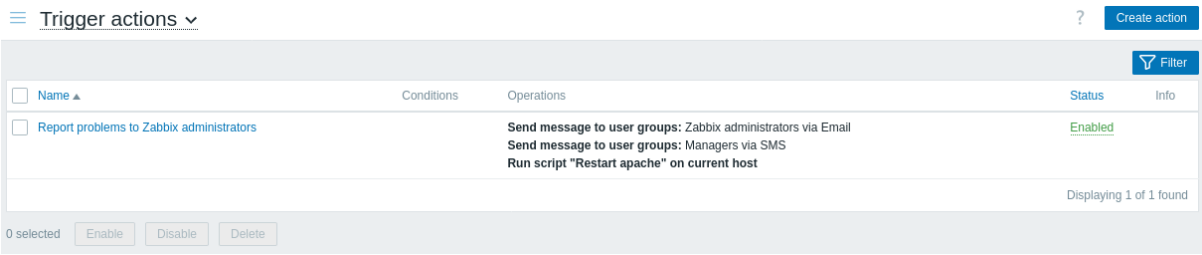
1 Действия

Обзор

В разделе *Оповещения* → *Действия* пользователи могут настраивать и поддерживать действия.

Отображаемые действия — это действия, назначенные выбранному источнику событий (триггер, сервисы, обнаружение, авторегистрация, внутренние действия).

Чтобы переключиться на другой источник событий, нажмите *Действия* в разделе меню *Оповещения*. Также можно переключаться между источниками с помощью выпадающего списка в заголовке в левом верхнем углу.



Отображаемые данные:

Column	Description
Name	Имя действия. Щелчок по имени действия открывает форму настройки действия.
Conditions	Отображаются условия действия.
Operations	Отображаются операции действия. В списке операций также отображается тип средства оповещения (email, SMS или скрипт), используемого для уведомления, а также имя и фамилия (в скобках после имени пользователя) получателя уведомления. Операция действия может быть как уведомлением, так и удалённой командой в зависимости от выбранного типа операции.
Status	Отображается статус действия — *Включено* или *Отключено*. Щелчком по статусу его можно изменить. Подробнее о том, что происходит, если действие отключено во время выполняющейся эскалации, см. в разделе Эскалации.
Info	Если всё работает корректно, в этом столбце значок не отображается. В случае ошибок, например отсутствующих операций действия или условий после обновления/обнаружения низкого уровня, отображается предупреждающий значок. Наведите курсор 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 Super admin действия отображаются в соответствии с настройками прав доступа. Это означает, что в некоторых случаях пользователь без прав Super admin не может просматривать полный список действий из-за определённых ограничений прав доступа. Действие отображается пользователю без прав Super admin, если выполняются следующие условия:

- Пользователь имеет доступ на чтение-запись к группам узлов сети, узлам сети, шаблонам и триггерам в условиях действия
- Пользователь имеет доступ на чтение-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 предоставляют несколько вариантов массового редактирования:

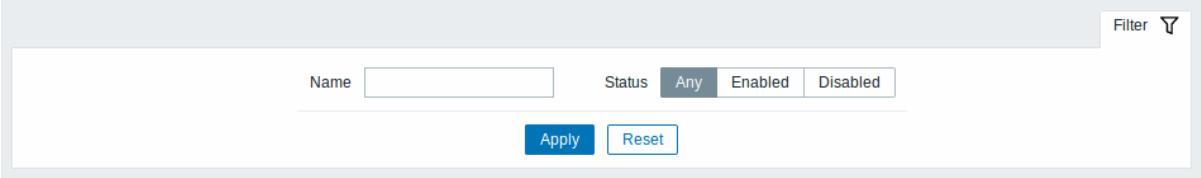
- *Enable* — изменить статус действия на *Enabled*
- *Disable* — изменить статус действия на *Disabled*
- *Delete* — удалить действия

Чтобы использовать эти параметры, установите флажки напротив соответствующих действий, затем нажмите требуемую кнопку.

Использование фильтра

Вы можете использовать фильтр, чтобы отображать только те действия, которые вас интересуют. Для повышения производительности поиска данные ищутся с неразрешёнными макросами.

Ссылка *Фильтр* доступна над списком действий. Если щёлкнуть по ней, станет доступен фильтр, в котором можно отфильтровать действия по имени и состоянию.



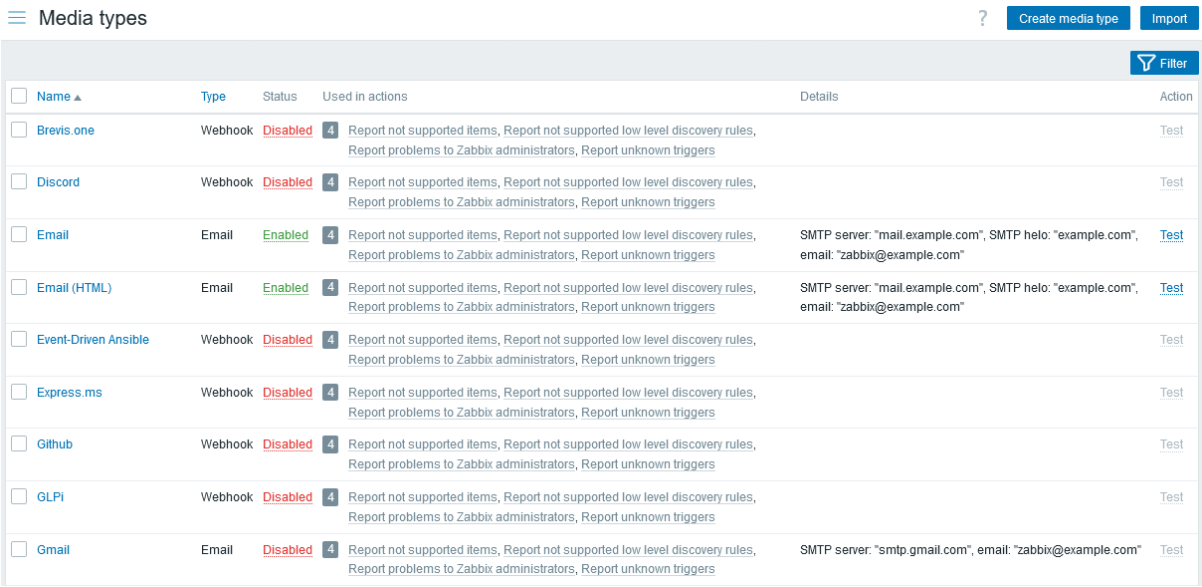
2 Способы оповещения

Обзор

В разделе *Оповещения → Способы оповещения* пользователи могут настраивать и поддерживать информацию о способах оповещения.

Информация о способе оповещения содержит общие инструкции по использованию средства в качестве канала доставки уведомлений. Конкретные сведения, например индивидуальные адреса электронной почты, на которые следует отправлять уведомления, хранятся у отдельных пользователей.

Отображается список существующих способов оповещения с их подробными сведениями.



Отображаемые данные:

Column	Description
Name	Имя способа оповещения. Щелчок по имени открывает форму настройки способа оповещения.
Type	Отображается тип средства оповещения (email, SMS и т. д.).

Column	Description
Status	Отображается состояние способа оповещения — *Включено* или *Отключено*. Щелчком по состоянию его можно изменить.
Used in actions	Отображаются действия, в которых используется способ оповещения, перед которыми указано общее количество этих действий. Щелчок по имени действия открывает форму настройки действия. Если у пользователя нет прав доступа к действию, имя не является кликабельным.
Details	Отображается подробная информация о способе оповещения.
Actions	Доступно следующее действие: **Test** — нажмите, чтобы открыть форму тестирования, где можно ввести параметры способа оповещения (например, адрес получателя, тему и текст тестового сообщения) и отправить тестовое сообщение, чтобы убедиться, что настроенный способ оповещения работает. См. также: Тестирование способа оповещения.

Чтобы настроить новый способ оповещения, нажмите кнопку *Создать способ оповещения* в правом верхнем углу.

Чтобы импортировать способ оповещения, нажмите кнопку *Импорт* в правом верхнем углу.

Параметры массового редактирования

Кнопки под списком предоставляют несколько вариантов массового редактирования:

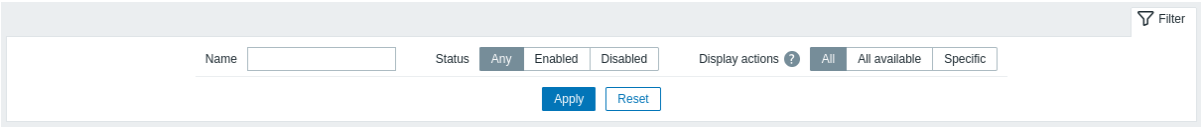
- *Enable* — изменить состояние типа оповещения на *Enabled*
- *Disable* — изменить состояние типа оповещения на *Disabled*
- *Export* — экспортировать типы оповещений в файл YAML, XML или JSON
- *Delete* — удалить типы оповещений

Чтобы использовать эти параметры, установите флажки напротив соответствующих типов 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 неразрешёнными макросами.

Ссылка *Фильтр* доступна над списком типов медиа. Если щёлкнуть по ней, станет доступен фильтр, в котором можно отфильтровать типы медиа по имени и состоянию. Кроме того, вы можете использовать фильтр для отображения действий в столбце *Используется в действиях* в зависимости от области использования их типа медиа (определяется параметром *Отправить на тип медиа* в сведениях об операции действия).



3 Скрипты

Обзор

В разделе *Оповещения* > *Скрипты* можно настраивать и поддерживать пользовательские глобальные скрипты.

Глобальные скрипты, в зависимости от настроенной области действия, а также прав пользователя, доступны для выполнения:

- из меню узла сети в различных разделах веб-интерфейса (*Панель мониторинга*, *Проблемы*, *Последние данные*, *Карты сети* и т. д.)
- из меню события
- могут запускаться как операция действия

Скрипты выполняются на Zabbix агент, Zabbix сервер (прокси) или только на Zabbix сервере.
См. также Выполнение команд.

Как на Zabbix агент, так и на Zabbix прокси удалённые скрипты по умолчанию отключены.
Их можно включить следующим образом:

- Для удалённых команд, выполняемых на Zabbix агент:
  - добавить параметр AllowKey=system.run[<command>,*] для каждой разрешённой команды в конфигурацию агент, где * обозначает режим wait и nowait;

- Для удалённых команд, выполняемых на Zabbix прокси:
  - **Предупреждение: Не требуется включать удалённые команды на Zabbix прокси, если удалённые команды выполняются на Zabbix агент, который мониторится Zabbix прокси.** Однако, если всё же требуется выполнять удалённые команды на Zabbix прокси, установите параметр *EnableRemoteCommands* в значение '1' в конфигурации прокси.

Выполнение глобальных скриптов на Zabbix сервере можно отключить, установив `EnableGlobalScripts=0` в конфигурации сервера.
Для новых установок, начиная с Zabbix 7.0, выполнение глобальных скриптов на Zabbix сервере по умолчанию отключено.

Отображается список существующих скриптов с их подробной информацией.



Отображаемые данные:

Column	Description
Name	Имя скрипта. Щелчок по имени скрипта открывает форму настройки скрипта.
Scope	Область действия скрипта — операция действия, ручное действие с узлом сети или ручное действие с событием. Этот параметр определяет, где доступен скрипт.
Used in actions	Отображаются все действия, в которых используется скрипт, перед которыми указано общее количество этих действий. Щелчок по имени действия открывает форму настройки действия. Если у пользователя нет прав на действие, имя не является ссылкой.
Type	Отображается тип скрипта — команда *URL*, *Webhook*, *Script*, *SSH*, *Telnet* или *IPMI*.
Execute on	Отображается, будет ли скрипт выполняться на Zabbix агент, Zabbix прокси или сервере, либо только на Zabbix сервере.
Commands	Отображаются все команды, которые будут выполнены в рамках скрипта. Для вебхуков здесь ничего не отображается.
User group	Отображается группа пользователей, для которой доступен скрипт (или *All* для всех групп пользователей).
Host group	Отображается группа узлов сети, для которой доступен скрипт (или *All* для всех групп узлов сети).
Host access	Отображается уровень прав для группы узлов сети — *Read* или *Write*. Только пользователи с требуемым уровнем прав смогут выполнять скрипт.

Чтобы настроить новый скрипт, нажмите кнопку *Create script* в правом верхнем углу.

Параметры массового редактирования

Кнопка под списком предоставляет один вариант массового редактирования:

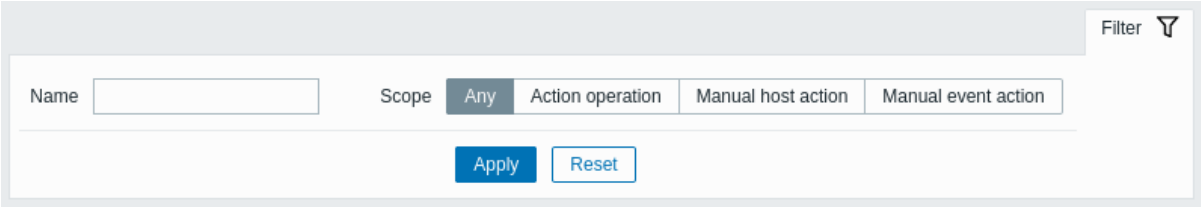
- *Удалить* — удалить скрипты

Чтобы использовать этот вариант, установите флажки напротив соответствующих скриптов и нажмите *Удалить*.

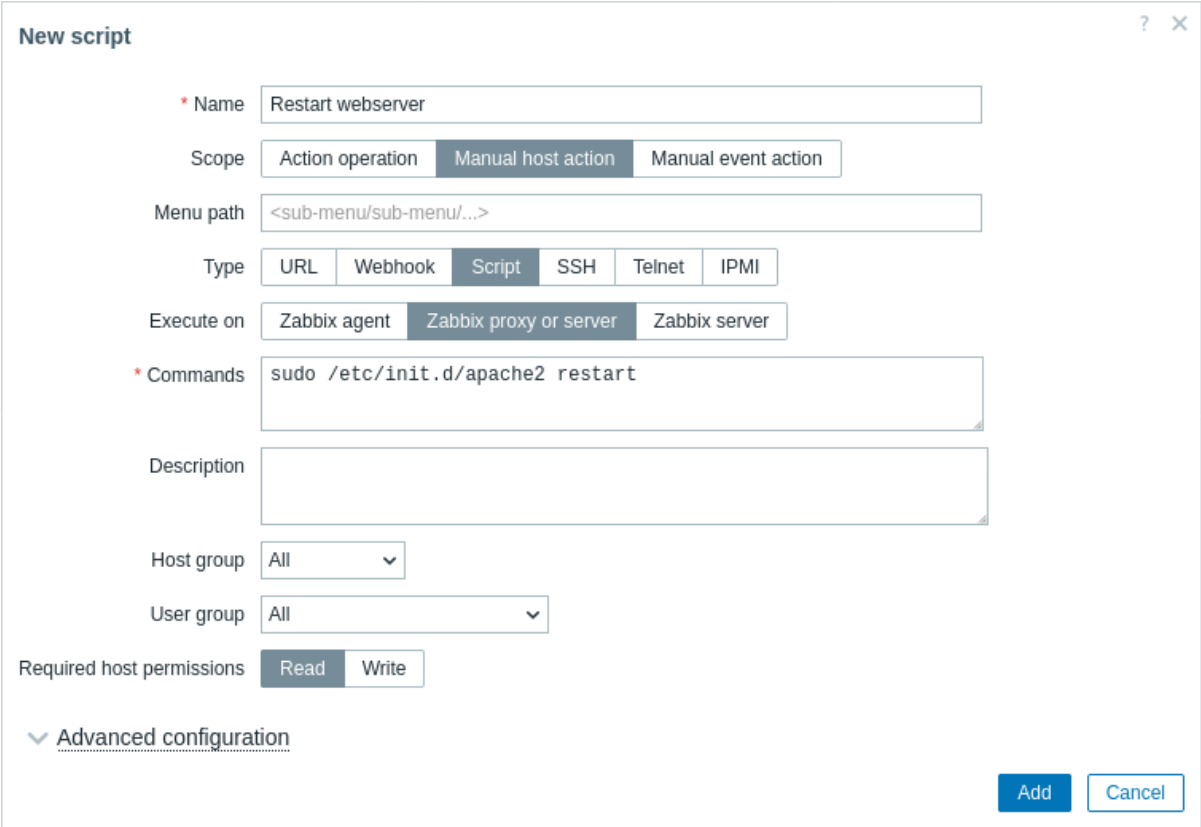
Использование фильтра

Вы можете использовать фильтр, чтобы отображать только те скрипты, которые вас интересуют. Для повышения производительности поиска данные ищутся с неразрешёнными макросами.

Ссылка *Фильтр* доступна над списком скриптов. Если вы нажмёте на неё, станет доступен фильтр, в котором можно отфильтровать скрипты по имени и области действия.



Настройка глобального скрипта



Note:
Для повышения безопасности в глобальных скриптах рекомендуется использовать функции макросов вместо обычных макросов, так как макросы не экранируются автоматически.

Атрибуты скрипта:

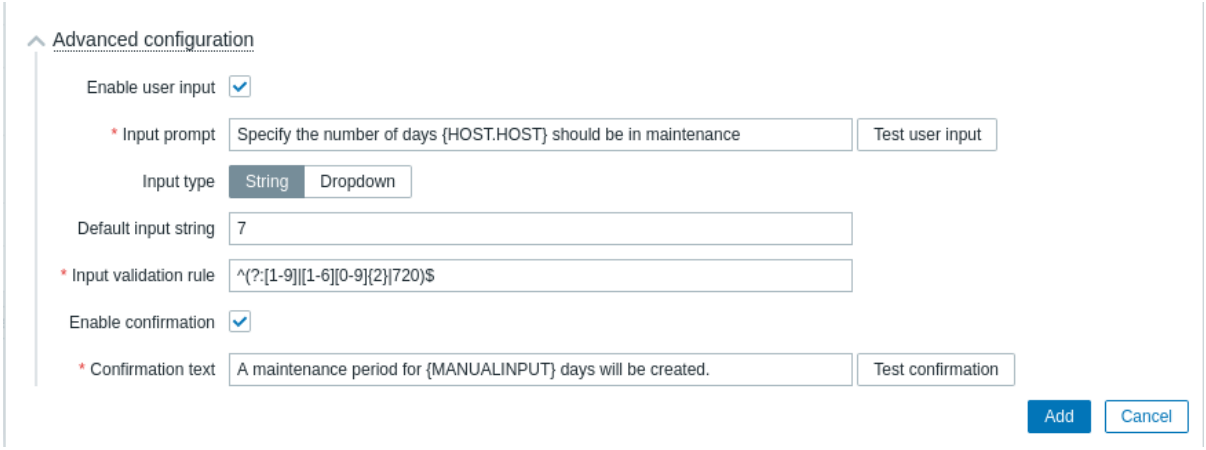
Параметр	Описание
Имя	Уникальное имя скрипта. Например, `Clear /tmp filesystem`

Параметр	Описание
Область применения	Область применения скрипта — операция действия, ручное действие с узлом сети или ручное действие с событием. Этот параметр определяет, где можно использовать скрипт: в удалённых командах операций действия, из меню узла сети или из меню события соответственно. Если установить область применения в значение «Операция действия», скрипт будет доступен всем пользователям, имеющим доступ к *Оповещения* > *Действия*. Если скрипт фактически используется в действии, его область применения нельзя изменить с «операция действия» на другое значение. **Поддержка макросов** Область применения влияет на набор доступных макросов. Например, в скриптах поддерживаются пользовательские макросы ({USER.*}), чтобы можно было передавать информацию о пользователе, запустившем скрипт. Однако они не поддерживаются, если область применения скрипта — операция действия, так как операции действия выполняются автоматически. Макрос {MANUALINPUT} позволяет указать ручной ввод во время выполнения скрипта. Он поддерживается для скриптов ручных действий с узлом сети и ручных действий с событием. Чтобы узнать, какие ещё макросы поддерживаются, выполните поиск по строкам «Уведомления и команды на основе триггеров/Команды на основе триггеров», «Скрипты ручных действий с узлом сети» и «Скрипты ручных действий с событием» в таблице поддерживаемых макросов. Обратите внимание: если макрос может разрешаться в значение с пробелами (например, имя узла сети), не забудьте при необходимости заключить его в кавычки.
Путь меню	Желаемый путь меню к скрипту. Например, `Default` или `Default/` отобразит скрипт в соответствующем каталоге. Меню могут быть вложенными, например: `Main menu/Sub menu1/Sub menu2`. При доступе к скриптам через меню узла сети/события в разделах мониторинга они будут организованы в соответствии с указанными каталогами. Это поле отображается только если в *Область применения* выбрано «Ручное действие с узлом сети» или «Ручное действие с событием».
Тип	Нажмите соответствующую кнопку, чтобы выбрать тип скрипта: **URL**, **Вебхук**, **Скрипт**, **SSH**, **Telnet** или команда **IPMI**. Тип **URL** доступен только если в *Область применения* выбрано «Ручное действие с узлом сети» или «Ручное действие с событием».
Тип скрипта: **URL**	
URL	Укажите URL для быстрого доступа из меню узла сети или меню события. Поддерживаются макросы и пользовательские пользовательские макросы. Поддержка макросов зависит от области применения скрипта (см. *Область применения* выше). Используйте макрос {MANUALINPUT} в этом поле, чтобы можно было указать ручной ввод во время выполнения скрипта, например: `http://{MANUALINPUT}/zabbix/zabbix.php?action=dashboard.view` Значения макросов не должны быть URL-кодированы.
Открывать в новом окне	Определяет, должен ли URL открываться в новой вкладке браузера или в текущей.
Тип скрипта: **Вебхук**	
Параметры	Укажите переменные вебхука как пары атрибут-значение. См. также: настройка способа оповещения Webhook. В значениях параметров поддерживаются макросы и пользовательские пользовательские макросы. Поддержка макросов зависит от области применения скрипта (см. *Область применения* выше).
Скрипт	Введите код JavaScript в модальном редакторе, который открывается при щелчке в поле параметра или по значку карандаша рядом с ним. Поддержка макросов зависит от области применения скрипта (см. *Область применения* выше). См. также: настройка способа оповещения Webhook, Дополнительные объекты Javascript.
Тайм-аут	Тайм-аут выполнения JavaScript (1-60s, по умолчанию 30s). Поддерживаются суффиксы времени, например 30s, 1m.
Тип скрипта: **Скрипт**	

Параметр	Описание
Выполнять на	Нажмите соответствующую кнопку, чтобы выполнить shell-скрипт на: **Zabbix агент** — скрипт будет выполнен агентом Zabbix на узле сети, если элемент данных system.run разрешён **Zabbix прокси или сервер** — скрипт будет выполнен Zabbix прокси или сервером в зависимости от того, отслеживается ли узел сети прокси или сервером. Он будет выполнен на прокси, если это разрешено параметром EnableRemoteCommands. Он будет выполнен на сервере, если глобальные скрипты включены параметром сервера EnableGlobalScripts. **Zabbix сервер** — скрипт будет выполнен только Zabbix сервером. Этот вариант будет недоступен, если глобальные скрипты отключены параметром сервера EnableGlobalScripts.
Команды	Введите полный путь к командам, которые должны быть выполнены в скрипте. Поддержка макросов зависит от области применения скрипта (см. *Область применения* выше). Поддерживаются пользовательские пользовательские макросы.
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».
Кодовая фраза	Введите кодовую фразу. Это поле доступно, если в качестве метода аутентификации выбран «Открытый ключ».
Порт	Введите порт удалённой службы SSH на целевом узле сети, к которому будет подключаться Zabbix.
Команды	Введите команды. Поддержка макросов зависит от области применения скрипта (см. *Область применения* выше). Поддерживаются пользовательские пользовательские макросы.
Тип скрипта: **Telnet**	
Имя пользователя	Введите имя пользователя.
Пароль	Введите пароль.
Порт	Введите порт удалённой службы Telnet на целевом узле сети, к которому будет подключаться Zabbix.
Команды	Введите команды. Поддержка макросов зависит от области применения скрипта (см. *Область применения* выше). Поддерживаются пользовательские пользовательские макросы.
Тип скрипта: **IPMI**	
Команда	Введите команду IPMI. Поддержка макросов зависит от области применения скрипта (см. *Область применения* выше). Поддерживаются пользовательские пользовательские макросы.
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 *Область применения* выбрано «Ручное действие с узлом сети» или «Ручное действие с событием».
Требуемые права на узел сети	Выберите уровень прав для группы узлов сети — *Чтение* или *Запись*. Только пользователи с требуемым уровнем прав смогут выполнять скрипт. Это поле отображается только если в *Область применения* выбрано «Ручное действие с узлом сети» или «Ручное действие с событием».
Расширенная настройка	Нажмите на метку *Расширенная настройка*, чтобы отобразить параметры расширенной настройки. Это поле отображается только если в *Область применения* выбрано «Ручное действие с узлом сети» или «Ручное действие с событием».

Расширенная конфигурация

Дополнительные параметры конфигурации доступны в сворачиваемом разделе *Расширенная конфигурация*:



Параметр	Описание
Включить пользовательский ввод	Установите флажок, чтобы включить ручной пользовательский ввод перед выполнением скрипта. Ручной пользовательский ввод заменит значение макроса {MANUALINPUT} в скрипте. См. также: Ручной пользовательский ввод.
Подсказка для ввода	Введите произвольный текст-подсказку для пользовательского ввода. Этот текст будет отображаться над полем ввода во всплывающем окне *Ручной ввод*. Чтобы увидеть предварительный просмотр всплывающего окна *Ручной ввод*, нажмите *Тест пользовательского ввода*. Предварительный просмотр также позволяет проверить, соответствует ли строка ввода правилу валидации ввода (см. параметры ниже). Поддержка макросов и пользовательских макросов зависит от области действия скрипта (см. *Область действия* в общих параметрах конфигурации скрипта).
Тип ввода	Выберите тип ручного ввода: **Строка** — одна строка; **Выпадающий список** — значение выбирается из нескольких вариантов выпадающего списка.
Параметры выпадающего списка	Введите уникальные значения для выпадающего списка пользовательского ввода в виде списка, разделённого запятыми. Чтобы включить пустой вариант в выпадающий список, добавьте лишнюю запятую в начале, середине или конце списка. Это поле отображается только если в качестве *Типа ввода* выбран 'Выпадающий список'.
Строка ввода по умолчанию	Введите строку пользовательского ввода по умолчанию (или оставьте пустым). Это поле будет проверяться по регулярному выражению, указанному в поле *Правило валидации ввода*. Введённое здесь значение будет по умолчанию отображаться во всплывающем окне *Ручной ввод*. Это поле отображается только если в качестве *Типа ввода* выбрана 'Строка'.
Правило валидации ввода	Введите регулярное выражение для проверки строки пользовательского ввода. Поддерживаются глобальные регулярные выражения. Это поле отображается только если в качестве *Типа ввода* выбрана 'Строка'.
Включить подтверждение	Установите флажок, чтобы перед выполнением скрипта отображалось сообщение с подтверждением. Эта возможность может быть особенно полезна для потенциально опасных операций (например, скрипта перезагрузки) или операций, выполнение которых может занять много времени.
Текст подтверждения	Введите произвольный текст подтверждения для всплывающего окна подтверждения, включаемого флажком выше (например, *Удалённая система будет перезагружена. Вы уверены?*). Чтобы увидеть, как будет выглядеть текст, нажмите *Тест подтверждения* рядом с полем. Макросы и пользовательские макросы пользователя поддерживаются. *Примечание:* при тестировании сообщения подтверждения макросы не будут развёрнуты.

Если настроены и ручной пользовательский ввод, и сообщение подтверждения, они будут отображаться в последовательных всплывающих окнах.

Ручной ввод пользователя

Ручной ввод пользователя позволяет передавать пользовательский параметр при каждом выполнении скрипта. Это избавляет от необходимости создавать несколько похожих пользовательских скриптов, отличающихся только одним параметром.

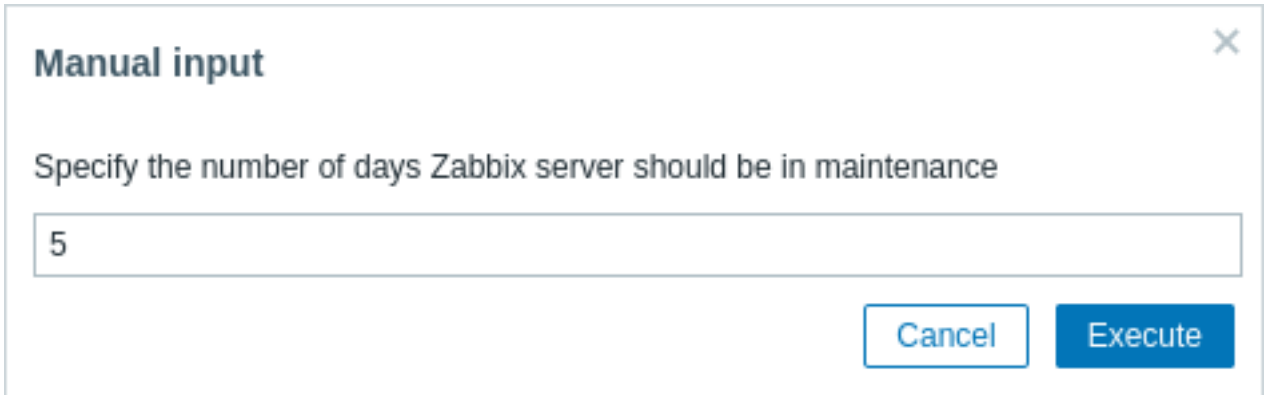
Например, при выполнении скрипта может потребоваться передать другое целое число или другой URL-адрес.

Чтобы включить ручной ввод пользователя:

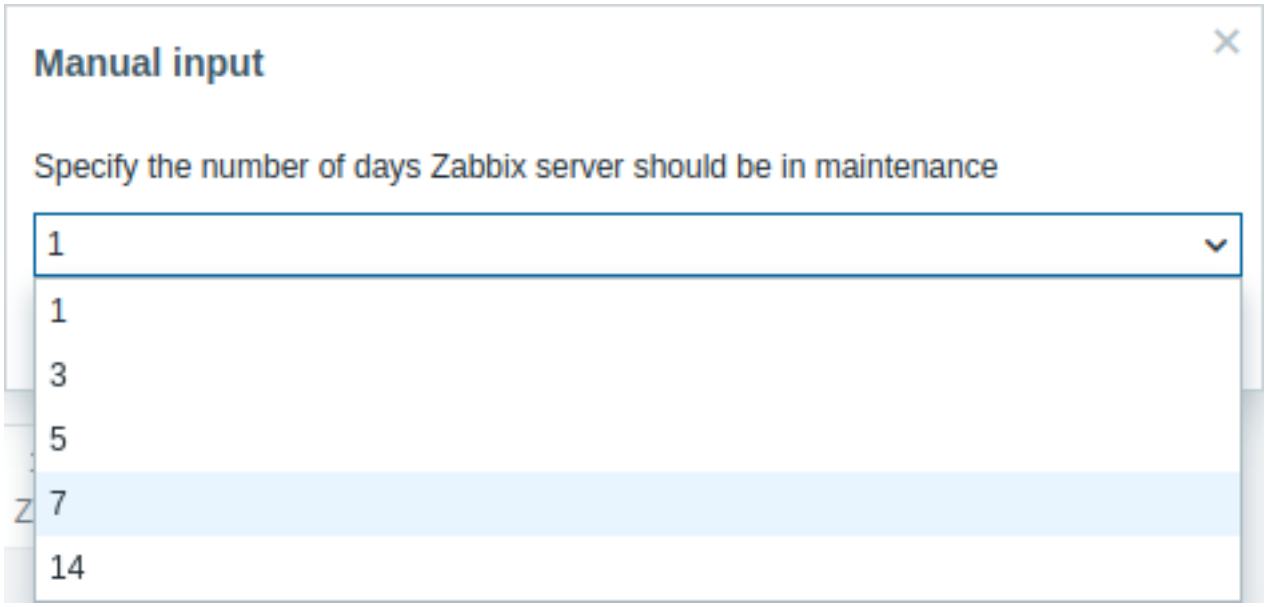
- используйте макрос {MANUALINPUT} в скрипте (командах, скрипте, параметре скрипта) там, где это необходимо; или в поле URL для URL-скриптов;
- в расширенной настройке скрипта включите ручной ввод пользователя и настройте параметры ввода.

Если ввод пользователя включен, перед выполнением скрипта пользователю будет показано всплывающее окно *Ручной ввод* с запросом указать пользовательское значение. Указанное значение заменит {MANUALINPUT} в скрипте.

В зависимости от конфигурации пользователю будет предложено ввести строковое значение:



Или выбрать значение из выпадающего списка заранее заданных вариантов:



Ручной ввод пользователя доступен только для скриптов, у которых область применения — «Ручное действие с узлом сети» или «Ручное действие с событием».

Выполнение скрипта и результат

Скрипты, запускаемые сервером Zabbix, выполняются в порядке, описанном на странице выполнения команд.

Результат выполнения скрипта отображается во всплывающем окне, которое появляется после запуска скрипта. Возвращаемое значение скрипта представляет собой стандартный вывод:

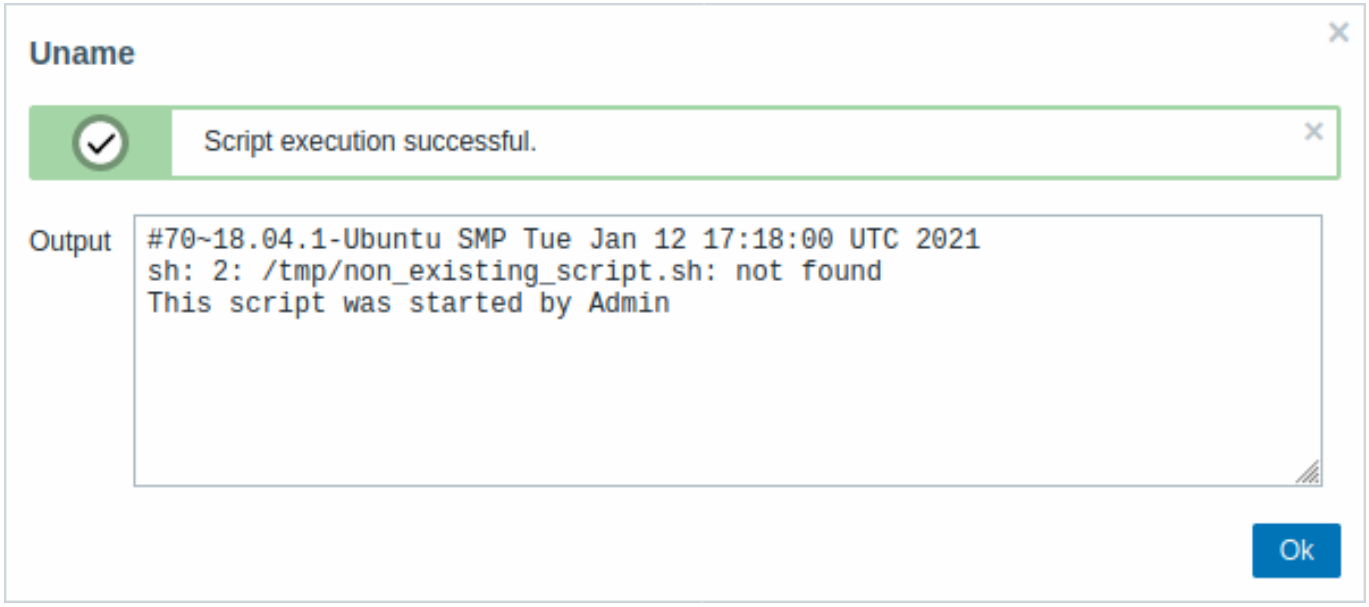
- Если скрипт завершается успешно (код завершения 0), возвращаемое значение ограничено 16 МБ (включая завершающие пробельные символы, которые усекаются).
- Если скрипт завершается с ошибкой (ненулевой код завершения), возвращаемое значение представляет собой стандартный поток ошибок, ограниченный 2 КБ.

По умолчанию Zabbix не сохраняет расширенный вывод скрипта. Чтобы сохранить полные сведения о выводе, вы можете реализовать журналирование 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 приведён 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.

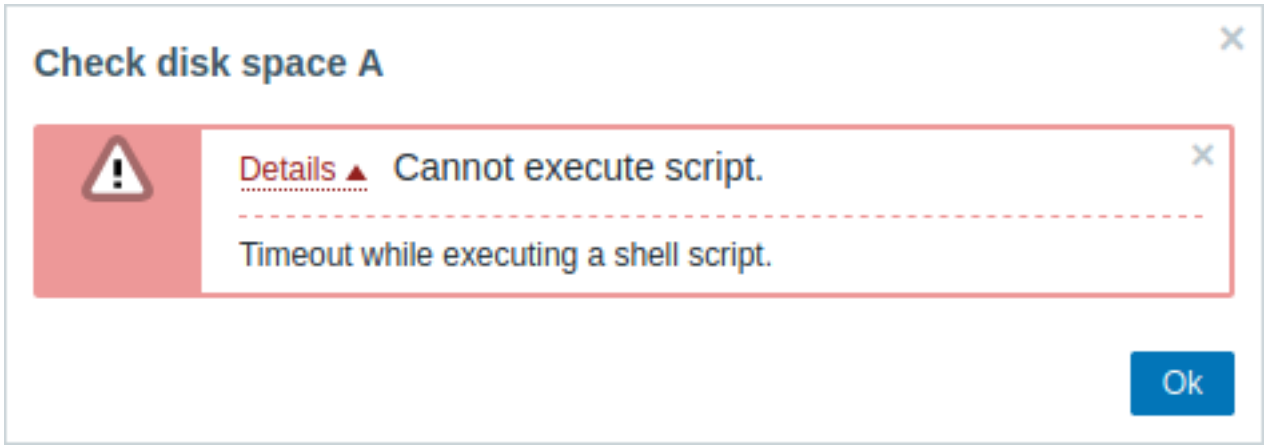
Время ожидания выполнения скриптов

Zabbix агент

Вы можете столкнуться с ситуацией, когда при выполнении скрипта возникает тайм-аут.

Ниже приведен пример скрипта, выполняемого на Zabbix агенте, и окно результата:

```
sleep 5
df -h
```



В этом случае будет показано следующее сообщение об ошибке:

```
Timeout while executing a shell script.
```

Чтобы избежать таких ситуаций, рекомендуется оптимизировать сам скрипт (в приведенном выше примере — «5»), а не изменять параметр `Timeout` в конфигурации Zabbix агента и конфигурации Zabbix сервера. Однако для Zabbix агента в активном режиме параметр `Timeout` в конфигурации Zabbix сервера должен быть как минимум на несколько секунд больше, чем параметр `RefreshActiveChecks` в конфигурации Zabbix агента. Это гарантирует, что серверу будет достаточно времени для получения результатов активных проверок от агента. Обратите внимание, что выполнение скриптов на активном агенте поддерживается начиная с Zabbix агента 7.0.

Если параметр `Timeout` был изменен в конфигурации Zabbix агента, появится следующее сообщение об ошибке:

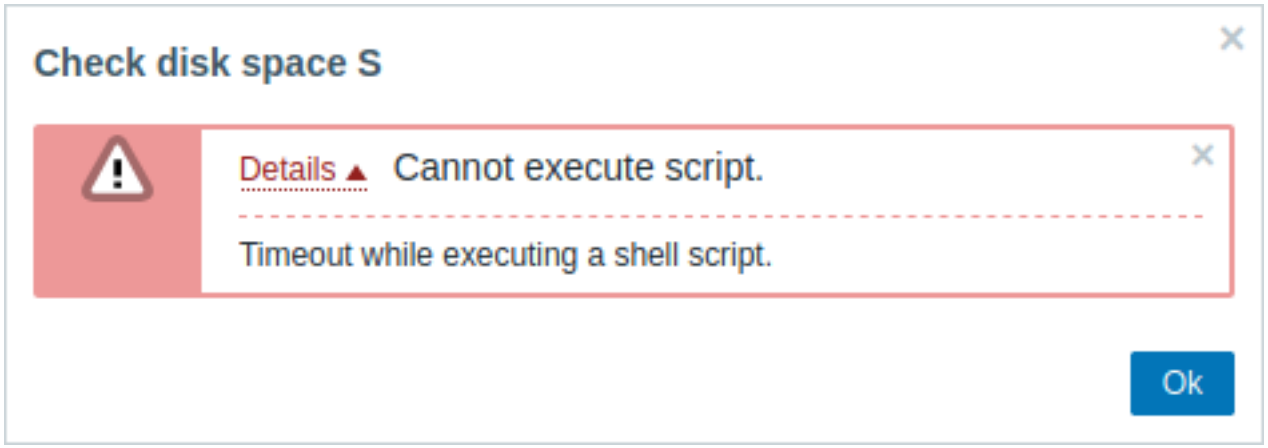
```
Get value from agent failed: ZBX_TCP_READ() timed out.
```

Это означает, что изменение было внесено в конфигурацию Zabbix агента, но также необходимо изменить параметр `Timeout` в конфигурации Zabbix сервера.

Сервер/прокси Zabbix

Ниже приведен пример скрипта, выполняемого на сервере Zabbix, и окно результата:

```
sleep 11
df -h
```



Также рекомендуется оптимизировать сам скрипт (вместо изменения параметра `TrapperTimeout` на соответствующее значение (в нашем случае, > 11) путем изменения конфигурации сервера Zabbix).

8 Пользователи

Обзор

Это меню содержит разделы, связанные с настройкой пользователей в Zabbix. Это меню доступно только пользователям типа SuperAdmin.

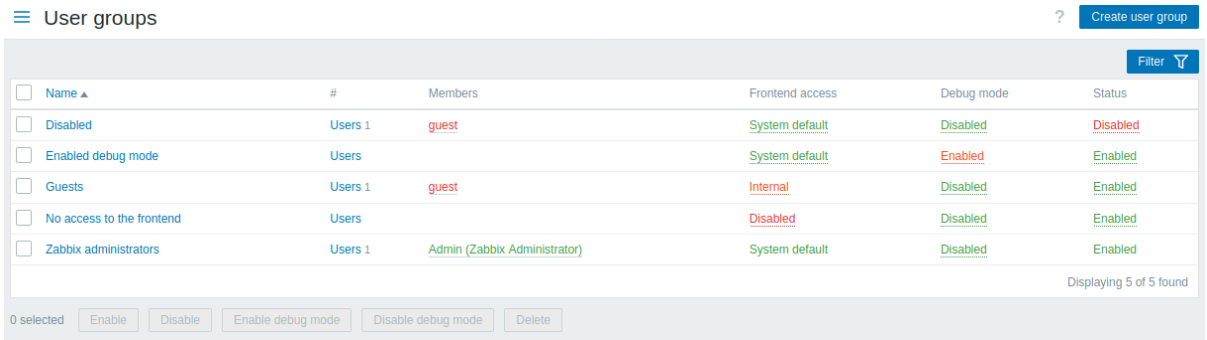
1 Группы пользователей

Обзор

В разделе *Пользователи* → *Группы пользователей* ведется учет групп пользователей.

Группы пользователей

Отображается список существующих групп пользователей с их сведениями.



Отображаемые данные:

Столбец	Описание
Имя	Имя группы пользователей. Щелчок по имени группы пользователей открывает форму настройки группы пользователей.
#	Количество пользователей в группе. Щелчок по *Пользователи* отобразит соответствующих пользователей, отфильтрованных в списке пользователей.

Столбец	Описание
Участники	Имена пользователей отдельных пользователей в группе пользователей (с именем и фамилией в скобках). Щелчок по имени пользователя открывает форму настройки пользователя. Пользователи из отключенных групп отображаются красным цветом.
Доступ к веб-интерфейсу	Отображается уровень доступа к веб-интерфейсу: **По умолчанию для системы** — пользователи проходят аутентификацию через Zabbix, LDAP или HTTP (в зависимости от метода аутентификации, заданного глобально); **Внутренний** — пользователи проходят аутентификацию через Zabbix; игнорируется, если глобально по умолчанию используется HTTP-аутентификация; **LDAP** — пользователи проходят аутентификацию через LDAP; игнорируется, если глобально по умолчанию используется HTTP-аутентификация; **Отключено** — доступ к веб-интерфейсу Zabbix запрещен для этой группы. Щелчок по текущему уровню позволяет изменить его.
Режим отладки	Отображается состояние режима отладки — *Включено* или *Отключено*. Щелчок по состоянию позволяет изменить его.
Состояние	Отображается состояние группы пользователей — *Включено* или *Отключено*. Щелчок по состоянию позволяет изменить его.

Чтобы настроить новую группу пользователей, нажмите кнопку *Создать группу пользователей* в правом верхнем углу.

Параметры массового редактирования

Кнопки под списком предоставляют несколько параметров массового редактирования:

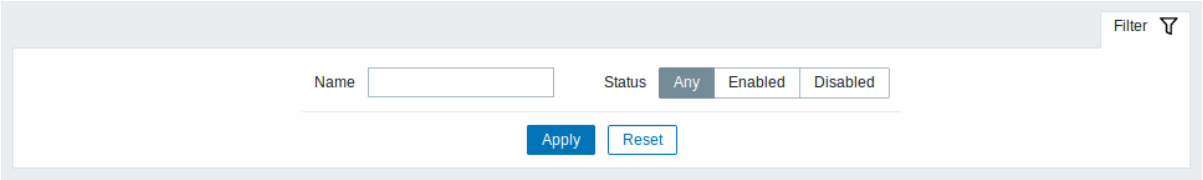
- *Включить* — изменить состояние группы пользователей на *Включено*
- *Отключить* — изменить состояние группы пользователей на *Отключено*
- *Включить режим отладки* — включить режим отладки для групп пользователей
- *Отключить режим отладки* — отключить режим отладки для групп пользователей
- *Удалить* — удалить группы пользователей

Чтобы использовать эти параметры, установите флажки напротив соответствующих групп 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 неразрешёнными макросами.

Ссылка *Фильтр* доступна над списком групп пользователей. Если вы нажмёте на неё, станет доступен фильтр, в котором можно отфильтровать группы пользователей по имени и статусу.



2 Роли пользователей

Обзор

В разделе *Пользователи* > *Роли пользователей* вы можете создавать роли пользователей.

Роли пользователей позволяют создавать детализированные права доступа на основе изначально выбранного типа пользователя (*User*, *Admin*, *Super admin*).

При выборе типа пользователя предоставляются (по умолчанию отмечаются) все доступные права для этого типа пользователя.

Права доступа можно только отзывать из набора, доступного для типа пользователя; их нельзя расширить сверх того, что доступно для типа пользователя.

Флажки для недоступных прав доступа отображаются серым цветом; пользователи не смогут получить доступ к элементу даже при вводе прямого URL этого элемента в браузере.

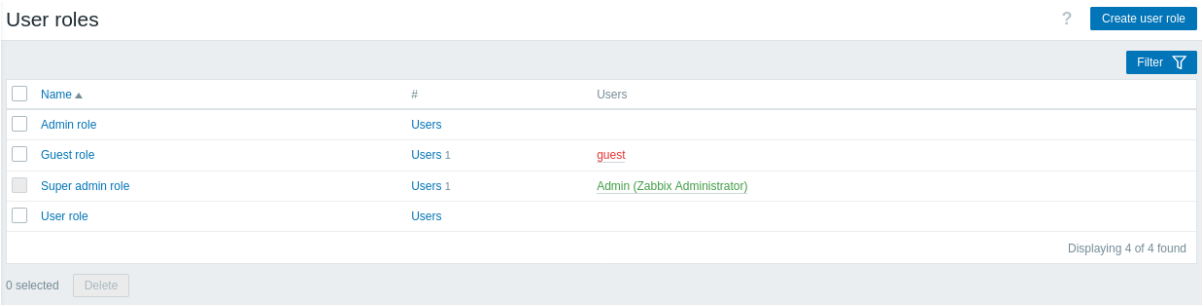
Note:
Ограничение доступа к некоторым элементам пользовательского интерфейса лишь предотвращает открытие этой страницы — оно не устраняет возможность доступа к базовым данным в других частях интерфейса.

Роли пользователей могут быть назначены системным пользователям. Каждому пользователю может быть назначена только одна роль.

Роли пользователей по умолчанию

По умолчанию Zabbix настроен с четырьмя ролями пользователей, для которых заранее определён набор прав:

- Роль Guest
- Роль User
- Роль Admin
- Роль Super admin

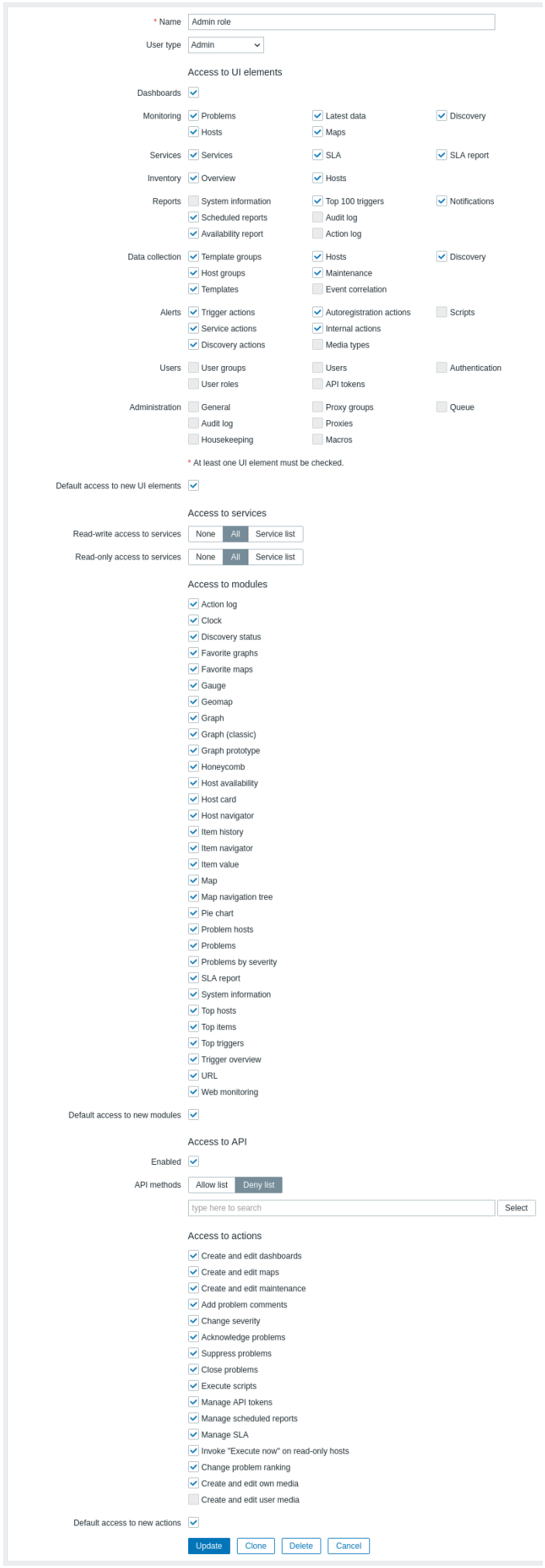


Они основаны на основных типах пользователей в Zabbix. Отображается список всех пользователей, которым назначена соответствующая роль. Пользователи, входящие в отключённые группы, выделены красным цветом. *Роль Guest* — это роль типа пользователя с единственными правами на просмотр некоторых разделов веб-интерфейса.

Note:
Роль *Super admin* по умолчанию нельзя изменить или удалить, поскольку в Zabbix должен существовать как минимум один пользователь Super admin с неограниченными привилегиями. Пользователи типа *Super admin* могут изменять настройки собственной роли, но не тип пользователя.

Настройка

Чтобы создать новую роль, нажмите кнопку *Создать роль пользователя* в правом верхнем углу. Чтобы обновить существующую роль, нажмите на имя роли, чтобы открыть форму настройки.


* Name
Admin role
User type
Admin
Access to UI elements
Dashboards
Monitoring
Problems
Latest data
Discovery
Hosts
Maps
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 correlation
Alerts
Trigger actions
Autoregistration actions
Scripts
Service actions
Internal actions
Discovery actions
Media types
Users
User groups
Users
Authentication
User roles
API tokens
Administration
General
Proxy groups
Queue
Audit log
Proxies
Housekeeping
Macros
* At least one UI element must be checked.
Default access to new UI elements
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 history
Item navigator
Item value
Map
Map navigation tree
Pie chart
Problem hosts
Problems
Problems by severity
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
type here to search
Select
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
Default access to new actions
Update
Clone
Delete
Cancel

Отображаются доступные разрешения. Чтобы отозвать определенное разрешение, снимите соответствующий флажок.

Ниже описаны доступные разрешения вместе со значениями по умолчанию для каждой предварительно настроенной роли пользователя в Zabbix.

Разрешения по умолчанию

Доступ к элементам веб-интерфейса

Доступ по умолчанию к разделам меню зависит от типа пользователя. Подробности см. на странице Permissions здесь.

Доступ к другим параметрам

Параметр	**Описание**	**Роли пользователей по умолчанию**			
		Роль Super admin	*Роль Admin*	*Роль User*	*Роль Guest*
Доступ по умолчанию к новым элементам веб-интерфейса	Этот параметр определяет, как новые разделы меню будут доступны после обновления Zabbix. Существующие разделы меню модулей остаются без изменений.	Да	Да	Да	Да
Доступ к услугам					
Доступ на чтение-запись к услугам	Выберите доступ на чтение-запись к услугам: **Нет** — доступа нет совсем **Все** — доступ ко всем услугам предоставляется на чтение и запись **Список услуг** — выберите услуги, к которым будет предоставлен доступ на чтение и запись Если доступ на чтение-запись предоставлен, он имеет приоритет над настройками доступа только на чтение и динамически наследуется дочерними услугами.	Все	Все	Нет	Нет
Доступ на чтение-запись к услугам с тегом	Укажите имя тега и, при необходимости, значение, чтобы дополнительно предоставить доступ на чтение-запись к услугам, соответствующим тегу. Этот параметр доступен, если в параметре *Доступ на чтение-запись к услугам* выбран вариант 'Список услуг'. Если доступ на чтение-запись предоставлен, он имеет приоритет над настройками доступа только на чтение и динамически наследуется дочерними услугами.				
Доступ только на чтение к услугам	Выберите доступ только на чтение к услугам: **Нет** — доступа нет совсем **Все** — доступ ко всем услугам предоставляется только на чтение **Список услуг** — выберите услуги, к которым будет предоставлен доступ только на чтение Доступ только на чтение не имеет приоритета над доступом на чтение-запись и динамически наследуется дочерними услугами.			Все	Все

Доступ только на чтение к услугам с тегом	Укажите имя тега и, при необходимости, значение, чтобы дополнительно предоставить доступ только на чтение к услугам, соответствующим тегу. Этот параметр доступен, если в параметре *Доступ только на чтение к услугам* выбран вариант 'Список услуг'. Доступ только на чтение не имеет приоритета над доступом на чтение-запись и динамически наследуется дочерними услугами.				
Доступ к модулям					
<Имя модуля>	Разрешить/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 новые модули и виджеты будут доступны после обновления Zabbix. Он также применяется к модулям и виджетам, добавленным в разделе *Administration > General > Modules*.				
Доступ к API					
Включено Методы API	Включить/отключить доступ к API. Выберите *Список разрешений*, чтобы разрешить, или *Список запретов*, чтобы запретить методы API, указанные в поле поиска. Обратите внимание, что невозможно разрешить одни методы API и запретить другие. В поле поиска начните вводить имя метода, затем выберите метод из списка автодополнения. Вы также можете нажать кнопку Select и выбрать методы из полного списка, доступного для данного типа пользователя. Обратите внимание, что если определенное действие в блоке Access to actions снято, пользователи не смогут использовать методы API, связанные с этим действием. Поддерживаются подстановочные знаки. Примеры: `dashboard.*` (все методы API-сервиса 'dashboard.') `*` (любой метод), `*.export` (методы с именем '.export' из всех API-сервисов). Если методы не указаны, правило *Список разрешений/запретов* будет проигнорировано.	Да	Да	Да	Нет
Доступ к действиям					
Создание и редактирование панелей мониторинга	Снятие этого флажка также отзовет права на использование методов API `.create`, `.update` и `.delete` для соответствующих элементов.	Да	Да	Да	Нет
Создание и редактирование карт					
Создание и редактирование обслуживания				Нет	

Добавление комментариев к проблемам Изменение важности Подтверждение проблем Подавление проблем Закрытие проблем	Снятие этого флажка также отзовет права на выполнение соответствующего действия через метод API `event.acknowledge`.		Да
Выполнение скриптов	Снятие этого флажка также отзовет права на использование метода API `script.execute`.		
Управление API-токенами	Снятие этого флажка также отзовет права на использование всех методов API `token.*`.		
Управление запланированными отчетами	Снятие этого флажка также отзовет права на использование всех методов API `report.*`.		Нет
Управление SLA	Включить/отключить права на управление SLA.		
Вызов ”Выполнить сейчас” для узлов сети с доступом только на чтение	Разрешить использование параметра ”Выполнить сейчас” в последних данных для элементов данных узлов сети с доступом только на чтение.		Да
Изменение ранжирования проблем	Разрешить изменять ранжирование проблем с причины на симптом и наоборот.		
Создание и редактирование собственных способов оповещения	Разрешить создавать/редактировать собственные способы оповещения.		
Создание и редактирование способов оповещения пользователей	Разрешить создавать/редактировать способы оповещения для пользователей. Этот параметр доступен только для пользователей Super admin.	Нет	Нет
Доступ по умолчанию к новым действиям	Этот параметр определяет, как новые действия будут доступны после обновления Zabbix.	Да	Да

См. также:

- Настройка пользователя

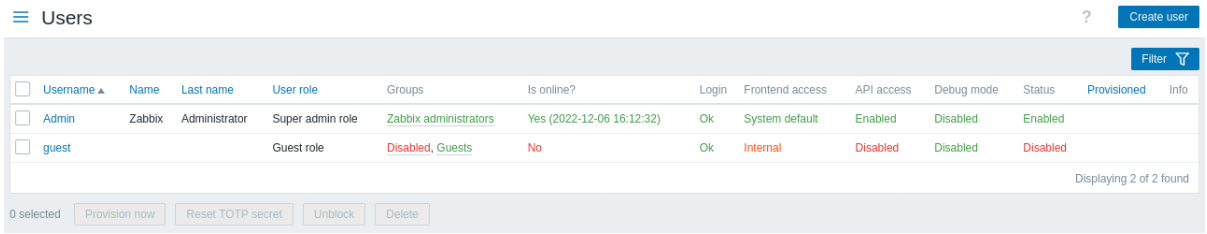
3 Пользователи

Обзор

В разделе *Пользователи* → *Пользователи* выполняется управление пользователями системы.

Пользователи

Отображается список существующих пользователей с их данными.



Отображаемые данные:

Столбец	Описание
Имя пользователя	Имя пользователя для входа в Zabbix. Нажатие на имя пользователя открывает форму настройки пользователя.
Имя	Имя пользователя.
Фамилия	Фамилия пользователя.
Роль пользователя	Отображается роль пользователя.
Группы	Перечислены группы, в которые входит пользователь. Нажатие на имя группы пользователей открывает форму настройки группы пользователей. Отключенные группы отображаются красным цветом.
В сети?	Отображается статус присутствия пользователя в сети — *Да* или *Нет*. В скобках отображается время последней активности пользователя.
Вход	Отображается статус входа пользователя — *Ok* или *Blocked*. Пользователь может быть временно заблокирован при превышении количества неудачных попыток входа, заданного в разделе *Администрирование → Общие → Прочее* (по умолчанию пять). Нажав на *Blocked*, вы можете разблокировать пользователя.
Доступ к веб-интерфейсу	Отображается уровень доступа к веб-интерфейсу — *System default*, *Internal*, *LDAP* или *Disabled* — в зависимости от настройки, заданной для всей группы пользователей.
Доступ к API	Отображается статус доступа к API — *Enabled* или *Disabled* — в зависимости от настройки, заданной для роли пользователя.
Режим отладки	Отображается статус режима отладки — *Enabled* или *Disabled* — в зависимости от настройки, заданной для всей группы пользователей.
Статус	Отображается статус пользователя — *Enabled* или *Disabled* — в зависимости от настройки, заданной для всей группы пользователей.
Provisioned	Отображается дата последнего provision пользователя. Используется для пользователей, созданных с помощью JIT provision из LDAP/SAML.
Информация	Отображается информация об ошибках. Желтое предупреждение отображается для пользователей без групп пользователей. Красное предупреждение отображается для пользователей без ролей, а также для пользователей без ролей и групп пользователей.

Чтобы настроить нового пользователя, нажмите кнопку *Создать пользователя* в правом верхнем углу.

Параметры массового редактирования

Кнопки под списком предоставляют несколько вариантов массового редактирования:

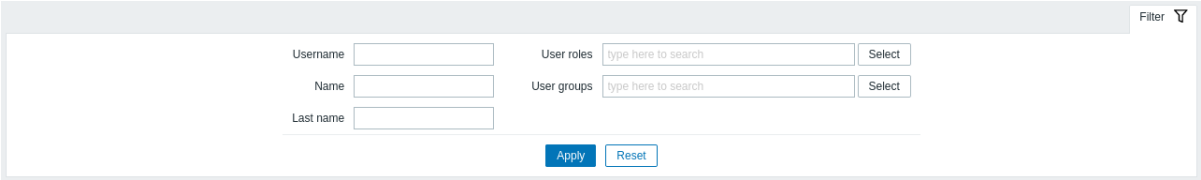
- *Provision now* — обновить информацию о пользователе из LDAP (этот параметр доступен, только если выбран пользователь LDAP)
- *Reset TOTP secret* — сбросить TOTP-секреты пользователя для всех методов TOTP и удалить пользовательский сеанс (этот параметр доступен, только если включена MFA; для пользователей без TOTP-секретов их сеанс удалён не будет)
- *Unblock* — повторно предоставить заблокированным пользователям доступ к системе
- *Delete* — удалить пользователей

Чтобы использовать эти параметры, установите флажки напротив соответствующих пользователей, затем нажмите нужную кнопку.

Использование фильтра

Вы можете использовать фильтр, чтобы отображать только тех пользователей, которые вас интересуют. Для повышения производительности поиска данные ищутся с неразрешёнными макросами.

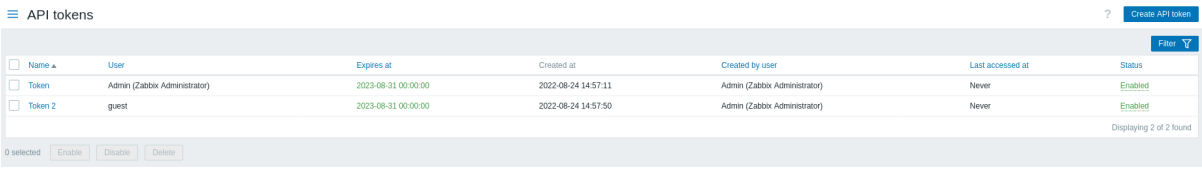
Ссылка *Фильтр* доступна над списком пользователей. Если вы нажмёте на неё, станет доступен фильтр, в котором можно отфильтровать пользователей по имени пользователя, имени, фамилии, роли пользователя и группе пользователей.



4 API-токены

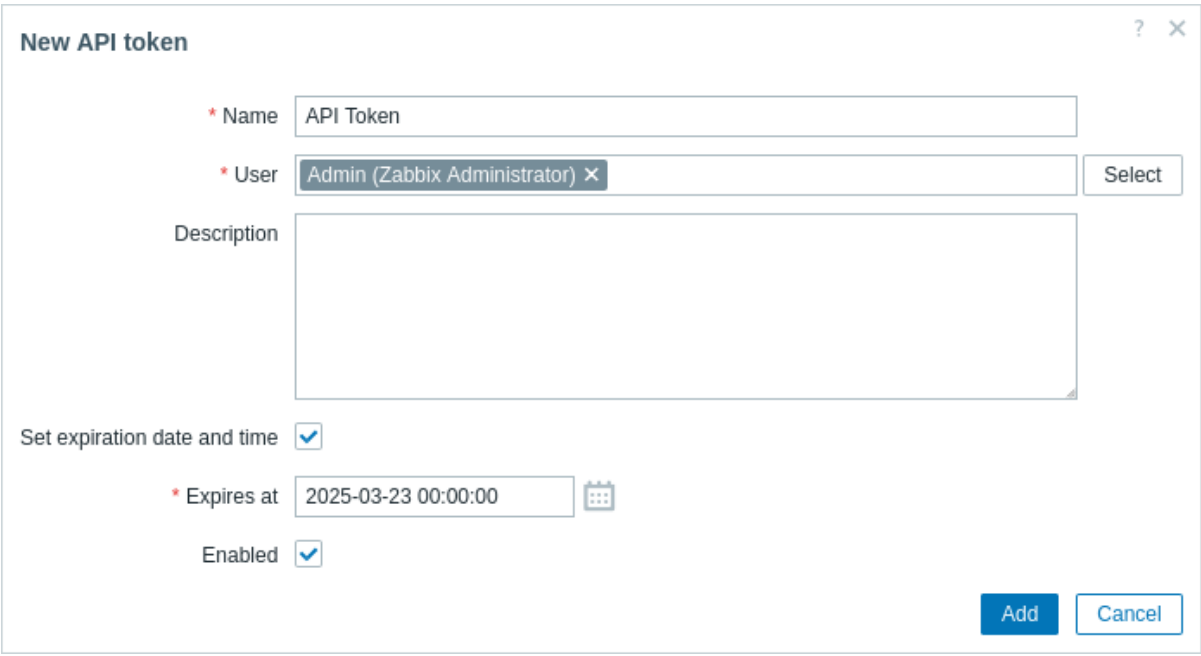
Обзор

Этот раздел позволяет создавать API-токены и управлять ими.



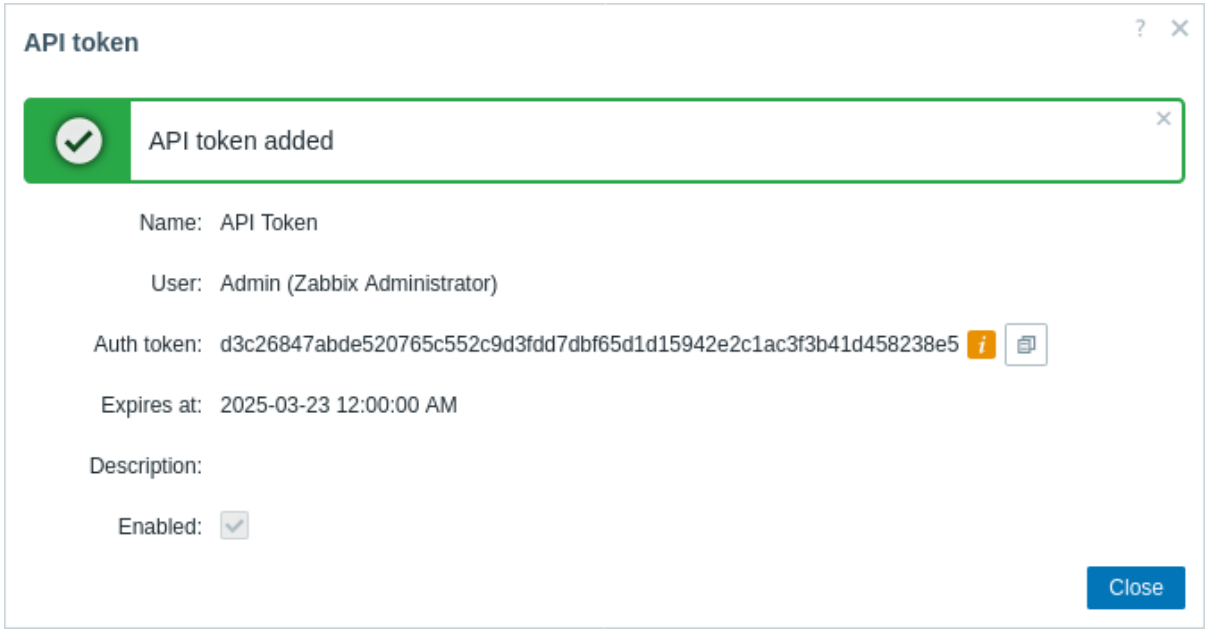
Вы можете фильтровать API-токены по имени, пользователям, которым назначены токены, дате истечения срока действия, пользователям, создавшим токены, или статусу (включен/отключен). Нажмите на статус токена в списке, чтобы быстро включить/отключить токен. Вы также можете массово включать/отключать токены, выбрав их в списке, а затем 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.
User	Пользователь, которому должен быть назначен токен. Чтобы быстро выбрать пользователя, начните вводить имя пользователя, имя или фамилию, затем выберите нужного пользователя из списка автодополнения. Либо вы можете нажать кнопку *Select* и выбрать пользователя из полного списка пользователей. Токен может быть назначен только одному пользователю.
Description	Необязательное описание токена.
Set expiration date and time	Снимите этот флажок, если токен не должен иметь даты истечения срока действия.
Expiry date	Нажмите на значок календаря, чтобы выбрать дату истечения срока 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* позже нельзя будет просмотреть повторно. Оно доступно только сразу после создания токена. Если вы потеряете сохраненный токен, вам придется сгенерировать его заново, и при этом будет создана новая строка авторизации.

Нажмите на имя токена, чтобы изменить имя, описание, настройки даты истечения срока действия или статус токена. Обратите внимание, что изменить пользователя, которому назначен токен, невозможно. Нажмите кнопку *Update*, чтобы сохранить изменения. Если токен был утерян или скомпрометирован, вы можете нажать кнопку *Regenerate*, чтобы сгенерировать новое значение токена. Появится диалоговое окно подтверждения с запросом подтвердить эту операцию, поскольку после продолжения ранее сгенерированный токен станет недействительным.

Пользователи без доступа к разделу меню *Administration* могут просматривать и изменять сведения о назначенных им токенах только в *Профиль пользователя* → *API tokens* разделе, если разрешение *Manage API tokens* включено в правах их роли пользователя.

5 Аутентификация

Обзор

Раздел *Пользователи* → *Аутентификация* позволяет указать метод аутентификации пользователей для Zabbix и внутренние требования к паролю.

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 LDAP для всей системы. 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 тонко настроить на уровне группы пользователей. Даже если аутентификация LDAP задана глобально, некоторые группы пользователей всё равно могут проходить аутентификацию через Zabbix. Для таких групп параметр доступ к веб-интерфейсу должен быть установлен в значение Internal.

Также можно включить аутентификацию LDAP только для определённых групп пользователей, если глобально используется внутренняя аутентификация. В этом случае сведения для аутентификации LDAP можно указать и использовать для конкретных групп пользователей, у которых доступ к веб-интерфейсу затем должен быть установлен в значение LDAP. Если пользователь включён хотя бы в одну группу пользователей с аутентификацией LDAP, этот пользователь не сможет использовать метод внутренней аутентификации.

Методы аутентификации HTTP, SAML 2.0 и MFA могут использоваться в дополнение к методу аутентификации по умолчанию.

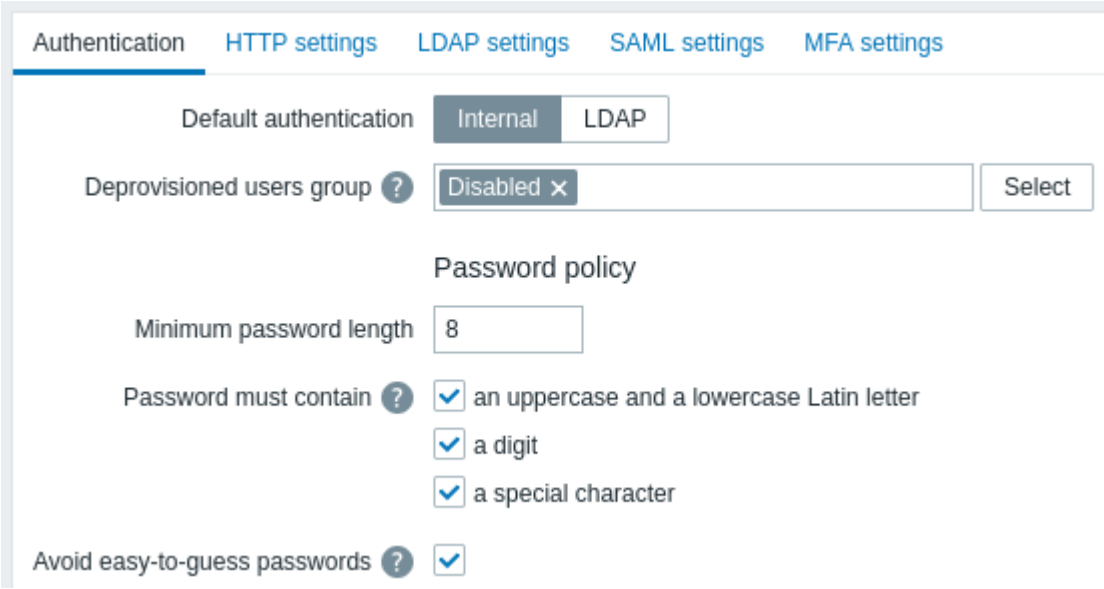
Zabbix поддерживает just-in-time (JIT) provision, что позволяет создавать учётные записи пользователей в Zabbix при первой аутентификации внешнего пользователя и выполнять provision этих учётных записей. JIT provision поддерживается для LDAP и SAML.

См. также:

- Аутентификация HTTP
- Аутентификация LDAP
- Аутентификация SAML
- Аутентификация MFA

Конфигурация

Вкладка *Аутентификация* позволяет задать метод аутентификации по умолчанию, указать группу для девыделенных пользователей и установить требования к сложности паролей для пользователей Zabbix.



Параметры конфигурации:

Параметр	Описание
Аутентификация по умолчанию	Выберите метод аутентификации по умолчанию для Zabbix — *Internal* или *LDAP*.
Группа девыделенных пользователей	Укажите группу пользователей для девыделенных пользователей. Этот параметр требуется только для JIT-подготовки, в отношении пользователей, которые были созданы в Zabbix из систем LDAP или SAML, но больше не нуждаются в подготовке. Необходимо указать отключенную группу пользователей.
Минимальная длина пароля	По умолчанию минимальная длина пароля составляет 8. Поддерживаемый диапазон: 1-70. Обратите внимание, что пароли длиннее 72 символов будут усечены.
Пароль должен содержать	Отметьте один или несколько флажков, чтобы потребовать использование указанных символов в пароле: - заглавную и строчную латинскую букву - цифру - специальный символ Наведите указатель на знак вопроса, чтобы увидеть подсказку со списком символов для каждого варианта.

Параметр	Описание
Избегать легко угадываемых паролей	Если отмечено, пароль будет проверяться на соответствие следующим требованиям: - не должен содержать имя, фамилию или имя пользователя - не должен быть одним из распространённых или контекстно-зависимых паролей. Список распространённых и контекстно-зависимых паролей формируется автоматически 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, включённые в этот список, так как такие пароли считаются слабыми из-за их широкого использования.

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*.

1 HTTP

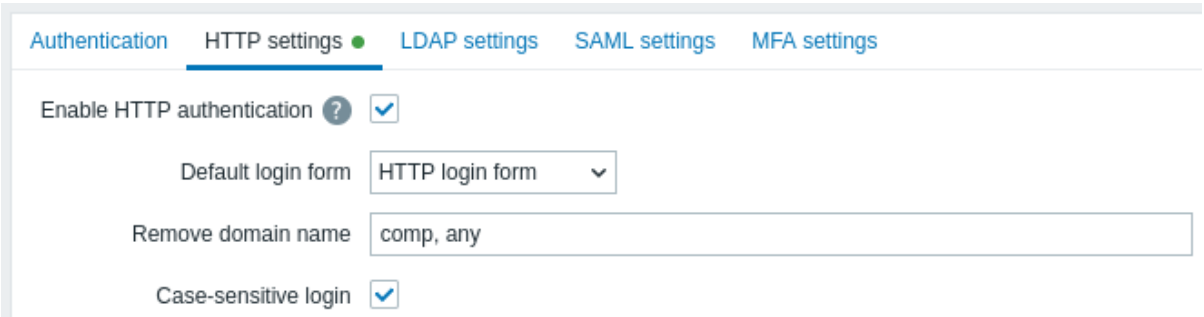
Обзор

Аутентификация на основе HTTP или веб-сервера (например: BasicAuthentication, NTLM/Kerberos) может использоваться для проверки имен пользователей и паролей. Обратите внимание, что пользователь также должен существовать в Zabbix, однако его пароль Zabbix использоваться не будет.

Attention:
Будьте осторожны! Убедитесь, что аутентификация веб-сервера настроена и работает должным образом, прежде чем включать ее.

HTTP-аутентификацию можно отключить в веб-интерфейсе, настроив соответствующую опцию на вкладке *Настройки HTTP* в разделе *Пользователи > Аутентификация*. Когда HTTP-аутентификация отключена, вкладка с параметрами HTTP-аутентификации не будет отображаться в веб-интерфейсе. Обратите внимание, что повторная установка веб-интерфейса (запуск setup.php) приведет к сбросу настроек аутентификации, включая конфигурацию HTTP-аутентификации.

Конфигурация



Параметры конфигурации:

Параметр	Описание
Включить HTTP-аутентификацию	Установите флажок, чтобы включить HTTP-аутентификацию. При наведении курсора на появится всплывающая подсказка с предупреждением о том, что в случае аутентификации веб-сервера все пользователи (даже если доступ к веб-интерфейсу установлен в LDAP/Internal) будут аутентифицироваться веб-сервером, а не Zabbix.

Параметр	Описание
Форма входа по умолчанию	Укажите, куда перенаправлять неаутентифицированных пользователей: **Форма входа Zabbix** — стандартная страница входа Zabbix. **HTTP-форма входа** — HTTP-страница входа. Рекомендуется включать аутентификацию на основе веб-сервера только для страницы `index_http.php`. Если для *Форма входа по умолчанию* установлено значение «HTTP-страница входа», пользователь войдёт автоматически, если модуль аутентификации веб-сервера установит корректное имя пользователя в переменной `$_SERVER`. Поддерживаются следующие ключи `$_SERVER`: `PHP_AUTH_USER`, `REMOTE_USER`, `AUTH_USER`.
Удалить доменное имя	Список доменных имён, разделённых запятыми, которые должны быть удалены из имени пользователя. Например, `comp,any` — если имя пользователя имеет вид 'Admin@any', 'comp\Admin', вход будет выполнен как для 'Admin'; если имя пользователя имеет вид 'notacompany\Admin', во входе будет отказано.
Учитывать регистр при входе	Снимите флажок, чтобы отключить учёт регистра в именах пользователей при входе (по умолчанию включено). Отключение учёта регистра позволяет, например, войти как "admin", даже если пользователь Zabbix указан как "Admin" или "ADMIN". Обратите внимание: если учёт регистра при входе отключён и существует несколько пользователей Zabbix с похожими именами (например, Admin и admin), вход для таких пользователей всегда будет отклонён со следующим сообщением об ошибке: "Authentication failed: supplied credentials are not unique."

Note:
Для внутренних пользователей, которые не могут войти с использованием HTTP-учётных данных (если HTTP-форма входа установлена по умолчанию), что приводит к ошибке 401, можно добавить строку `ErrorDocument 401 /index.php?form=default` в директивы базовой аутентификации, чтобы перенаправить пользователя на обычную форму входа Zabbix.

2 LDAP

Обзор

Внешняя LDAP-аутентификация может использоваться для проверки имен пользователей и паролей.

LDAP-аутентификация 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 Zabbix использоваться не будет. Если аутентификация выполнена успешно, Zabbix сопоставит локальное имя пользователя с атрибутом имени пользователя, возвращаемым LDAP.

JIT-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, когда пользователь входит в Zabbix в первый раз.

Когда пользователь LDAP вводит свой LDAP-логин и пароль, Zabbix проверяет *LDAP-сервер по умолчанию*, существует ли такой пользователь. Если пользователь существует и еще не имеет учетной записи в Zabbix, в Zabbix создается новый пользователь, и пользователь получает возможность войти в систему.

Пользователь, созданный с помощью JIT-подготовки, связывается с LDAP-сервером (каталогом), который установлен по умолчанию на момент создания. Последующее изменение LDAP-сервера по умолчанию не изменяет и не обновляет LDAP-сервер, связанный с пользователями, которые уже были подготовлены.

Attention:
Если JIT-подготовка включена, на вкладке *Аутентификация* необходимо указать группу пользователей для пользователей с отмененной подготовкой.

JIT-подготовка также позволяет обновлять подготовленные учетные записи пользователей на основе изменений в LDAP. Например, если пользователь перемещен из одной группы LDAP в другую, пользователь также будет перемещен из одной группы в другую в Zabbix; если пользователь удален из группы LDAP, пользователь также будет удален из группы в Zabbix и, если он не принадлежит ни к какой другой группе, добавлен в группу пользователей для пользователей с отмененной подготовкой. Подготовленные учетные записи пользователей обновляются на основе настроенного периода подготовки или когда пользователь входит в Zabbix.

Обратите внимание, что фоновая подготовка выполняется веб-интерфейсом Zabbix, когда пользователь взаимодействует с ним или по крайней мере имеет открытую страницу веб-интерфейса в браузере. Выделенных фоновых процессов для подготовки пользователей нет.

LDAP поддерживает три способа привязки к каталогу для аутентификации и поиска:

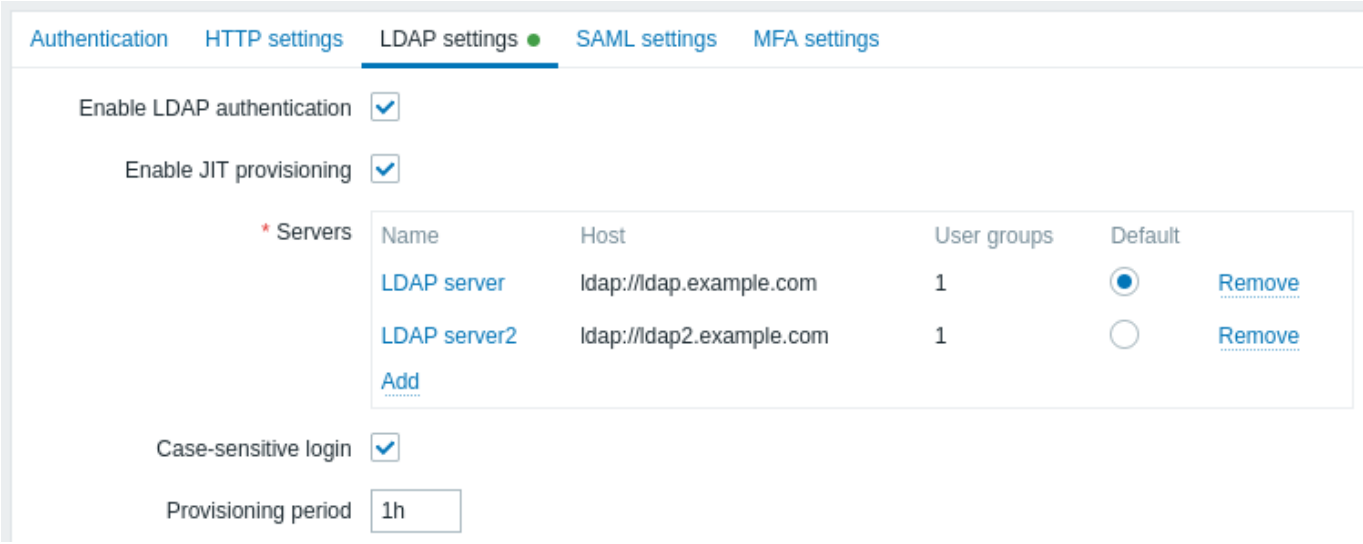
- Анонимная привязка — *Bind DN* / *Bind password* не указываются, и LDAP-сервер разрешает анонимные запросы.
- Выделенный пользователь привязки (служебная учетная запись) — в *Bind DN* / *Bind password* указывается конкретная учетная запись LDAP, и она используется Zabbix для поиска и подготовки. Это рекомендуемый и наиболее гибкий вариант, поскольку Zabbix может выполнять поиск и фоновую подготовку без учетных данных конечного пользователя.
- Прямая привязка пользователя — Zabbix выполняет привязку с использованием учетных данных, которые пользователь вводит при входе в систему (*Bind DN* / *Bind password* не настроены); это настраивается путем включения заполнителя, такого как `uid=%{user}`, в Base DN. В этом режиме Zabbix имеет доступ к паролю пользователя только во время интерактивного входа в систему. В результате действия подготовки, требующие аутентификации в LDAP вне сеанса входа пользователя (например, использование кнопки *Provision now* или выполнение фоновой подготовки, когда пользователь не вошел в систему активно), не могут пройти аутентификацию и поэтому не будут работать. Подготовка и обновления для прямой привязки пользователя происходят только в момент входа пользователя в систему.

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 появляется возможность выбрать требуемый LDAP-сервер для соответствующей группы пользователей.

Если пользователь состоит в нескольких группах пользователей и относится к нескольким LDAP-серверам, для аутентификации будет использоваться первый сервер в списке LDAP-серверов, отсортированном по имени в порядке возрастания.

Конфигурация

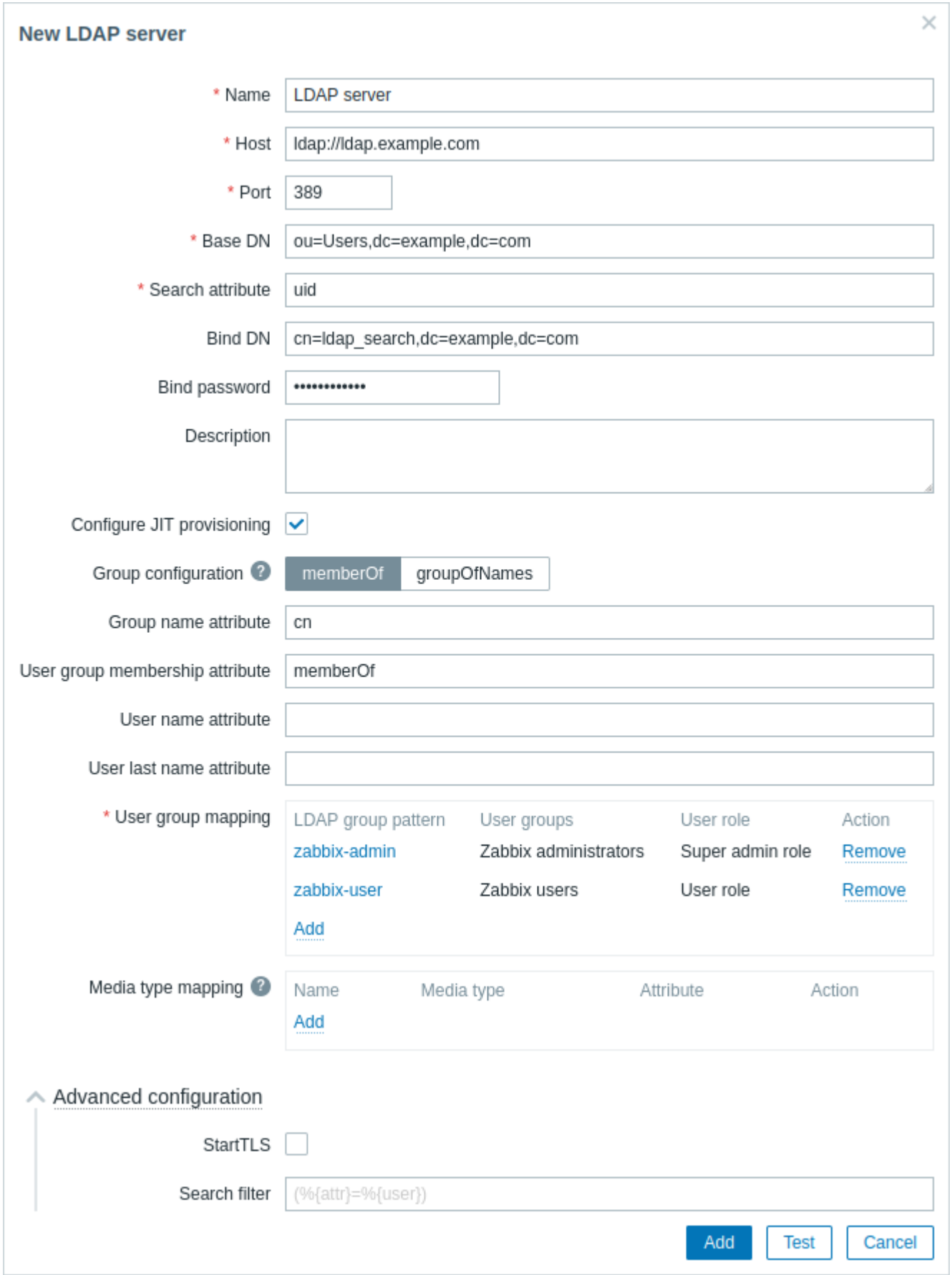


Параметры конфигурации:

Параметр	Описание
Включить LDAP-аутентификацию	Установите флажок, чтобы включить LDAP-аутентификацию.
Включить JIT-подготовку	Установите флажок, чтобы включить JIT-подготовку.
Серверы	Нажмите *Добавить*, чтобы настроить LDAP-сервер (см. Настройка LDAP-сервера ниже).

Параметр	Описание
Логин с учетом регистра	Снимите флажок, чтобы отключить учет регистра при входе для имен пользователей (по умолчанию включено). Отключение учета регистра при входе позволяет, например, войти как ”admin”, даже если пользователь Zabbix — ”Admin” или ”ADMIN”. Обратите внимание, что если учет регистра при входе отключен и существует несколько пользователей Zabbix с похожими именами (например, Admin и admin), вход для таких пользователей всегда будет отклонен со следующим сообщением об ошибке: ”Authentication failed: supplied credentials are not unique.”
Период подготовки	Установите период подготовки, то есть как часто для вошедшего пользователя будет выполняться подготовка во время работы с веб-интерфейсом.

Настройка LDAP-сервера



Параметры настройки LDAP-сервера:

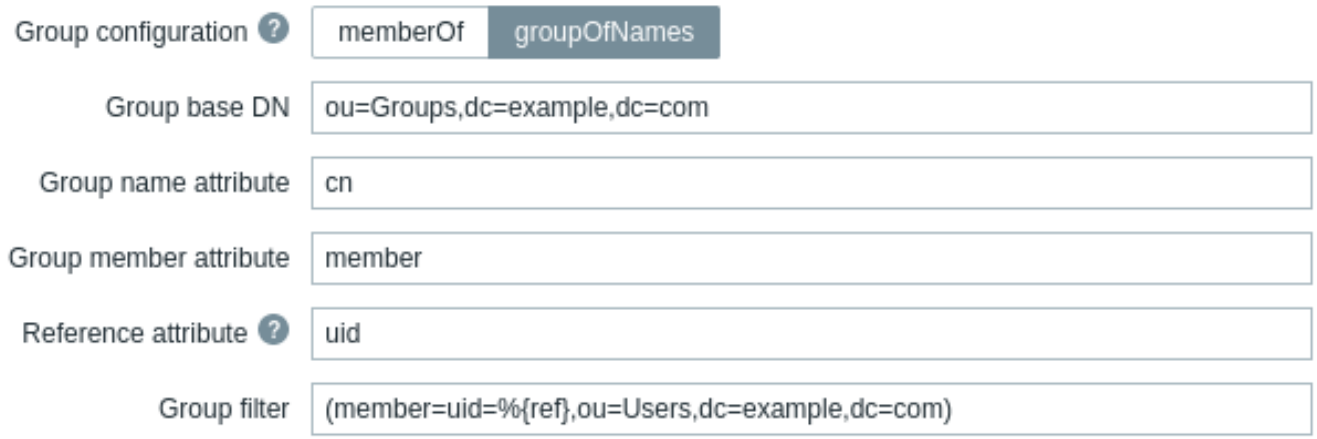
Параметр	Описание
Name	Имя LDAP-сервера в конфигурации Zabbix.

Параметр	Описание
Host	Имя узла, 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 узла. Пример: ldaps://ldap.example.com С OpenLDAP 2.x.x и новее можно использовать полный LDAP URI в формате ldap://hostname:port или ldaps://hostname:port.
Port	Порт LDAP-сервера. По умолчанию — 389. Для защищенного LDAP-соединения обычно используется порт 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 неавторизованным пользователям (информацию о пользователях, компьютерах, серверах, группах, службах и т. д.). По соображениям безопасности отключайте анонимную привязку на LDAP-узлах и используйте аутентифицированный доступ.
Bind password	LDAP-пароль учетной записи для привязки и поиска на LDAP-сервере.
Description	Описание LDAP-сервера.
Configure JIT provisioning	Установите этот флажок, чтобы отобразить параметры, связанные с JIT-подготовкой пользователей.
Group configuration	Выберите метод настройки групп: **memberOf** — путем поиска пользователей и атрибута их членства в группах **groupOfNames** — путем поиска групп через атрибут member Обратите внимание, что memberOf предпочтительнее, так как работает быстрее; используйте groupOfNames, если ваш LDAP-сервер не поддерживает `memberOf` или требуется фильтрация групп.
Group name attribute	Укажите атрибут для получения имени группы из 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 информацию о группах, в которые входит пользователь (например, `memberOf`). Например, атрибут memberOf может содержать такую информацию: `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 LDAP-группы с группой пользователей Zabbix и ролью пользователя. Это необходимо, чтобы определить, какую группу/роль получит подготовленный пользователь в Zabbix. Нажмите *Add*, чтобы добавить сопоставление. Поле *LDAP group pattern* поддерживает подстановочные знаки. Имя группы должно совпадать с существующей группой. Если LDAP-пользователь соответствует нескольким группам пользователей Zabbix, пользователь станет участником всех этих групп. Если пользователь соответствует нескольким ролям пользователей Zabbix, он получит ту из них, у которой самый высокий уровень прав.
Media type mapping	Сопоставьте LDAP-атрибуты пользователя способам оповещения Zabbix (например, email) для отправки уведомлений (значение атрибута используется в поле *Send to* пользовательского способа оповещения).
Advanced configuration	Нажмите на метку *Advanced configuration*, чтобы отобразить дополнительные параметры настройки (см. ниже).

Параметр	Описание
StartTLS	Установите флажок, чтобы использовать операцию StartTLS при подключении к LDAP-серверу. Соединение завершится ошибкой, если сервер не поддерживает StartTLS. StartTLS нельзя использовать с серверами, работающими по протоколу *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 регистрозависимый поиск в нечувствительной к регистру среде LDAP или Microsoft Active Directory, 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 к учетным данным пользователя только во время интерактивного входа. Поэтому задачи подготовки, выполняемые 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 выделенного пользователя для привязки (служебную учетную запись), если вам нужно, чтобы подготовка и LDAP-поиск работали из веб-интерфейса без необходимости входа конечного пользователя в систему.

Следующие поля относятся только к методу *Group configuration* "groupOfNames":



Параметр	Описание
Group base DN	Базовый путь к группам на LDAP-сервере.
Group name attribute	Укажите атрибут для получения имени группы по указанному базовому пути к группам. Имя группы необходимо для сопоставления групп пользователей.
Group member attribute	Укажите атрибут, содержащий информацию о членах 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, в которую входит пользователь. Например, `(member=uid=%{ref},ou=Users,dc=example,dc=com)` будет соответствовать "User1", если атрибут member группы имеет значение `uid=User1,ou=Users,dc=example,dc=com`, и вернет группу, в которую входит "User1".

Warning:
В случае проблем с сертификатами, чтобы заработало защищенное LDAP-соединение (ldaps), 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 учетную запись LDAP (*Bind DN*) с минимальными правами для выполнения привязки и поиска на LDAP-сервере 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*, когда пользователь меняет свой собственный пароль на LDAP-сервере.
В таблице выше это учетная запись *ldap_search*.

Проверка доступа

Кнопка *Test* позволяет проверить доступ пользователя:

Параметр	Описание
Login	Имя пользователя LDAP для проверки (предварительно заполнено текущим именем пользователя из веб-интерфейса Zabbix). Это имя пользователя должно существовать на LDAP-сервере. Zabbix не активирует LDAP-аутентификацию, если не сможет аутентифицировать тестового пользователя.
User password	Пароль пользователя LDAP для проверки.

3 SAML

Обзор

Аутентификация SAML 2.0 может использоваться для входа в Zabbix.

Если настроен только вход через SAML, пользователь также должен существовать в Zabbix, однако его пароль Zabbix использоваться не будет. Если аутентификация выполнена успешно, Zabbix сопоставит локальное имя пользователя с атрибутом имени пользователя, возвращаемым SAML.

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, когда пользователь впервые входит в Zabbix.

Attention:
Если JIT-подготовка включена, на вкладке *Authentication* необходимо указать группу пользователей для пользователей с отмененной подготовкой.

Помимо JIT-подготовки, также можно включить и настроить SCIM (System for Cross-domain Identity Management) — *непрерывное* управление учетными записями пользователей для тех пользователей, которые были созданы с помощью подготовки пользователей. Для SCIM-подготовки требуется Zabbix API token (с правами Super admin) для аутентификации в Zabbix.

Например, если пользователь перемещен из одной группы SAML в другую, пользователь также будет перемещен из одной группы в другую в Zabbix; если пользователь удален из группы SAML, пользователь также будет удален из группы в Zabbix и, если он не принадлежит ни к какой другой группе, добавлен в группу пользователей для пользователей с отмененной подготовкой.

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 ему будут добавлены только корректные способы оповещения.

Если SCIM не включен, пользователь SAML будет подготовлен (а позже обновлен) в момент входа пользователя в Zabbix.

Note:
Если включена SAML-аутентификация, пользователи смогут выбирать между локальным входом и входом через единый вход SAML. Если используется JIT-подготовка, тогда возможен только единый вход.

Настройка поставщика удостоверений

Для работы с Zabbix необходимо настроить поставщика удостоверений SAML (onelogin.com, auth0.com, okta.com и т. д.) 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:
Необходимо установить php-openssl, если вы хотите использовать SAML-аутентификацию в веб-интерфейсе.

Чтобы использовать SAML-аутентификацию, Zabbix должен быть настроен следующим образом:

1. Предоставьте закрытый ключ SP и сертификаты. Для активации SAML через веб-интерфейс необходимо предоставить сертификат IdP (вставить текст или выбрать файл). Сертификат SP и закрытый ключ SP необходимо предоставить, если выбраны параметры *Sign* или *Encrypt*.

Если в zabbix.conf.php задано `$SSO['CERT_STORAGE'] = 'database'`, вы можете вставить текст сертификата или загрузить файл сертификата в веб-интерфейсе во время настройки SAML — файлы в файловой системе не требуются.

Если в zabbix.conf.php задано `$SSO['CERT_STORAGE'] = 'file'`, сертификат должен быть доступен в файловой системе (по умолчанию в ui/conf/certs или по пути, настроенному в zabbix.conf.php), и веб-интерфейс не будет хранить сертификаты в базе данных. Обратите внимание, что если `$SSO['CERT_STORAGE']` не задан или закомментирован, предполагается хранение в файлах.

По умолчанию Zabbix будет искать 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:
Материалы сертификатов могут быть импортированы через веб-интерфейс путем вставки текста или загрузки файлов. Хотя для удобства в некоторых средах допускается импорт без шифрования, настоятельно рекомендуется использовать зашифрованную передачу (например, включить HTTPS/TLS для веб-интерфейса Zabbix). Если сертификаты/ключи SAML хранятся в базе данных, включите TLS/SSL для подключений к базе данных, чтобы защитить трафик между приложением и базой данных.
 Шифруйте резервные копии базы данных, которые могут содержать сертификаты SAML или закрытые ключи, и ограничьте доступ к файлам резервных копий и к таблицам базы данных, в которых хранятся материалы SAML. Ограничьте привилегии пользователя базы данных до минимально необходимых.

2. Все наиболее важные настройки можно задать в веб-интерфейсе Zabbix. Однако дополнительные настройки можно указать в файле конфигурации.



Параметры конфигурации, доступные в веб-интерфейсе Zabbix:

Parameter	Description
Enable SAML authentication	Установите этот флажок, чтобы включить SAML-аутентификацию.
Enable JIT provisioning	Установите этот флажок, чтобы включить JIT-подготовку пользователей.
IDP entity ID	Уникальный идентификатор сущности в поставщике удостоверений SAML.
SSO service URL	URL, на который пользователи будут перенаправлены при входе в систему.
SLO service URL	URL, на который пользователи будут перенаправлены при выходе из системы. Если оставить пустым, служба SLO использоваться не будет.

Parameter	Description
Username attribute	Атрибут SAML, который будет использоваться как имя 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
SP entity ID	Уникальный идентификатор поставщика услуг (если не совпадает, операция будет отклонена). Можно указать URL или любую строку данных.
SP name ID format	Запросить определенный формат name ID 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
IdP certificate	Содержимое сертификата поставщика услуг (SP) для настройки службы единого входа SAML (SSO).
SP private key	Содержимое закрытого ключа поставщика услуг (SP) для настройки службы единого входа SAML (SSO). Обеспечивает безопасную аутентификацию и обмен данными с поставщиком удостоверений (IdP).
SP certificate	Содержимое сертификата поставщика услуг (SP) для настройки службы единого входа SAML (SSO).
Sign	Установите флажки, чтобы выбрать сущности, для которых должна быть включена подпись SAML: *Messages* *Assertions* *AuthN requests* *Logout requests* *Logout responses*
Encrypt	Установите флажки, чтобы выбрать сущности, для которых должно быть включено шифрование SAML: *Name ID* *Assertions*
Case-sensitive login	Снимите этот флажок, чтобы отключить чувствительность к регистру при входе для имен пользователей (по умолчанию включено). Отключение чувствительности к регистру позволяет, например, войти как ”admin”, даже если пользователь Zabbix называется ”Admin” или ”ADMIN”. Обратите внимание, что если чувствительность к регистру при входе отключена и существует несколько пользователей Zabbix с похожими именами (например, Admin и admin), вход для таких пользователей всегда будет отклонен со следующим сообщением об ошибке: ”Authentication failed: supplied credentials are not unique.”
Configure JIT provisioning	Установите этот флажок, чтобы отобразить параметры, связанные с JIT-подготовкой пользователей.
Group name attribute	Укажите атрибут имени группы для JIT-подготовки пользователей.
User name attribute	Укажите атрибут имени пользователя для JIT-подготовки пользователей.
User last name attribute	Укажите атрибут фамилии пользователя для JIT-подготовки пользователей.

Parameter	Description
User group mapping	Сопоставьте шаблон группы пользователя SAML с группой пользователей Zabbix и ролью пользователя. Это необходимо, чтобы определить, какую группу/роль получит подготовленный пользователь в Zabbix. Нажмите *Add*, чтобы добавить сопоставление. Поле *SAML group pattern* поддерживает подстановочные знаки. Имя группы должно соответствовать существующей группе. Если пользователь SAML соответствует нескольким группам пользователей Zabbix, пользователь станет участником всех этих групп. Если пользователь соответствует нескольким ролям пользователей Zabbix, он получит наивысший уровень прав среди них.
Media type mapping	Сопоставьте SAML-атрибуты медиа пользователя (например, email) с медиа пользователя Zabbix для отправки уведомлений.
Enable SCIM provisioning	Установите этот флажок, чтобы включить подготовку SCIM 2.0.

См. примеры настройки поставщиков удостоверений SAML для входа и подготовки пользователей в Zabbix с помощью:

- Microsoft Azure AD
- Okta
- Onelogin

Примечания по SCIM-подготовке

Для SCIM-подготовки укажите путь к веб-интерфейсу Zabbix и добавьте к нему api_scim.php на стороне поставщика удостоверений, например:

```
https://<path-to-zabbix-ui>/api_scim.php
```

Атрибуты пользователя, которые используются в Zabbix (имя пользователя, имя, фамилия и атрибуты медиа), необходимо добавить как пользовательские атрибуты и, при необходимости, внешний namespace должен совпадать со схемой пользователя: `urn:ietf:params:scim:schemas:core:2.0:User`.

Дополнительные настройки

Дополнительные параметры SAML можно настроить в файле конфигурации веб-интерфейса Zabbix (*zabbix.conf.php*):

- $SSO['SP_KEY'] = '<путь к файлу закрытого ключа SP>';
- $SSO['SP_CERT'] = '<путь к файлу сертификата SP>';
- $SSO['IDP_CERT'] = '<путь к файлу сертификата IDP>';
- $SSO['SETTINGS']

Note:
Массив `$SSO['SETTINGS']` должен соответствовать той же структуре, которую ожидает библиотека *SAML PHP Toolkit* (поставляемая с Zabbix). Полное описание доступных параметров конфигурации см. в официальной документации библиотеки.

В составе $SSO['SETTINGS'] можно задавать 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 будут взяты из базы данных и не могут быть переопределены. Параметр *debug* будет проигнорирован.

Кроме того, если веб-интерфейс Zabbix находится за прокси или балансировщиком нагрузки, можно использовать пользовательский параметр *use_proxy_headers*:

- *false* (по умолчанию) — игнорировать параметр;
- *true* — использовать HTTP-заголовки X-Forwarded-* для формирования базового URL.

Если для подключения к экземпляру Zabbix используется балансировщик нагрузки, где балансировщик использует TLS/SSL, а Zabbix — нет, необходимо указать параметры 'baseurl', 'strict' и 'use_proxy_headers' следующим образом:

```
$SSO['SETTINGS'] = [
    'strict' => false,
    'baseurl' => 'https://zabbix.example.com/zabbix/',
    'use_proxy_headers' => true
];
```

Пример конфигурации:

```
$SSO['SETTINGS'] = [
    'security' => [
        'signatureAlgorithm' => 'http://www.w3.org/2001/04/xmldsig-more#rsa-sha384'
        'digestAlgorithm' => 'http://www.w3.org/2001/04/xmldsig-more#sha384',
        // ...
    ],
    // ...
];
```

Настройка веб-интерфейса с Kerberos/ADFS

Файл конфигурации веб-интерфейса Zabbix (*zabbix.conf.php*) можно использовать для настройки SSO с аутентификацией Kerberos и ADFS:

```
$SSO['SETTINGS'] = [
    'security' => [
        'requestedAuthnContext' => [
            'urn:oasis:names:tc:SAML:2.0:ac:classes:Kerberos',
        ],
        'requestedAuthnContextComparison' => 'exact'
    ]
];
```

В этом случае в конфигурации SAML в поле *SP name ID* укажите:

```
urn:oasis:names:tc:SAML:1.1:nameid-format:unspecified
```

4 MFA

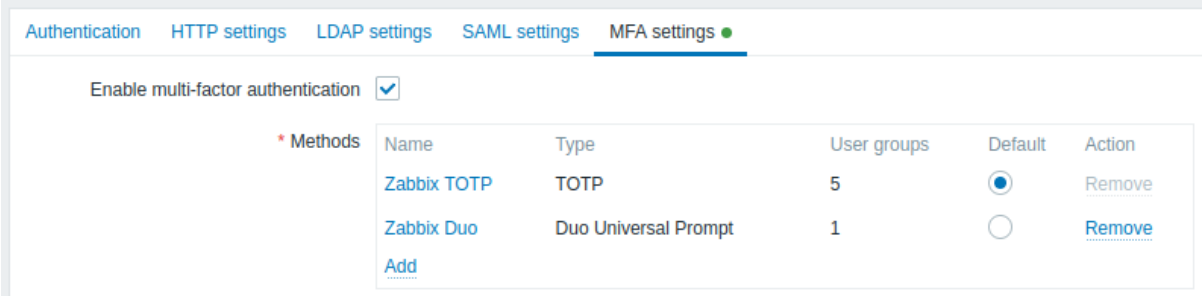
Обзор

Многофакторная аутентификация (MFA) может использоваться для входа в Zabbix, обеспечивая дополнительный уровень безопасности помимо имени пользователя и пароля.

При использовании MFA пользователь должен существовать в Zabbix, при входе должен указать учетные данные Zabbix, а также 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 наилучшим образом соответствует их требованиям безопасности и предпочтениям. Этими методами являются одноразовый пароль на основе времени (TOTP) и Duo Universal Prompt.

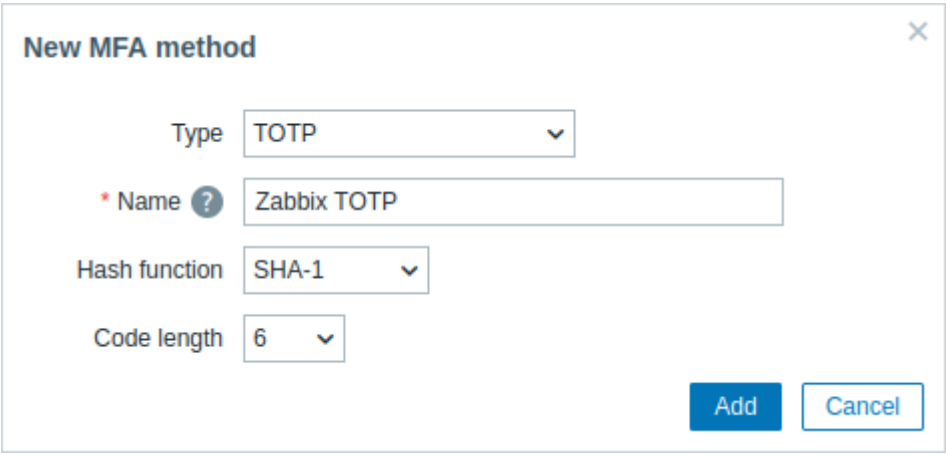
Настройка



Параметры настройки:

Параметр	Описание
Включить многофакторную аутентификацию	Установите этот флажок, чтобы включить многофакторную аутентификацию.
Методы	Нажмите *Добавить*, чтобы настроить метод MFA (см. настройку метода ниже).

Настройка метода



Параметры настройки метода:

Параметр	Описание
Тип	Выберите тип метода MFA: **TOTP** — использовать приложение-аутентификатор для генерации одноразовых паролей на основе времени; **Duo Universal Prompt** — использовать сервис аутентификации Duo для обеспечения многофакторной аутентификации.
Имя	Введите имя, которое отображается как имя учетной записи для всех пользователей MFA в приложениях-аутентификаторах (например, ”Zabbix”).
Хеш-функция	Выберите хеш-функцию (SHA-1, SHA-256 или SHA-512) для генерации кодов TOTP. Этот параметр доступен, если тип метода MFA установлен в значение ”TOTP”. Обратите внимание, что выбор SHA-256 или SHA-512 может значительно ограничить совместимость, так как 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 значение ”TOTP”.
Имя узла API	Введите имя узла API, предоставленное сервисом аутентификации Duo. Этот параметр доступен, если тип метода MFA установлен в значение ”Duo Universal Prompt”.

Параметр	Описание
ID клиента	Введите ID клиента, предоставленный сервисом аутентификации Duo. Этот параметр доступен, если тип метода MFA установлен в значение ”Duo Universal Prompt”.
Секрет клиента	Введите секрет клиента, предоставленный сервисом аутентификации Duo. Этот параметр доступен, если тип метода MFA установлен в значение ”Duo Universal Prompt”.

Примеры настройки

В этом разделе приведены примеры настройки MFA с использованием Time-Based One-Time Password (TOTP) и Duo Universal Prompt.

TOTP

Для TOTP пользователи должны подтверждать свою личность с помощью приложения-аутентификатора (например, приложения Google Authenticator).

1. Перейдите к настройкам MFA в Zabbix: *Пользователи* → *Аутентификация* и включите многофакторную аутентификацию.

2. Добавьте новый метод MFA со следующей конфигурацией:

- Тип: TOTP
- Имя: Zabbix TOTP
- Хеш-функция: SHA-1
- Длина кода: 6

3. Нажмите кнопку *Добавить*, а затем кнопку *Обновить*.

4. Перейдите в *Пользователи* → *Группы пользователей* и создайте новую группу пользователей со следующей конфигурацией:

- Имя группы: TOTP group
- Пользователи: Admin
- Многофакторная аутентификация: По умолчанию (или ”Zabbix TOTP”, если этот метод не установлен по умолчанию)

5. Выйдите из Zabbix и снова войдите, 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 необходимо ввести для завершения процесса входа.

7. При последующих входах получайте код подтверждения из приложения Google Authenticator и вводите его во время входа.

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 у вас включена 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 странице Duo Signup.

2. Откройте панель администрирования Duo, 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 к настройкам MFA в Zabbix: *Пользователи* → *Аутентификация* и включите многофакторную аутентификацию.

5. Добавьте новый метод MFA со следующей конфигурацией:

- Тип: Duo Universal Prompt
- Имя: Zabbix Duo
- API hostname: (используйте API hostname из Duo)
- Client ID: (используйте Client ID из Duo)
- Client secret: (используйте Client secret из Duo)

6. Нажмите кнопку *Add*, а затем кнопку *Update*.

7. Перейдите в *Пользователи* → *Группы пользователей* и создайте новую группу пользователей со следующей конфигурацией:

- Имя группы: Duo group
- Пользователи: Admin
- Многофакторная аутентификация: Default (или "Zabbix Duo", если этот метод не установлен по умолчанию)

8. Выйдите из Zabbix и снова войдите, используя свои учетные данные. После успешного входа вам будет предложено зарегистрироваться в MFA, и вы будете перенаправлены в Duo. Завершите настройку Duo и подтвердите пользователя с помощью приложения Duo на телефоне, чтобы войти в систему.

9. При последующих входах используйте соответствующий метод MFA, предоставляемый приложением Duo (например, получение кода подтверждения, ответ на push-уведомления или использование аппаратных ключей), и вводите необходимую информацию при входе.

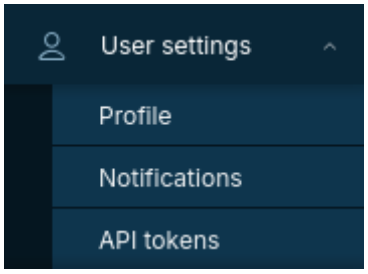
3 Настройки пользователя

Обзор

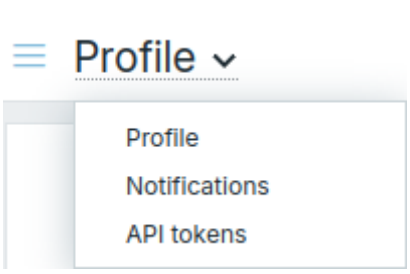
В зависимости от разрешений роли пользователя раздел *Настройки пользователя* может содержать следующие страницы:

- *Профиль* или *Профиль пользователя* — для настройки некоторых функций веб-интерфейса Zabbix;
- *Уведомления* — для настройки уведомлений текущего пользователя;
- *API-токены* — для управления API-токенами, назначенными текущему пользователю.

Список доступных страниц отображается после нажатия на значок пользователя в нижней части меню Zabbix (недоступно для пользователя *guest*). Также можно переключаться между страницами с помощью выпадающего списка заголовка в левом верхнем углу.



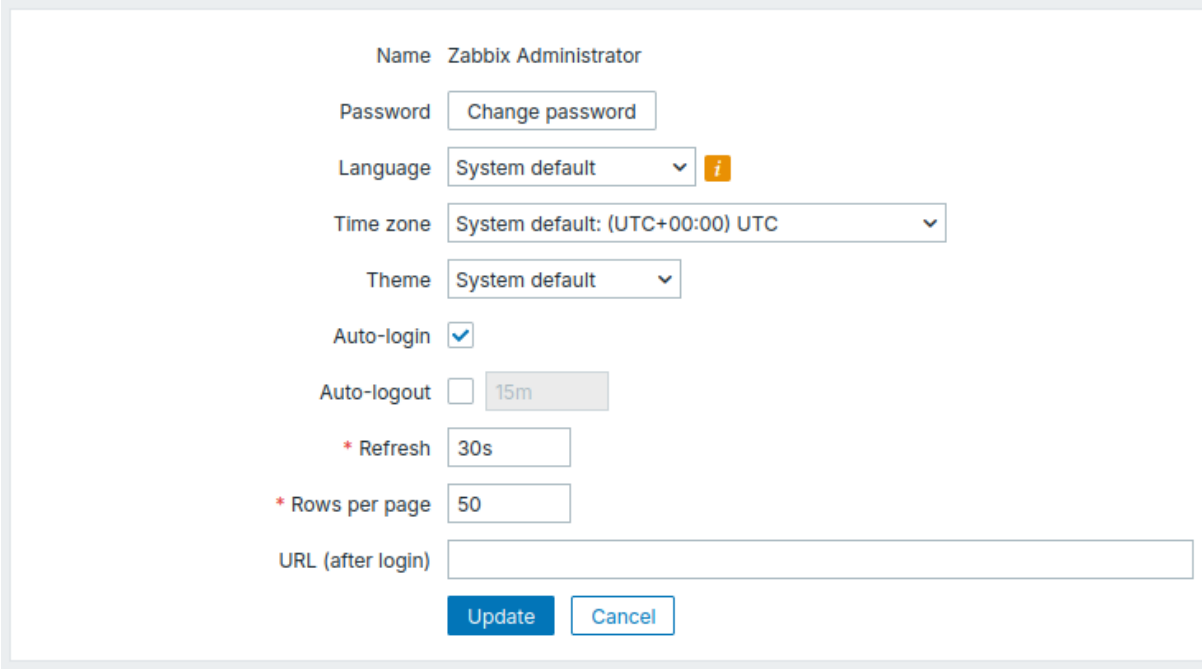
Меню третьего уровня.



Выпадающий список заголовка.

Профиль пользователя

Раздел *Профиль пользователя* предоставляет возможности для настройки пользовательского языка интерфейса, цветовой темы, количества строк, отображаемых в списках, и т. д. Изменения, внесенные здесь, будут применены только к текущему пользователю.



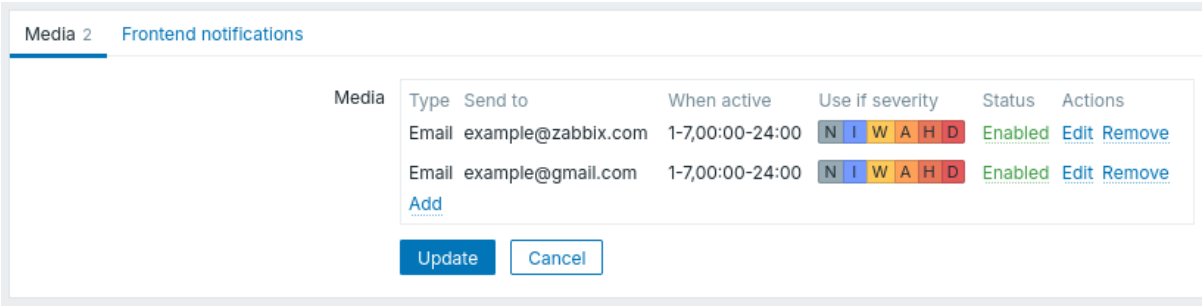
Параметр	Описание
Name	Отображаются имя и фамилия пользователя. Если имя и фамилия отсутствуют, отображается имя пользователя.
Password	Нажмите кнопку *Change password*, чтобы открыть три поля: *Old password*, *New password*, *New password (once again)*. При успешной смене пароля пользователь выйдет из всех активных сеансов. Обратите внимание, что пароль можно изменить только для пользователей, использующих внутреннюю аутентификацию Zabbix.
Language	Выберите нужный язык интерфейса или выберите **System default**, чтобы использовать системные настройки по умолчанию. Выбор *English (en_US)* также включает американский формат времени/даты в веб-интерфейсе. Подробнее см. в разделе Установка дополнительных языков веб-интерфейса.
Time zone	Выберите часовой пояс, чтобы переопределить глобальный часовой пояс на уровне пользователя, или выберите **System default**, чтобы использовать глобальные настройки часового пояса.
Theme	Выберите цветовую тему специально для своего профиля: **System default** — использовать системные настройки по умолчанию; **Blue** — стандартная синяя тема; **Blue (classic)** — стандартная синяя тема с использованием устаревших шрифтов; **Dark** — альтернативная темная тема; **Dark (classic)** — альтернативная темная тема с использованием устаревших шрифтов; **High-contrast light** — светлая тема с высокой контрастностью; **High-contrast dark** — темная тема с высокой контрастностью.
Auto-login	Установите этот флажок, чтобы Zabbix запомнил вас и автоматически выполнял вход в течение 30 дней. При входе с включенной опцией *Remember for 30 days*: - Таймер автоматического выхода сбрасывается (сеанс сохраняется до окончания 30-дневного периода). - Автоматический вход включается при последующих посещениях в течение 30 дней. При входе без *Remember for 30 days*: - Параметр автоматического входа очищается. - Автоматический выход выполняется по настроенному тайм-ауту, как обычно. Для этого используются cookie-файлы браузера.

Параметр	Описание
Auto-logout	Если этот флажок установлен, выход из системы будет выполняться автоматически по истечении заданного количества секунд (минимум 90 секунд, максимум 1 день). Поддерживаются суффиксы времени, например: 90s, 5m, 2h, 1d. Обратите внимание, что эта опция не будет работать в следующих случаях: * Когда страницы меню *Monitoring* выполняют фоновое обновление информации. Если оставить открытыми страницы, которые обновляют данные через определенные интервалы времени (дашборды, графики, последние данные и т. д.), время жизни сеанса соответственно продлевается, что отключает функцию автоматического выхода. * При входе с установленным флажком *Remember me for 30 days*; в этом случае таймер автоматического выхода переопределяется, и сеанс остается активным в течение всего периода. * Когда аутентификация выполняется через поставщика удостоверений SSO (IdP), автоматический выход завершает только сеанс Zabbix. SSO-сеанс у IdP остается активным, поэтому пользователь может снова войти без ввода пароля. *Auto-logout* также может принимать значение ”0”, что означает отключение функции автоматического выхода после обновления настроек профиля.
Refresh	Укажите, как часто будет обновляться информация на страницах меню *Monitoring* (минимум 0 секунд, максимум 1 час). Поддерживаются суффиксы времени, например: 30s, 90s, 1m, 1h.
Rows per page	Укажите, сколько строк будет отображаться на странице в списках. Меньшее количество строк (и меньшее количество отображаемых записей) обеспечивает более быструю загрузку.
URL (after login)	Укажите конкретный URL, который будет отображаться после входа. Вместо *Dashboards* по умолчанию это может быть, например, URL страницы *Monitoring* > *Triggers*.

Уведомления

Раздел *Уведомления* предоставляет параметры для настройки уведомлений текущего пользователя.

Вкладка *Способы оповещения* позволяет указать сведения о способах оповещения для пользователя, такие как используемые типы способов оповещения и адреса, а также время их использования для доставки уведомлений.



Если тип способа оповещения отключен:

- после имени отображается жёлтый 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*)

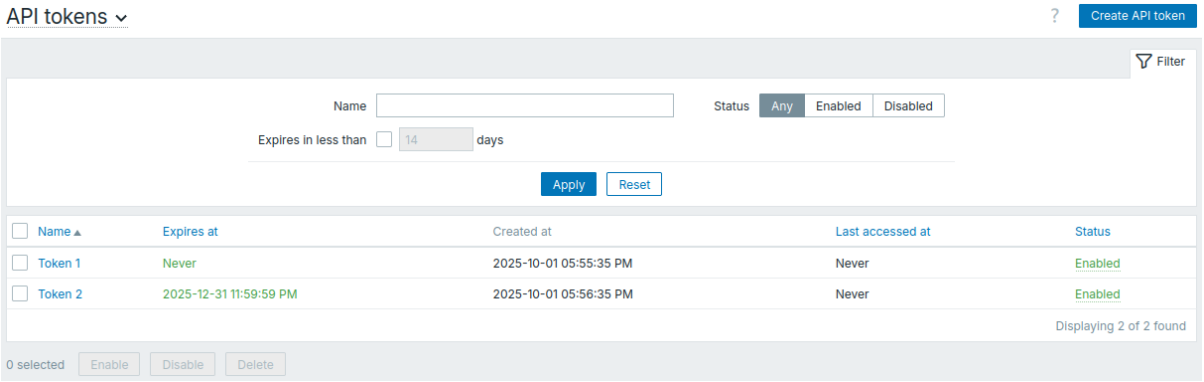
Обратите внимание, что для подготовленных пользователей:

- подготовленные способы оповещения пользователя нельзя удалить;
- подготовленные способы оповещения пользователя можно отключать/включать;
- такие поля подготовленных способов оповещения пользователя, как *Активно в течение*, *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.

Токены API

Раздел *Токены API* позволяет просматривать токены, назначенные пользователю, редактировать сведения о токенах и создавать новые токены. Этот раздел доступен пользователю только в том случае, если в настройках роли пользователя разрешено действие *Управление токенами API*.



Вы можете фильтровать токены API по имени, дате истечения срока действия или статусу (*Включен/Отключен*). Нажмите на статус токена в списке, чтобы быстро включить/отключить токен. Вы также можете включать/отключать сразу несколько токенов, выбрав их в списке, а затем нажав кнопки *Включить/Отключить* 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.

1 Глобальные оповещения

Обзор

Глобальные уведомления позволяют отображать проблемы в реальном времени непосредственно на текущем экране в веб-интерфейс Zabbix.

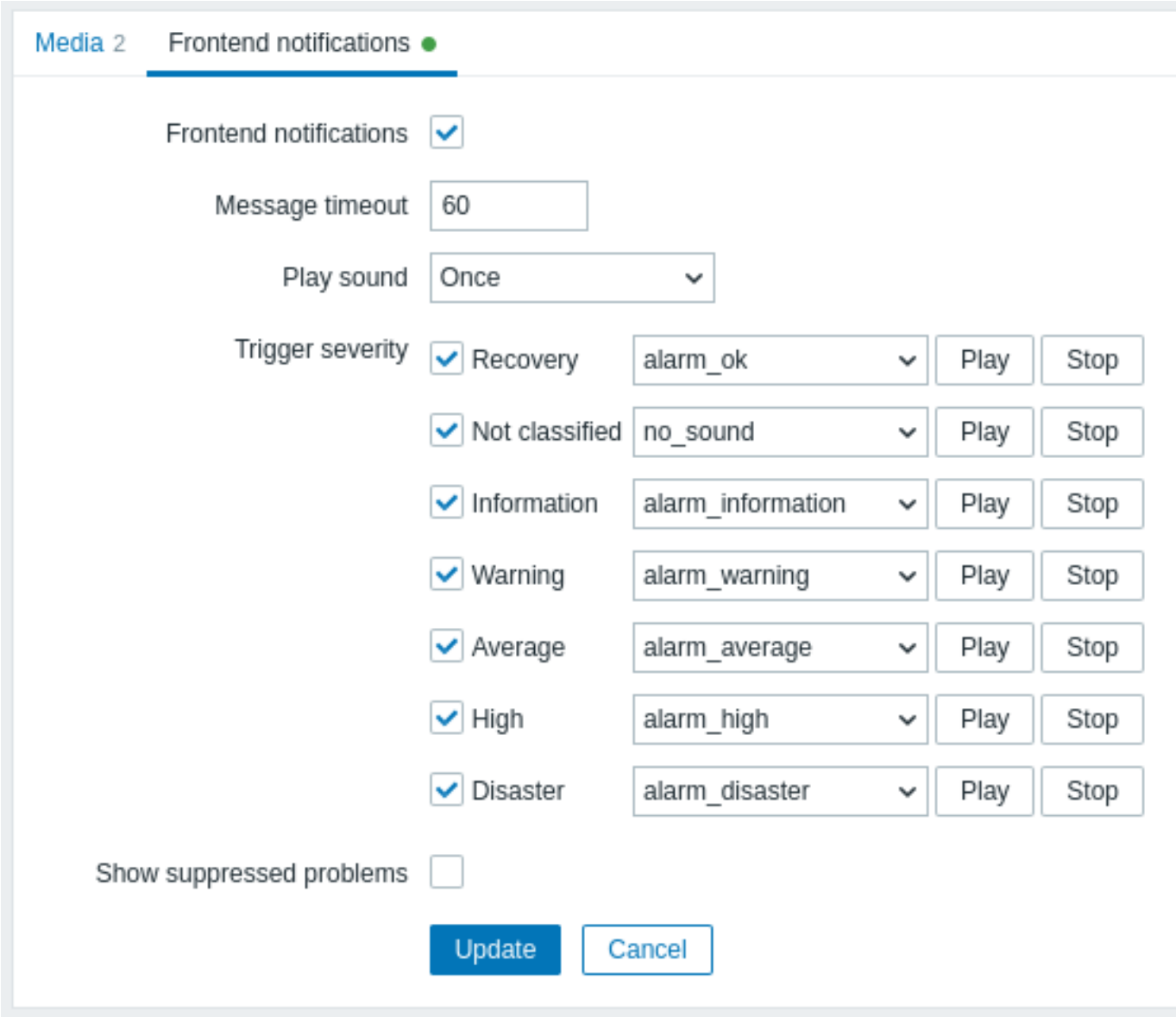
Без глобальных уведомлений при работе вне разделов *Проблемы* или *Панель мониторинга* вы не получали бы никакой информации о текущих проблемах. Глобальные уведомления обеспечивают отображение этой информации независимо от того, где именно вы находитесь в веб-интерфейс Zabbix.

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

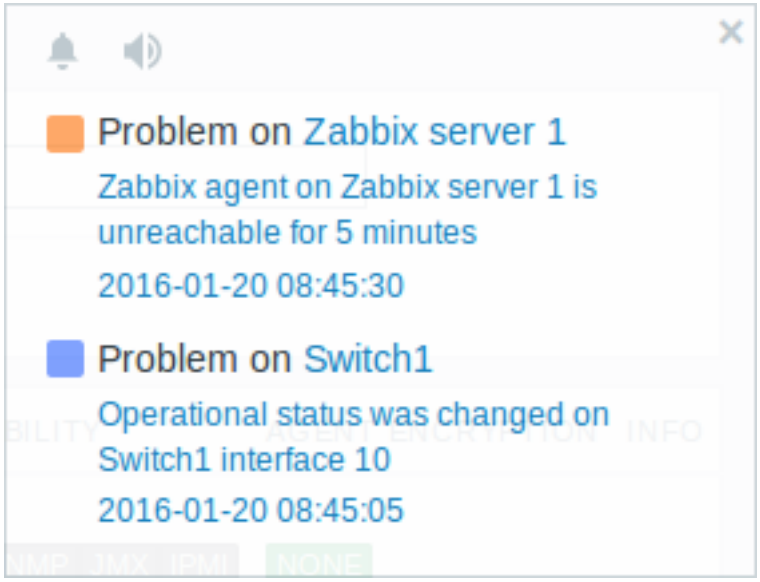
Глобальные уведомления можно включить для каждого пользователя на вкладке **Уведомления веб-интерфейса** в разделе Уведомления.



Параметр	Описание
Уведомления веб-интерфейса	Установите флажок, чтобы включить глобальные уведомления.
Тайм-аут сообщения	Укажите длительность, в течение которой будет отображаться сообщение. По умолчанию сообщения остаются на экране 60 секунд. Поддерживаются суффиксы времени, например: 30s, 5m, 2h, 1d.
Воспроизводить звук	Укажите длительность, в течение которой будет воспроизводиться звук. **Один раз** — звук воспроизводится один раз полностью; **10 секунд** — звук повторяется в течение 10 секунд; **Тайм-аут сообщения** — звук повторяется, пока сообщение отображается.
Важность триггера	Укажите важности триггера, для которых будут активированы глобальные уведомления и звуки. Вы также можете выбрать звуки, подходящие для различных важностей. Если ни одна важность не отмечена, сообщения отображаться не будут. Кроме того, сообщения о восстановлении будут отображаться только для отмеченных важностей. Например, если отмечены *Восстановление* и *Катастрофа*, глобальные уведомления будут отображаться для проблем и восстановлений триггеров с важностью *Катастрофа*.
Показывать подавленные проблемы	Установите флажок, чтобы отображать уведомления о проблемах, которые в противном случае были бы подавлены (не показаны) из-за обслуживания узла сети.

Отображаемые глобальные сообщения

По мере поступления сообщения отображаются во всплывающей области в правой части экрана. Вы можете свободно перемещать эту область, перетаскивая ее заголовок.



Для этой области доступны несколько элементов управления:

- Кнопка **Snooze** отключает текущий активный звуковой сигнал тревоги;
- Кнопка **Mute/Unmute** переключает воспроизведение звуковых сигналов тревоги между включенным и выключенным состоянием.

2 Звук в браузерах

Обзор

Звук используется в глобальных уведомлениях.

Чтобы звуки воспроизводились в веб-интерфейсе Zabbix, в профиле пользователя на вкладке *Frontend notifications* должен быть включён параметр *Frontend notifications*, при этом должны быть отмечены все уровни серьёзности триггеров. Кроме того, звуки должны быть включены во всплывающем окне глобальных уведомлений.

Если по какой-либо причине на устройстве невозможно воспроизвести звук, кнопка во всплывающем окне глобальных уведомлений будет постоянно оставаться в состоянии «mute», а при наведении курсора на кнопку будет отображаться сообщение «Cannot support notification audio for this device».

Звуки, включая аудиоклипы по умолчанию, поддерживаются только в формате MP3.

Звуки веб-интерфейса Zabbix были успешно протестированы в последних версиях браузеров Firefox и Opera в Linux, а также в браузерах Chrome, Firefox, Microsoft Edge и Opera в Windows.

Attention:
Автоматическое воспроизведение звуков может быть отключено (по умолчанию) в последних версиях браузеров. В таких случаях необходимо включить этот параметр вручную.

4 Глобальный поиск

В веб-интерфейсе Zabbix имеется возможность поиска узлов сети, групп узлов сети, шаблонов и групп шаблонов.

Поле поиска располагается в меню под логотипом Zabbix. Поиск можно начать, нажав *Enter* или на иконку поиска .



Если есть узел сети, название которого содержит введенную строку, то появится раскрывающийся список, в котором будут перечислены все подходящие узлы сети (совпадающая часть имени будет выделена оранжевым). В раскрывающемся списке также будет указан узел сети, если его видимое имя совпадает с техническим именем, введенным в строке поиска; в этом случае узел сети будет показан в списке, но без выделения цветом.

Атрибуты для поиска

Узлы сети можно искать по следующим свойствам:

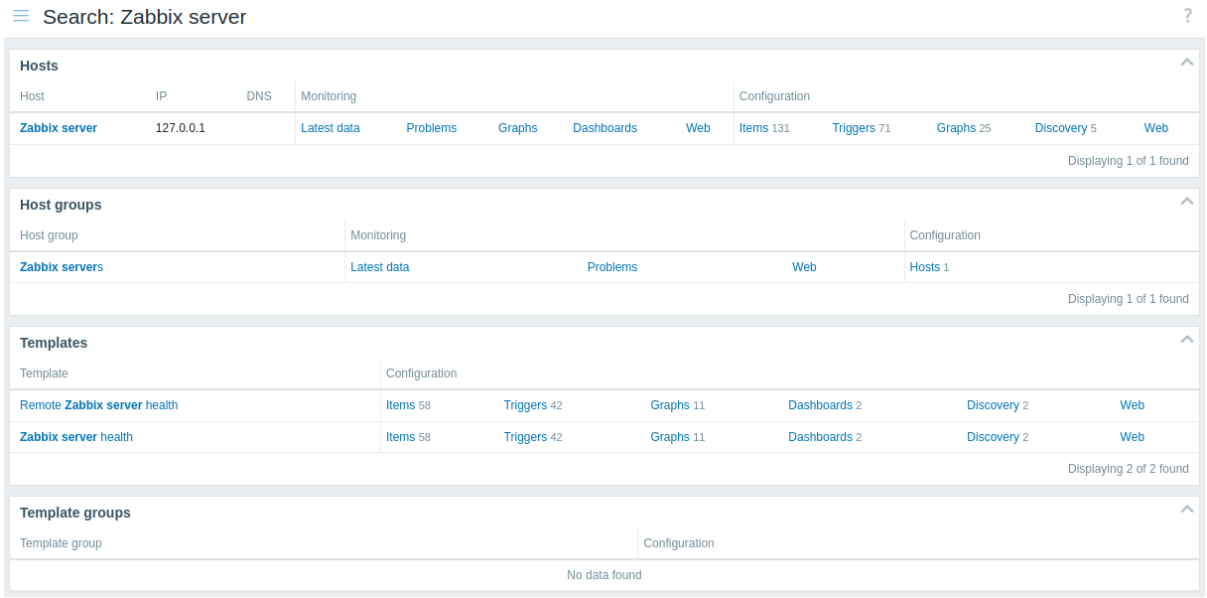
- Имя узла сети
- Отображаемое имя
- IP-адрес
- DNS-имя

Шаблоны можно искать по имени или отображаемому имени. Если выполнять поиск по имени, которое отличается от отображаемого имени (шаблона/узла сети), в результатах поиска оно отображается под отображаемым 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.



Можно свернуть/развернуть каждый отдельный блок. Количество записей отображается внизу каждого блока, например: *Отображается 13 из 13 найденных*. Если записей нет, количество 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. **Обратите внимание**, что если сущностей нет, число

не отображается.

Включенные узлы сети отображаются синим цветом, отключенные — красным.

Доступные ссылки

Для каждой записи доступны следующие ссылки:

- Узлы сети
  - Мониторинг
    - Последние данные
    - Проблемы
    - Графики
    - Панели узлов 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 шаблонов
    - Правила обнаружения
    - Веб-сценарии
- Группы шаблонов
  - Конфигурация
    - Шаблоны

5 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 её целостности.

Пока веб-интерфейс Zabbix находится в режиме обслуживания, вы можете безопасно остановить базу данных и выполнить задачи обслуживания.

Пользователи с указанных IP-адресов смогут работать с веб-интерфейсом в обычном режиме во время режима обслуживания.

Конфигурация

Чтобы включить режим обслуживания, откройте файл `maintenance.inc.php` (расположенный в `/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` расположен в каталоге `/conf` каталога HTML-документов Zabbix на веб-сервере. Однако некоторые операционные системы и веб-серверы могут использовать другое расположение.
Например, расположение для:

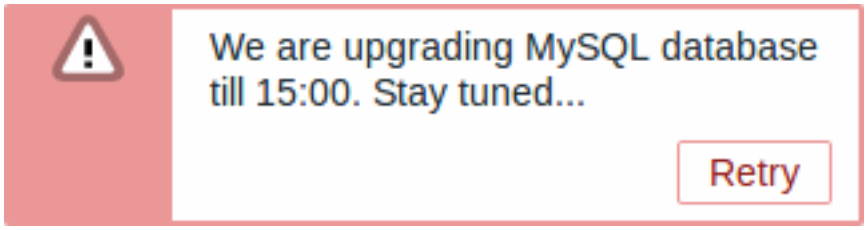
- SUSE и RedHat — `/etc/zabbix/web/maintenance.inc.php`.
- систем на основе Debian — `/usr/share/zabbix/conf/`.

См. также Копирование PHP-файлов.

Параметр	Подробности
ZBX_DENY_GUI_ACCESS	Если определен с любым значением, будет включен режим обслуживания. Чтобы отключить режим обслуживания, закомментируйте или удалите его.
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 об обслуживании (необязательно). Если не определено, будет использовано сообщение по умолчанию *'Zabbix is under maintenance'*.

Отображение

Пользователи увидят следующий экран при попытке получить доступ к веб-интерфейсу Zabbix, когда он находится в режиме обслуживания. Экран обновляется каждые 30 секунд, чтобы автоматически вернуться в обычное состояние после завершения обслуживания, без вмешательства пользователя.



IP-адреса, указанные в *ZBX_GUI_ACCESS_IP_RANGE*, смогут получать доступ к веб-интерфейсу как обычно.

6 Параметры страницы

Обзор

Большинство страниц веб-интерфейса Zabbix поддерживают различные HTTP GET-параметры, которые определяют, что будет отображаться. Их можно передавать, указывая пары parameter=value после URL, отделяя их от URL вопросительным знаком (?) и друг от друга амперсандом (&).

Мониторинг > Проблемы

Поддерживаются следующие параметры:

Параметр	Описание	Пример
show	Опция фильтра *Показывать*. Возможные значения: 1 - недавние проблемы; 2 - все; 3 - в состоянии проблемы.	show=1
name	Опция фильтра *Проблема*: строка в свободной форме.	name=Zabbix agent

Параметр	Описание	Пример
severities	Опция фильтра *Важность*: массив выбранных уровней важности в формате `severities[*]=*` (замените * на уровень важности). Возможные значения: 0 - не классифицировано; 1 - информация; 2 - предупреждение; 3 - средняя; 4 - высокая; 5 - катастрофа.	severities[3]=3
inventory	Опция фильтра *Инвентарные данные узла сети*: массив полей инвентарных данных `[field],[value]`	inventory[0][field]=type&invento
evaltype	Опция фильтра *Теги*: метод вычисления тегов. Возможные значения: 0 - и/или; 2 - или.	evaltype=0
tags	Опция 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	Опция фильтра *Показывать теги*. Возможные значения: 0 - нет; 1 - один; 2 - два; 3 - три.	show_tags=3
tag_name_format	Опция фильтра *Имя тега*. Возможные значения: 0 - полное имя; 1 - сокращённое имя; 2 - нет.	tag_name_format=1
tag_priority	Опция фильтра *Приоритет отображения тегов*: строка с приоритетом отображения тегов, разделённая запятыми	tag_priority=customer, target
show_suppressed	Опция фильтра *Показывать подавленные проблемы*. Возможные значения: 1 - показывать; 2 - не показывать.	show_suppressed=1
acknowledgement_status	Опция фильтра *Статус подтверждения*. Возможные значения: 0 - все; 1 - неподтверждённые; 2 - подтверждённые.	acknowledgement_status=0
acknowledged_by_me	Опция фильтра *Мной*; поддерживается только с `acknowledgement_status=2`. Возможные значения: 0 - отключено; 1 - включено.	acknowledged_by_me=1

Параметр	Описание	Пример
compact_view	Опция фильтра *Компактный вид*. Возможные значения: 0 - отключено; 1 - включено.	compact_view=1
highlight_row	Опция фильтра *Подсвечивать всю строку*. Возможные значения: 0 - отключено; 1 - включено.	highlight_row=1
filter_name	Опция свойств фильтра *Имя*: строка в свободной форме	filter_name=Databases
filter_show_counter	Опция свойств фильтра *Показывать количество записей*. Возможные значения: 0 - отключено; 1 - включено.	filter_show_counter=1
filter_custom_time	Опция свойств фильтра *Задать пользовательский период времени*. Возможные значения: 0 - отключено; 1 - включено.	filter_custom_time=1
sort	Столбец для сортировки. Возможные значения: clock - сортировка по столбцу *Время*; host - сортировка по столбцу *Узел сети*; severity - сортировка по столбцу *Важность*; name - сортировка по столбцу *Проблема*.	sort=clock
sortorder	Порядок сортировки результатов. Возможные значения: DESC - по убыванию; ASC - по возрастанию.	sortorder=DESC
age_state	Опция фильтра *Возраст менее*; поддерживается только с `show`=3. Возможные значения: 0 - отключить параметр `age`; 1 - включить параметр `age`.	age_state=1
age	Опция фильтра *Возраст менее*: integer, количество дней; поддерживается только с `age_state`=1 и `show`=3.	age=7
groupids	Опция фильтра *Группы узлов сети*: массив идентификаторов групп узлов сети.	groupids[]=4
hostids	Опция фильтра *Узлы сети*: массив идентификаторов узлов сети	hostids[]=10084
triggerids	Опция фильтра *Триггеры*: массив идентификаторов триггеров	triggerids[]=22382
show_timeline	Опция фильтра *Показывать временную шкалу*; не поддерживается с `compact_view`=1. Возможные значения: 0 - не показывать; 1 - показывать.	show_timeline=1
details	Опция 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: опции фильтра страницы «Проблемы».

Режим киоска

Режим киоска на поддерживаемых страницах веб-интерфейса можно активировать с помощью параметров URL. Например, на панелях:

- `/zabbix.php?action=dashboard.view&kiosk=1` - активировать режим киоска
- `/zabbix.php?action=dashboard.view&kiosk=0` - активировать обычный режим

Слайд-шоу

На панели можно активировать слайд-шоу:

- `/zabbix.php?action=dashboard.view&slideshow=1` - активировать слайд-шоу

7 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 на простых графиках и значение по умолчанию в выпадающем списке 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 сессии веб-интерфейса Zabbix.

По умолчанию: zbx_sessionid

- ZBX_DATA_CACHE_TTL

Тайм-аут TTL в секундах, используемый для сброса кэша данных ответа Vault. Установите 0, чтобы отключить кэширование ответа 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 отображаемых строк виджета.

По умолчанию: 1000

8 Создание своей темы

Обзор

По умолчанию Zabbix поставляется с некоторым количеством предустановленных тем. Вы можете выполнить процедуру приведенную здесь, шаг за шагом, для того, чтобы создать свою собственную тему. Не стесняйтесь поделиться результатами своей работы с сообществом Zabbix, если создали что-то действительно приятное.

Шаг 1

Для того чтобы создать свою тему, вам необходимо создать CSS файл и сохранить его в `assets/styles/` (назвав его, например, *custom-theme.css*) Вы можете скопировать эти файлы из другой темы и создать свою тему на основе этой темы или начать с нуля.

Шаг 2

Добавьте свою тему в список тем, возвращаемых методом APP::getThemes(). Вы можете сделать это, переопределив метод ZBase::getThemes() в классе приложения. Это можно сделать, добавив следующий код перед закрывающими фигурными скобками в *include/classes/core/APP.php*:

public static function getThemes() { return array_merge(parent::getThemes(), ['custom-theme' => _('Custom theme')]); }

Attention:
Обратите внимание, что имя, указанное в первой пара кавычек должна соответствовать имени файла темы без расширения.

Чтобы добавить несколько тем, просто перечислите их под первой темой, например:

public static function getThemes() { return array_merge(parent::getThemes(), ['custom-theme' => *('Custom theme'), 'anoth-ertheme'* => ('Another theme'), 'onemoretheme' => _('One more theme')]); }

Обратите внимание, что в каждой строке, кроме последней, должна быть завершающая запятая.

Note:
Чтобы изменить цвета графика, необходимо добавить запись в таблицу базы данных *graph_theme*.

Шаг 3

Активируйте новую тему.

Во внешнем интерфейсе Zabbix вы можете установить эту тему как тему по умолчанию. или изменить тему в профиле пользователя.

Наслаждайтесь новым внешним видом!

9 Режим отладки

Обзор

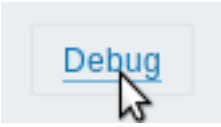
Режим отладки можно использовать для диагностики проблем с производительностью страниц веб-интерфейса Zabbix.

Настройка

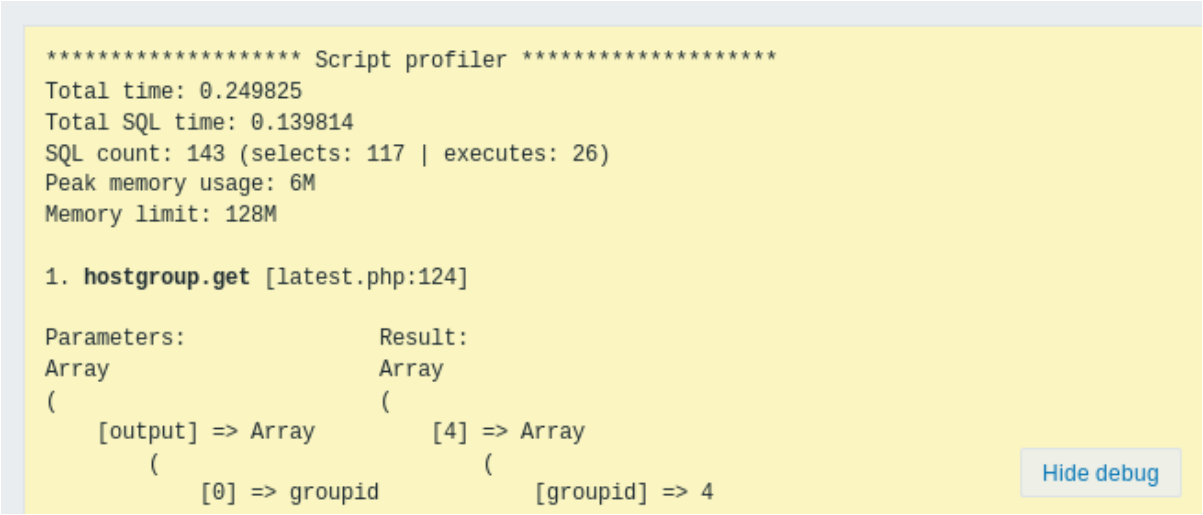
Режим отладки можно включить для отдельных пользователей, которые входят в группу пользователей:

- при настройке группы пользователей;
- при просмотре настроенных групп пользователей.

Если для группы пользователей включен *Режим отладки*, ее пользователи увидят кнопку *Отладка* в правом нижнем углу окна браузера:



Нажатие на кнопку *Отладка* открывает новое окно под содержимым страницы, содержащее статистику SQL для страницы, а также список вызовов API и отдельных SQL-запросов:



В случае проблем с производительностью страницы это окно можно использовать для поиска первопричины проблемы.

Warning:
Включенный *Режим отладки* отрицательно влияет на производительность веб-интерфейса.

10 Файлы cookie используемые Zabbix

Обзор

На этой странице приведён список файлов cookie, используемых Zabbix.

Name	Description	Values	Expires/Max-Age	HttpOnly[1]	Secure[2]
ZBX_SESSION_NAME	Данные сессии веб-интерфейса Zabbix, хранящиеся в виде JSON, закодированного с помощью base64		Сессия (истекает при завершении сеанса браузера)	+	+ (только если HTTPS включён на веб-сервере)
tab	Номер активной вкладки; этот cookie используется только на страницах с несколькими вкладками (например, на странице настройки *узла сети*, *триггера* или *действия*) и создаётся, когда пользователь переходит с основной вкладки на другую вкладку (например, *Метки* или *Зависимости*). 0 используется для основной вкладки.	Пример: 1	Сессия (истекает при завершении сеанса браузера)	-	-
browserwarning_ignore	Сохранить или игнорировать предупреждение об использовании устаревшего браузера.	yes	Сессия (истекает при завершении сеанса браузера)	-	-
system-message-ok	Сообщение, которое будет показано сразу после перезагрузки страницы.	Текстовое сообщение	Сессия (истекает при завершении сеанса браузера) или до 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.	Текстовое сообщение	Сессия (истекает при завершении сеанса браузера) или до перезагрузки страницы	+	-

Note:
Принудительная установка флага 'HttpOnly' для файлов cookie Zabbix с помощью директивы веб-сервера не поддерживается.

Сноски

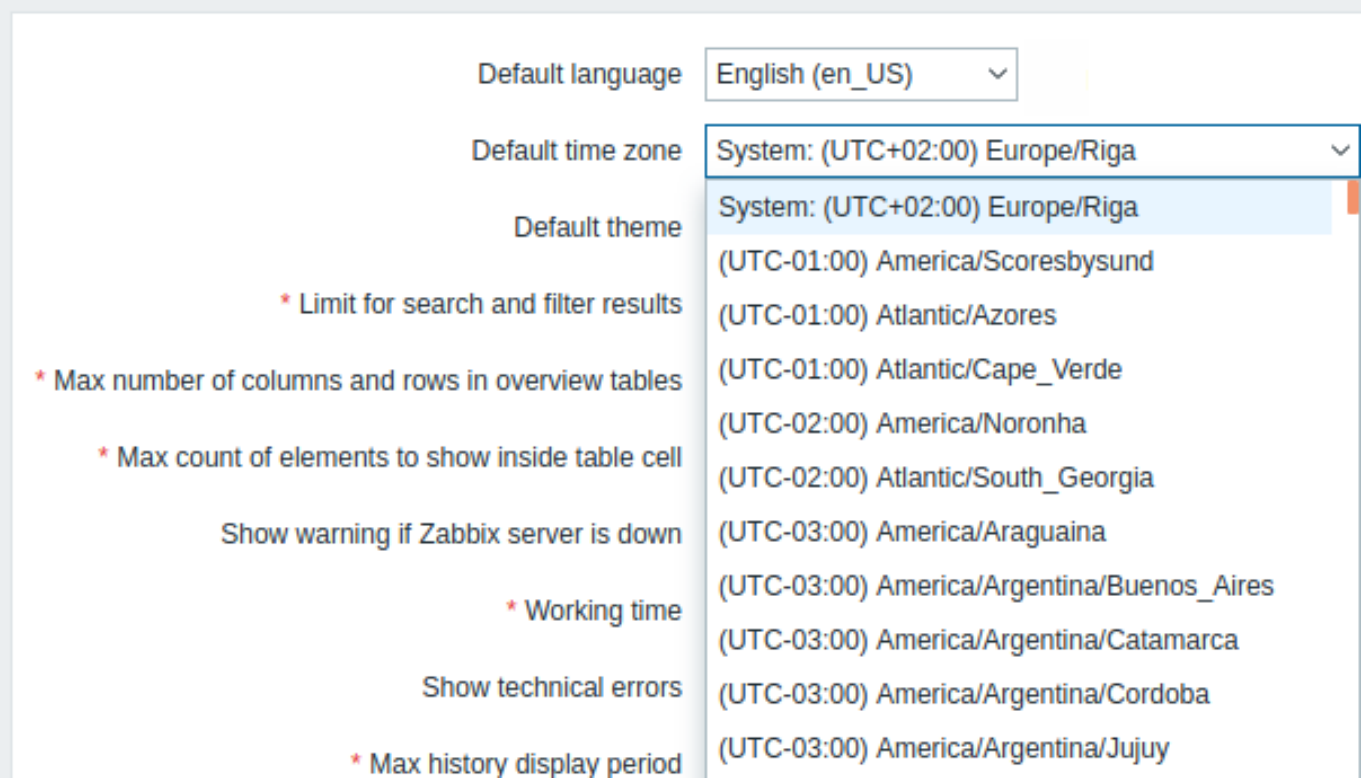
[1] Если `HttpOnly` имеет значение 'true', cookie будет доступен только через протокол HTTP. Это означает, что cookie не будет доступен для языков сценариев, таких как JavaScript. Этот параметр может эффективно помочь снизить риск кражи учётных данных через XSS-атаки (хотя он поддерживается не всеми браузерами).

[2] `Secure` указывает, что cookie должен передаваться только по защищённому HTTPS-соединению от клиента. Если установлено значение 'true', cookie будет установлен только при наличии защищённого соединения.

11 Часовые пояса

Обзор

Часовой пояс веб-интерфейса можно задать глобально в веб-интерфейсе и настроить отдельно для каждого пользователя.



Если выбрано *System*, для веб-интерфейса будет использоваться часовой пояс веб-сервера (включая значение 'date.timezone' из php.ini, если оно задано), тогда как сервер Zabbix будет использовать часовой пояс машины, на которой он запущен.

Note:
Сервер Zabbix будет использовать указанный глобальный/пользовательский часовой пояс только при раскрытии макросов в уведомлениях (например, {EVENT.TIME} может раскрываться в разный часовой пояс для каждого пользователя) и для ограничения по времени при отправке уведомлений (см. настройку "When active" в конфигурации медиа пользователя).

Выбор часового пояса не влияет на формат даты/времени в веб-интерфейсе. Вместо этого можно изменить язык интерфейса (либо во время установки, либо в настройках пользователя) — при выборе *English (en_US)* в веб-интерфейсе также будет использоваться американский формат даты/времени.

Конфигурация

Глобальный часовой пояс:

- можно задать вручную при установке веб-интерфейса
- можно изменить в *Administration* → *General* → *GUI*

Часовой пояс на уровне пользователя:

- можно задать при настройке/обновлении пользователя
- каждый пользователь может задать его в своём профиле пользователя

См. также: Выравнивание часовых поясов при использовании интервалов планирования.

12 Сброс пароля

Обзор В этом разделе описаны шаги по сбросу паролей пользователей в Zabbix.

Шаги Обратитесь к своему администратору Zabbix, если вы забыли свой пароль Zabbix и не можете войти в систему.

Пользователь с ролью супер администратора может изменить пароли для всех пользователей в форме настроек пользователя.

Если пользователь с ролью супер администратора забыл свой пароль и не может войти в систему, необходимо выполнить следующий SQL-запрос, чтобы сбросить пароль к паролю по умолчанию для пользователя с ролью супер администратора (замените 'Admin' соответствующим именем пользователя с ролью супер администратора):

```
UPDATE users SET passwd = '$2a$10$ZXIvHAEP2ZM.dLXTm6uPHOMVlARXX7cqjbhM6Fn0cANzkCQBWpMrS' WHERE username =
```

После выполнения этого запроса пароль пользователя будет установлен в значение *zabbix*. Обязательно измените пароль по умолчанию при первом входе в систему.

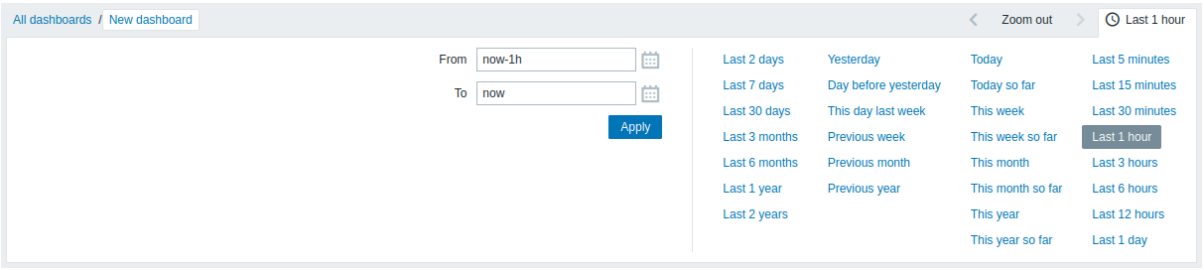
13 Селекторы периода времени и узла сети

Обзор

На этой странице описаны селекторы периода времени и узла сети, доступные на панелях. Эти селекторы позволяют динамически настраивать отображаемые данные во всех совместимых виджетах панели без изменения настроек отдельных виджетов.

Селектор периода времени

Селектор периода времени позволяет выбирать часто требуемые периоды одним щелчком мыши. Его можно развернуть или свернуть, нажав кнопку селектора периода времени в правом верхнем углу панели мониторинга.



Такие параметры, как *Сегодня*, *Эта неделя* и т. д., отображают весь период, включая часы/дни в будущем. Такие параметры, как *Сегодня до текущего момента*, *Эта неделя до текущего момента* и т. д., отображают только уже прошедшие часы.

После выбора периода его можно перемещать назад и вперед во времени, нажимая кнопки со стрелками  . Кнопка *Уменьшить масштаб* позволяет уменьшить масштаб периода на 50% в каждом направлении.

Note:
Для графиков (кроме графиков с *Пользовательским* периодом времени) выбрать отображаемый период времени также можно, выделив область на графике левой кнопкой мыши. После отпускания левой кнопки мыши график будет увеличен до выделенной области. Уменьшение масштаба также возможно двойным щелчком по графику.

Поля *От/До* отображают выбранный период либо в синтаксисе абсолютного времени (в формате `Y-m-d H:i:s`), либо в синтаксисе относительного времени. Относительный период времени может содержать одну или несколько математических операций (- или +), например `now-1d` или `now-1d-2h+5m`.

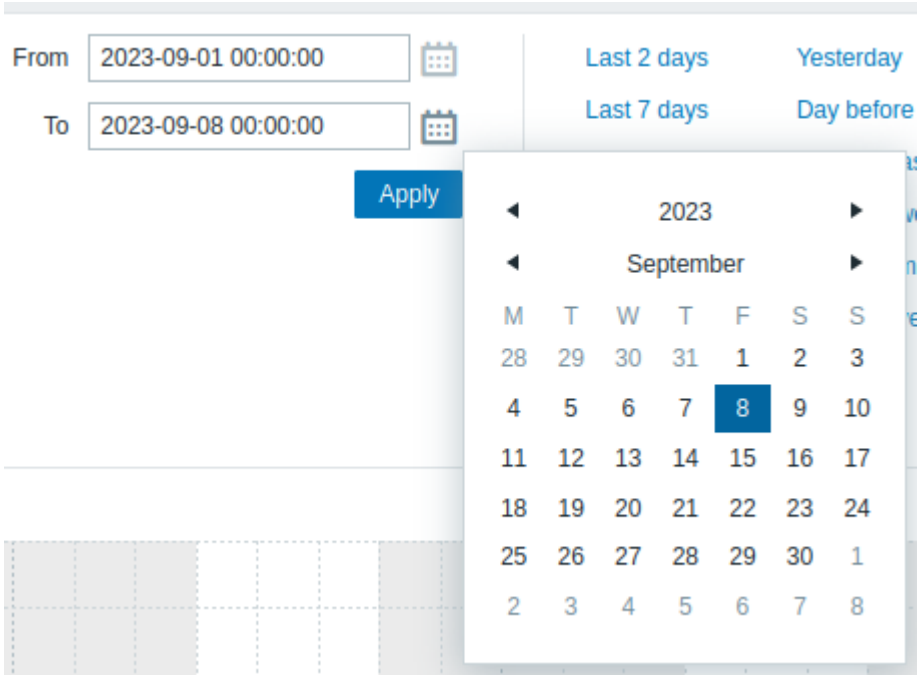
Поддерживаются следующие сокращения относительного времени:

- `now`
- `s` (секунды)
- `m` (минуты)
- `h` (часы)
- `d` (дни)
- `w` (недели)
- `M` (месяцы)
- `y` (годы)

В селекторе периода времени поддерживается точность (например, `/M` в `now-1d/M`). Подробности 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/датой с помощью `Tab`, `Shift+Tab` и клавиш со стрелками. Нажатие `Enter` подтверждает выбор.

Примеры:

От	До	Выбранный период
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 этой недели
now-1y/w	now-1y/w	Неделя с понедельника 00:00:00 по воскресенье 23:59:59 год назад

Attention:
Использование "now/M+1M" для параметра *До* может добавить 31 день, что может привести к смещению даты на 1-3 дня в зависимости от количества дней в месяце. Например, при использовании в январе результатом может быть 02 марта вместо ожидаемого 28 февраля. Чтобы избежать этой проблемы, используйте "now/M-3d+1M/M", которое корректно учитывает длину месяца. Аналогично, если настраивается *От* для перехода назад, используйте "now/M+3d-1M/M".

Селектор узла сети

Селектор узла сети позволяет выбрать узел сети, чтобы его данные отображались в совместимых виджетах панели мониторинга.

Он становится доступен после редактирования и сохранения панели мониторинга, если хотя бы один виджет настроен с параметром *Override host*, установленным на использование панели мониторинга в качестве источника данных.



17 Лучшие практики

Обзор В этом разделе описаны рекомендации по настройке Zabbix.

Хотя эти рекомендации не являются обязательными для работы Zabbix, их внедрение настоятельно рекомендуется для обеспечения оптимального и безопасного использования.

1 Рекомендации по обеспечению безопасности

Обзор

Этот раздел содержит рекомендации по безопасной настройке Zabbix.

Практики, описанные в этом разделе, не являются обязательными для работы Zabbix, но рекомендуются для повышения безопасности системы.

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 существуют известные проблемы безопасности.

Пути установщика Windows

При использовании установщиков Windows рекомендуется использовать пути по умолчанию, предлагаемые установщиком. Использование пользовательских путей без надлежащих 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, так как макросы не экранируются автоматически.

Рекомендации по безопасности Zabbix и база данных CVE

См. *Рекомендации по безопасности Zabbix и база данных CVE*.

HTML-шаблоны электронной почты

При создании или редактировании шаблонов сообщений, используемых для HTML-писем, всегда оборачивайте каждый макрос в макрофункцию `htmlencode()`. Например:

```
<b>Problem started</b> at {{EVENT.TIME}.htmlencode()} on {{EVENT.DATE}.htmlencode()}<br><b>Problem name:</
{{EVENT.NAME}.htmlencode()}<br><b>Host:</b> {{HOST.NAME}.htmlencode()}
<br><b>Severity:</b>
{{EVENT.SEVERITY}.htmlencode()}<br><b>Operational
data:</b> {{EVENT.OPDATA}.htmlencode()}<br><b>Original problem ID:</b> {{EVENT.ID}.htmlencode()}<br>{{TRIG
```

Использование `htmlencode()` гарантирует, что любые HTML-символы в значениях макросов будут экранированы, и предотвращает внедрение HTML в уведомления (например, когда злоумышленник вставляет вредоносную/фишинговую ссылку в уведомление).

Примечание: в стандартных HTML-сообщениях электронной почты, предоставляемых Zabbix, `htmlencode()` уже применяется к макросам. Эта рекомендация относится к редактированию существующих шаблонов и созданию новых — перед использованием шаблона для отправки HTML-писем убедитесь, что макросы закодированы.

1 Контроль доступа

Обзор

В этом разделе содержатся рекомендации по безопасной настройке контроля доступа.

Принцип наименьших привилегий

Учетные записи пользователей всегда должны работать с минимально возможными привилегиями. Это означает, что учетные записи пользователей в веб-интерфейсе Zabbix, пользователи базы данных или пользователь для процессов сервера/прокси/агента Zabbix должны иметь только те привилегии, которые необходимы для выполнения предусмотренных функций.

> **Attention:**
> Предоставление дополнительных привилегий пользователю 'zabbix' позволит ему получать доступ к конфигурационным файлам 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 непривилегированным пользователям. Например, хотя *Alerts* → *Scripts* доступно только пользователям типа *Super Admin*, скрипты также можно получить через Zabbix API. В этом случае ограничение прав на скрипты и исключение конфиденциальной информации из скриптов (например, учетных данных доступа) может помочь избежать раскрытия конфиденциальной информации, доступной в глобальных скриптах.

Защищенный пользователь для Zabbix агент

По умолчанию процессы Zabbix сервер, прокси и агент (или агент 2) используют одного общего пользователя `zabbix`. Чтобы предотвратить доступ Zabbix агент/агент 2 (запущенного на той же машине, что и сервер/прокси) к конфиденциальным данным в конфигурации сервера/прокси (например, учетным данным базы данных), агент следует запускать от имени другого пользователя:

Для Zabbix агент:

1. Создайте защищенные группу и пользователя (например, `zabbix-agent`).
2. Укажите этого пользователя в параметре User файла конфигурации агента.
3. Перезапустите агент, чтобы сбросить привилегии до нового пользователя.

Для Zabbix агент 2 конфигурация должна применяться на уровне service, поскольку файл конфигурации агент 2 не поддерживает параметр `User`. Пример см. в ZBX-26442.

Отзовите права на запись к 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

для пользователей без прав администратора к этому каталогу. В противном случае агент будет загружать настройки SSL из пути, который могут изменять непривилегированные пользователи, что создает потенциальную уязвимость безопасности.

Усиление безопасности компонентов Zabbix

Некоторые функции можно отключить для усиления безопасности компонентов Zabbix:

- выполнение глобальных скриптов на сервере Zabbix можно отключить, установив EnableGlobalScripts=0 в конфигурации сервера;
- выполнение глобальных скриптов на прокси Zabbix по умолчанию отключено (можно включить, установив EnableRemoteCommands=1 в конфигурации прокси);
- выполнение глобальных скриптов на агентах Zabbix по умолчанию отключено (можно включить, добавив параметр AllowKey=system.run[<command>,*] для каждой разрешённой команды в конфигурации агента);
- HTTP-аутентификацию пользователей можно отключить, установив `$ALLOW_HTTP_AUTH=false` в файле конфигурации веб-интерфейса (zabbix.conf.php). Обратите внимание, что повторная установка веб-интерфейса (запуск setup.php) удалит этот параметр.

Доступ к UNC-путям в Windows через агент Zabbix

Агенты Zabbix в Windows обрабатывают UNC-пути (SMB-ресурсы, такие как `\\server\share\file.txt`) в элементах данных, таких как `vfs.file.*`, `vfs.dir.*`, `modbus.get` и `perf_counter*`. В некоторых случаях это может представлять риск для безопасности.

Когда Windows получает запрос на доступ к UNC-пути, она пытается пройти аутентификацию на этом сервере. Это означает, что вредоносный запрос к агенту Zabbix может раскрыть NTLM-хеш серверу, отправившему запрос. При необходимости пользователи могут снизить этот риск с помощью параметров конфигурации AllowKey и DenyKey.

1 Защита MySQL/MariaDB

Обзор

В этом разделе содержатся рекомендации по обеспечению безопасности базы данных MySQL/MariaDB.

> **Note:**
> Для базовой настройки см. стандартные инструкции по созданию базы данных MySQL/MariaDB, которые включают создание пользователя 'zabbix' с полными привилегиями для базы 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 Zabbix сервера и прокси:

```
CREATE ROLE 'zbx_srv';
GRANT DELETE, INSERT, SELECT, UPDATE ON zabbix.* TO 'zbx_srv';
FLUSH PRIVILEGES;
```

- **zbx_web** - роль для запуска веб-интерфейса Zabbix и API:

```
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 определенных 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';
--    MariaDB:                  (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 ролей их можно назначить пользователям.

Назначение ролей пользователей

Чтобы назначить созданные роли пользователей, создайте пользователей и назначьте им соответствующие роли. При необходимости замените <user>, <host>, <role> и <password>.

```
CREATE USER '<user>'@'<host>' IDENTIFIED BY '<password>';
GRANT '<role>' TO '<user>'@'<host>';
SET DEFAULT ROLE '<role>' TO '<user>'@'<host>';
-- For MariaDB: SET DEFAULT ROLE '<role>' FOR '<user>'@'<host>'
FLUSH PRIVILEGES;
```

Например, чтобы создать и назначить роль для запуска сервера Zabbix и прокси:

```
CREATE USER 'usr_srv'@'localhost' IDENTIFIED BY 'password';
GRANT 'zbx_srv' TO 'usr_srv'@'localhost';
SET DEFAULT ROLE ALL TO 'usr_srv'@'localhost';
FLUSH PRIVILEGES;
```

2 Защита PostgreSQL/TimescaleDB

Обзор

В этом разделе содержатся рекомендации по обеспечению безопасности базы данных PostgreSQL.

Note:
Для базовой настройки см. стандартные инструкции по созданию базы данных PostgreSQL, которые включают создание пользователя 'zabbix' с полными привилегиями для базы 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 безопасную схему 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 предусмотренных 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, изначально принадлежащей владельцу базы данных, и не может быть предоставлено или отозвано.

Attention:
Следующие команды на этой странице должны выполняться только 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 сервера Zabbix и прокси:

```
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. При необходимости замените <user>, <role> и <password>.

```
CREATE USER <user> WITH ENCRYPTED password '<password>';
GRANT <role> TO <user>;
```

Например, чтобы создать и назначить роль для запуска сервера Zabbix и прокси:

```
CREATE USER usr_srv WITH ENCRYPTED password 'password';
GRANT zbx_srv TO usr_srv;
```

Секционирование базы данных с помощью TimescaleDB

Секционирование базы данных выполняется с помощью TimescaleDB. Для использования TimescaleDB серверу Zabbix требуются привилегии владельца базы данных.

Если схема PostgreSQL `zabbix` уже создана в базе данных `zabbix`, вы можете включить TimescaleDB следующей командой:

```
echo "CREATE EXTENSION IF NOT EXISTS timescaledb WITH SCHEMA zabbix CASCADE;" | sudo -u postgres psql zabb
```

2 Криптография

Обзор

В этом разделе содержатся 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 запрашиваемые поля. Самая важная строка — та, в которой запрашивается `Common Name`. Необходимо указать доменное имя, которое вы хотите связать с вашим сервером. Если у вас нет доменного имени, 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
```

3 Веб-сервер

Обзор

В этом разделе содержатся рекомендации по безопасной настройке веб-сервера.

Настройка перенаправления корневого URL на SSL Zabbix

В системах на базе RHEL добавьте виртуальный хост в конфигурацию Apache (`/etc/httpd/conf/httpd.conf`) и настройте постоянное перенаправление для корневого каталога документов на SSL URL Zabbix. Обратите внимание, что `example.com` следует заменить фактическим именем сервера.

```
#### Add lines:

<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 с понижением уровня 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
```

Принудительное применение атрибутов Secure и SameSite для cookie сессии в Zabbix

При настройке Zabbix крайне важно принудительно применять атрибуты Secure и SameSite для cookie сессии, чтобы повысить безопасность и предотвратить атаки межсайтовой подделки запросов (CSRF). Однако применение `SameSite=Strict` может вызывать проблемы в некоторых сценариях, например:

- Виджеты URL на панели мониторинга отображают сообщение «пользователь не вошёл в систему» при встраивании iframe с того же домена.
- Пользователи, открывающие панель мониторинга по HTTP вместо HTTPS, могут столкнуться с проблемами входа.
- Невозможность делиться URL-адресами на определённые разделы меню Zabbix или узлы сети.

Чтобы уменьшить влияние этих проблем, у пользователей должна быть возможность настраивать политику SameSite.

1. Secure cookie

Установка флага `secure` гарантирует, что cookie передаются только по HTTPS, предотвращая их раскрытие через незашифрованные соединения.

Чтобы включить Secure cookie в Zabbix, добавьте или измените следующий параметр в конфигурации веб-сервера:

Для Apache:

```
Header always edit Set-Cookie ^(.*)$ $1;Secure
```

Для Nginx:

```
proxy_cookie_path / "/; Secure";
```

Убедитесь, что доступ к веб-интерфейсу Zabbix осуществляется 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; #       ~    'zbx_session'
```

Включение Content Security Policy (CSP) на веб-сервере

Чтобы защитить веб-интерфейс Zabbix от Cross Site Scripting (XSS), внедрения данных и подобных атак, мы рекомендуем 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 установки веб-интерфейса Zabbix по умолчанию и для случаев, когда все содержимое поступает из домена сайта (за исключением поддоменов). Другая конфигурация заголовка CSP может потребоваться, если вы, например, настраиваете виджет *URL* для отображения содержимого 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, 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 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 Content-Security-Policy: "default-src 'self' *.openstreetmap.org; script-src 'self' 'unsafe
</VirtualHost>
```

3. Перезапустите службу Apache, чтобы применить изменения:

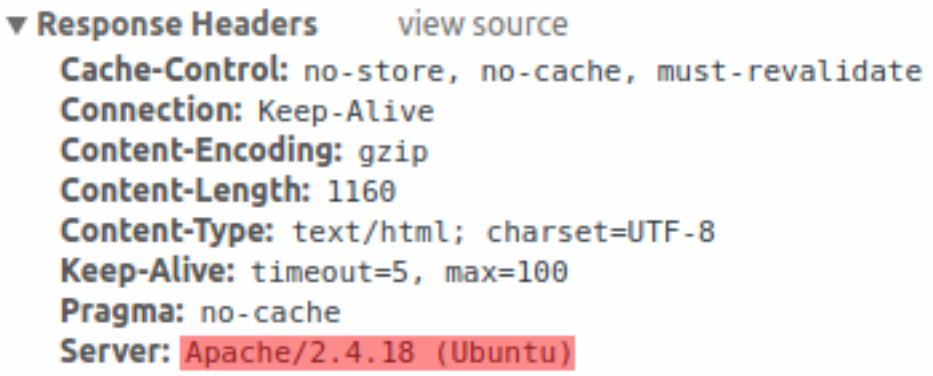
```
####             RHEL:
systemctl restart httpd.service

####   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
```

Чтобы изменения в файлах конфигурации вступили в силу, требуется перезапуск веб-сервера.

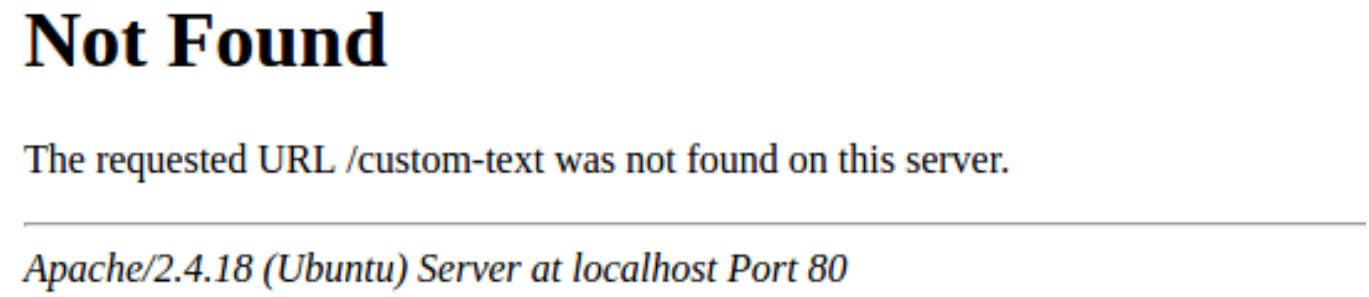
Для дополнительной безопасности можно использовать инструмент *mod_security* с Apache (пакет *libapache2-mod-security2*). Этот инструмент позволяет удалить сигнатуру сервера полностью, а не только убрать из нее версию сервера. Сигнатуру сервера можно изменить на любое значение, установив "SecServerSignature" в нужное значение после установки *mod_security*.

Обратитесь к документации вашего веб-сервера, чтобы узнать, как удалить/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:



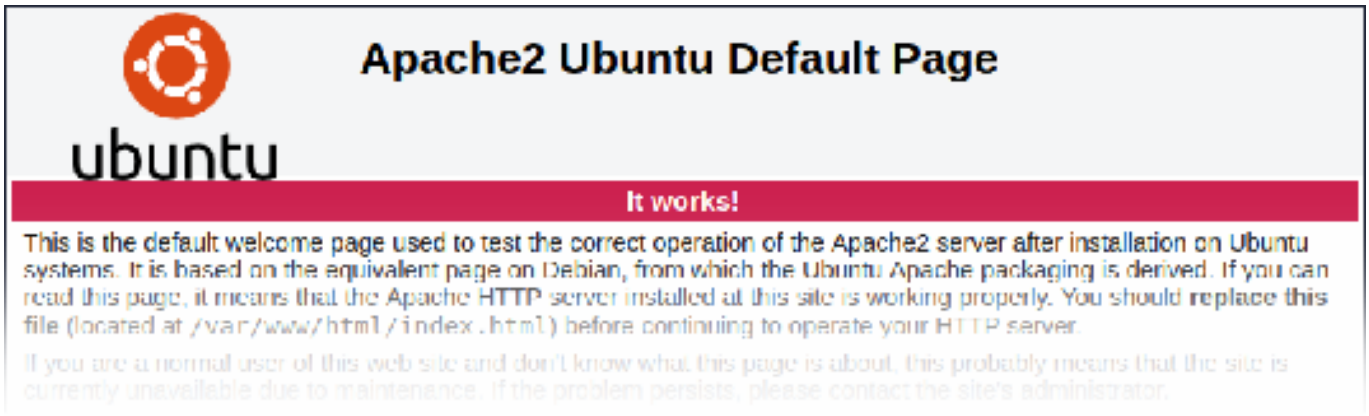
Эти стандартные страницы ошибок следует заменить/удалить. Например, директиву "ErrorDocument" можно использовать для определения пользовательской страницы/текста ошибки для веб-сервера Apache.

Обратитесь к документации вашего веб-сервера, чтобы узнать, как заменить/удалить стандартные страницы ошибок.

Удаление тестовой страницы веб-сервера

Чтобы избежать раскрытия информации, рекомендуется удалить тестовую страницу веб-сервера.

По умолчанию корневой каталог веб-сервера Apache содержит тестовую страницу `index.html`:



Обратитесь к документации вашего веб-сервера, чтобы узнать, как удалить тестовые страницы по умолчанию.

Установка HTTP-заголовка ответа X-Frame-Options

По умолчанию Zabbix настроен с параметром *Use X-Frame-Options HTTP header*, установленным в `SAMEORIGIN`. Это означает, что содержимое может загружаться только во фрейме, имеющем тот же источник, что и сама страница.

Элементы веб-интерфейса Zabbix, которые получают содержимое из внешних URL (а именно, виджет URL панели мониторинга), отображают полученное содержимое в песочнице со всеми включёнными ограничениями sandbox.

Эти настройки повышают безопасность веб-интерфейса Zabbix и обеспечивают защиту от атак XSS и clickjacking. Пользователи с ролью *Super admin* могут при необходимости изменить параметры *Use iframe sandboxing* и *Use X-Frame-Options HTTP header*. Перед изменением настроек по умолчанию следует тщательно взвесить риски и преимущества. Не рекомендуется полностью отключать sandboxing iframe или HTTP-заголовок X-Frame-Options.

Скрытие файла со списком распространённых паролей

Чтобы повысить сложность атак перебора паролей, рекомендуется ограничить доступ к файлу `ui/data/top_passwords.txt`. Этот файл содержит список наиболее распространённых и контекстно-зависимых паролей и не позволяет пользователям устанавливать такие пароли (если в политике паролей включён параметр *Избегать легко угадываемых 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;
}
```

2 Лучшие практики настройки

Обзор

В этом разделе описаны рекомендации по настройке Zabbix для достижения оптимальной производительности и удобства использования.
Эти рекомендации основаны на советах разработчиков Zabbix, а также на практическом опыте тренеров и инженеров службы поддержки Zabbix.

Каждая установка Zabbix уникальна, и некоторые из этих рекомендаций могут не подходить для вашей конкретной конфигурации.
Тем не менее, рекомендуется по возможности придерживаться этих рекомендаций, чтобы избежать распространённых потенциальных проблем.

> **Note:**
> Если вы считаете, что эту страницу можно улучшить, мы будем рады услышать ваше мнение! Пожалуйста, выделите соответствующий текст и нажмите **ctrl+Enter**, чтобы сообщить об ошибке или поделиться своим отзывом.

Узлы сети и элементы данных Определение узла сети

Узел сети в Zabbix — это не физическая машина или устройство, а логическая сущность. Для целей мониторинга вы можете создать отдельные узлы сети для базы данных или, например, виртуальной машины. Либо можно создать общий узел сети *ноутбук Джона* и отслеживать все метрики в рамках этого узла сети.

Рекомендуется создавать отдельный узел сети для каждого независимого экземпляра, такого как виртуальная машина, база данных, контейнер или сетевой коммутатор. Используя этот подход, вы сможете:

1. Избежать путаницы в данных мониторинга благодаря наличию отдельных элементов данных, триггеров и уведомлений о проблемах для каждого узла сети.
2. Точно настроить уровни доступа пользователей. Вы можете настроить роли пользователей, чтобы предоставить доступ на просмотр и/или настройку только определённых узлов сети. См. также принцип наименьших привилегий.

Узлы сети с дублирующимися элементами данных

Если у вас есть несколько похожих узлов сети, например, *Network switch 1* и *Network switch 2*, Zabbix предоставляет несколько способов быстро воссоздать узел сети. Вы можете просто клонировать узел сети со всеми его метриками, нажав кнопку Clone, но в этом случае, чтобы позже обновить элемент данных, вам потребуется сделать это вручную на каждом узле сети.

Рекомендуется создать шаблон со всеми необходимыми метриками, например, *Network switch template*. Затем сгруппируйте похожие узлы сети в группу узлов сети, в приведенном выше примере это может быть *Network switches*. Теперь в разделе *Data Collection -> Hosts* вы можете отфильтровать все узлы сети по группе узлов сети и использовать кнопку *Mass update*, чтобы привязать шаблон ко всем вашим сетевым коммутаторам.

Зависимые элементы данных

Чтобы свести к минимуму количество запросов к целевому объекту, Zabbix позволяет создавать основные и зависимые элементы данных. В этом случае основной элемент данных собирает большой набор информации одним запросом. Затем можно настроить зависимые элементы данных так, чтобы они извлекали определенные фрагменты данных из этого набора с помощью предобработки и сохраняли их как отдельные метрики.

Например, основной элемент данных может получать ответ JSON или XML, содержащий несколько метрик, либо выполнять запрос к базе данных, который возвращает несколько столбцов данных (например, количество открытых соединений, количество прерванных соединений, максимально допустимое число одновременных соединений и общее накопленное количество соединений с момента запуска), а зависимые элементы данных будут разбирать и сохранять каждое требуемое значение отдельно.

Рекомендуемая практика для такой конфигурации — отбрасывать историю основного элемента данных сразу после сбора и хранить только данные зависимых элементов данных.

Серверы и прокси

Если все узлы сети находятся в одной локальной сети с сервером Zabbix и нет проблем с масштабируемостью или производительностью, прокси может не понадобиться. В более крупных или сложных средах прямого мониторинга узлов сети сервером Zabbix может быть недостаточно. Добавление прокси и назначение части узлов сети этому прокси позволяет более равномерно распределить нагрузку.

Рекомендуется добавлять прокси Zabbix в следующих случаях:

1. Вы осуществляете мониторинг нескольких узлов сети с использованием различных методов сбора метрик за межсетевым экраном. Прокси будет собирать данные с узлов сети и пересылать их на сервер Zabbix, уменьшая необходимость открытия нескольких портов в межсетевом экране.
2. Вы осуществляете мониторинг удалённых площадок, филиалов и/или сетей. В случае прерывания сетевого соединения между сервером Zabbix и вашими удалёнными площадками прокси Zabbix, развернутые на удалённых площадках, продолжат сбор данных и отправят собранные данные обратно на сервер Zabbix, как только сетевое соединение будет восстановлено.
3. У вас крупномасштабное развертывание, и вы хотите снизить нагрузку на сервер Zabbix и повысить производительность. Определение крупномасштабного развертывания очень широко и зависит не только от количества узлов сети, но и от числа значений, собираемых в секунду.

Секретные макросы

Вы можете использовать секретные пользовательские макросы либо как секретный текст, либо как секретные vault-макросы.

Для повышения безопасности при использовании секретных vault-макросов рекомендуется настроить получение значений макросов сервером Zabbix и прокси Zabbix независимо друг от друга. По умолчанию значения секретных макросов получает сервер Zabbix и передает их прокси Zabbix.

18 API

Обзор Zabbix API позволяет программно получать и изменять конфигурацию Zabbix, а также предоставляет доступ к историческим данным. Он широко используется для следующих задач:

- создания новых приложений для работы с Zabbix;
- интеграции Zabbix со сторонним программным обеспечением;
- автоматизации рутинных задач.

Zabbix API — это API на основе HTTP, поставляемый как часть веб-интерфейса. Он использует протокол JSON-RPC 2.0, что означает следующее:

- API состоит из набора отдельных методов;
- запросы и ответы между клиентами и API кодируются с использованием формата JSON.

Дополнительную информацию о протоколе и JSON см. в спецификации JSON-RPC 2.0 и на домашней странице формата JSON.

Дополнительную информацию об интеграции функциональности Zabbix в ваши приложения Python см. в разделе Библиотека Python для Zabbix.

> **Note:**
> Доступ пользователя 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 условно сгруппированы в отдельные API. Каждый метод выполняет одну отдельную задачу. Например, метод `host.create` относится к API *узла сети* и используется для создания новых узлов сети. Исторически для API иногда используется название "классы".

> **Note:**
> Большинство API содержит по крайней мере четыре метода: `get`, `create`, `update` и `delete` для получения, создания, обновления и удаления данных соответственно, однако некоторые API могут предоставлять совершенно другой 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 по адресу `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` с одним из следующих значений: `application/json-rpc`, `application/json` или `application/jsonrequest`.

Объект запроса должен содержать следующие свойства:

- `jsonrpc` - версия протокола JSON-RPC, используемая 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 он опущен, API рассматривает запрос как уведомление).

Если запрос корректен, ответ, возвращаемый API, должен выглядеть следующим образом:

```
{
    "jsonrpc": "2.0",
    "result": "8.0.0",
    "id": 1
}
```

Объект ответа, в свою очередь, содержит следующие свойства:

- `jsonrpc` - версия протокола JSON-RPC;
- `result` - данные, возвращаемые методом;
- `id` - идентификатор соответствующего запроса.

Пример рабочего процесса В следующем разделе приведено несколько примеров использования более детально.

Аутентификация Для доступа к любым данным в Zabbix необходимо:

- либо использовать существующий токен API (созданный в веб-интерфейсе Zabbix или с помощью Token API);
- либо использовать токен аутентификации, полученный методом user.login.

Например, если вы хотите получить новый токен аутентификации, войдя в систему как стандартный пользователь *Admin*, запрос JSON будет следующим:

```
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 вы правильно указали учетные данные, ответ API должен содержать токен аутентификации пользователя:

```
{
    "jsonrpc": "2.0",
    "result": "0424bd59b807674191e7d77572075f33",
    "id": 1
}
```

Методы авторизации По заголовку "Authorization"

Все API-запросы требуют аутентификации или API-токен. 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, смотрите перенаправление заголовка Authorization.

Zabbix API принимает заголовки без учета регистра (например, `authorization`, `Authorization` и `AUTHORIZATION` обрабатываются одинаково).

Заголовок Authorization поддерживается в междоменных запросах (CORS).

По cookie Zabbix

Cookie *"zbx_session"* используется для авторизации API-запроса из веб-интерфейса Zabbix, выполняемого с помощью JavaScript (из модуля или пользовательского виджета).

Получение узлов сети Теперь у вас есть действительный токен аутентификации пользователя (представленный как переменная в следующих примерах), который можно использовать для доступа к данным в Zabbix. Например, вы можете использовать метод host.get, чтобы получить идентификаторы, имена и интерфейсы 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 Создайте новый элемент данных на узле сети "Zabbix server", используя данные, полученные из предыдущего запроса `host.get`. Это можно сделать при помощи 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 обозначения этого элемента данных в последующих запросах:

```
{
    "jsonrpc": "2.0",
    "result": {
        "itemids": [
            "24759"
        ]
    },
    "id": 3
}
```

Note:
Метод `item.create`, так же как и другие *методы создания*, может принимать массивы объектов и создавать несколько элементов данных с помощью одного вызова API.

Создание нескольких триггеров Соответственно, если *методы создания* принимают массивы, вы можете добавить несколько триггеров, например:

```
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 ID только что созданных триггеров:

```
{
    "jsonrpc": "2.0",
    "result": {
        "triggerids": [
            "17369",
            "17370"
        ]
    },
    "id": 4
}
```

Обновление элемента данных Активируйте элемент данных, указав для "status" 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`, так же как и другие *методы обновления*, может принимать массивы объектов и обновлять несколько элементов данных с помощью одного вызова API.

Обновление нескольких триггеров Включите несколько триггеров, установив их статус в "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 всё, что вы пробовали, работало нормально. Но что произойдёт, если вы попытаетесь выполнить некорректный вызов API? Попробуйте создать ещё один узел сети, вызвав host.create, но не указав 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,
```

Тогда ответ будет содержать сообщение об ошибке:

```
{
    "jsonrpc": "2.0",
    "error": {
        "code": -32602,
        "message": "Invalid params.",
        "data": "No groups for host \"Linux server\"."
    },
    "id": 7
}
```

Если произошла ошибка, то 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, истечении времени сеанса или попытке доступа к несуществующим объектам. Ваше приложение должно уметь корректно обрабатывать такие ошибки.

Версии API Чтобы упростить управление версиями API, начиная с Zabbix 2.0.4, версия API соответствует версии самого Zabbix. Вы можете использовать метод apiinfo.version, чтобы узнать версию API, с которой вы работаете. Это может быть полезно для адаптации вашего приложения к использованию возможностей, зависящих от версии.

Zabbix гарантирует обратную совместимость функциональности в пределах основной версии. При внесении обратно несовместимых изменений между основными выпусками Zabbix обычно оставляет старые возможности устаревшими в следующем выпуске и удаляет их только в последующем выпуске. Иногда Zabbix может удалять возможности между основными выпусками без обеспечения какой-либо обратной совместимости. Важно никогда не полагаться на устаревшие возможности и как можно скорее переходить на более новые альтернативы.

Note:
Вы можете отслеживать все изменения, внесённые в API, в журнале изменений API.

Дополнительная литература Теперь у вас достаточно знаний, чтобы начать работать с Zabbix API, однако не останавливайтесь на этом. Для дальнейшего изучения рекомендуется 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 в процессе мониторинга.

Панели мониторинга

Управляйте панелями мониторинга и создавайте на их основе отчеты по расписанию.

Dashboard API | API панелей мониторинга шаблонов | API отчетов

Кластер высокой доступности

Получение списка узлов сервера и их статуса.

API кластера высокой доступности

История

Получение исторических значений, собранных процессами мониторинга Zabbix, для отображения или дальнейшей обработки.

History API

Тренды

Получение значений трендов, вычисленных 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

Получение проблем в соответствии с заданными параметрами.

Problem API

Карты

Настраивайте карты, чтобы создавать подробные динамические представления вашей IT-инфраструктуры.

API карт

Задачи

Взаимодействие с менеджером задач Zabbix сервера, создание задач и получение ответа.

API задач

Услуги Zabbix API позволяет получать доступ к данным, собранным в ходе мониторинга услуг.

Соглашение об уровне обслуживания

Определяйте цели уровня обслуживания (SLO), получайте подробную информацию о показателях уровня обслуживания (SLI) о производительности сервиса.

SLA API

Услуги

Управляйте услугами для мониторинга уровня услуг и получайте подробную информацию об SLA любой услуги.

API услуг

Сбор данных Zabbix API позволяет управлять конфигурацией вашей системы мониторинга.

Узлы сети и группы узлов сети

Manage host groups, hosts and everything related to them, including host interfaces, host macros and maintenance periods.

Управление группами узлов сети, узлами сети и всем, что с ними связано, включая интерфейсы узла сети, макросы узла сети и периоды обслуживания.

API узлов сети | API групп узлов сети | API интерфейсов узлов сети | API пользовательских макросов | API преобразования значений | API обслуживания

Элементы данных

Определяет элементы данных для мониторинга.

API элементов данных

Триггеры

Настройка триггеров для оповещения вас о наличии проблем в вашей системе. Управление зависимостями триггеров.

API триггеров

Графики

Редактируйте графики или отдельные элементы графиков для более наглядного представления собранных данных.

API графиков | API элементов графиков

Шаблоны и группы шаблонов

Управляйте шаблонами и связывайте их с узлами сети или другими шаблонами.

API шаблонов | API групп шаблонов | API преобразования значений

Обнаружение низкого уровня

Настраивайте правила обнаружения низкого уровня, 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

Корреляция событий

Создавайте пользовательские правила корреляции событий.

API корреляции

Обнаружение сети

Управляйте правилами обнаружения на уровне сети, чтобы автоматически находить и отслеживать новые узлы сети. Получите полный доступ к информации об обнаруженных службах и узлах сети.

API правил обнаружения | API проверок обнаружения | API обнаруженных узлов сети | API обнаруженных служб

Экспорт и импорт

Экспорт и импорт данных конфигурации Zabbix для резервного копирования конфигурации, миграции или крупномасштабных обновлений конфигурации.

Configuration API

Веб-мониторинг

Настройте веб-сценарии для мониторинга ваших веб-приложений и сервисов.

API веб-сценариев

Оповещения Zabbix API позволяет управлять действиями и оповещениями вашей системы мониторинга.

Действия и оповещения

Определяйте действия и операции, чтобы уведомлять пользователей об определённых событиях или автоматически выполнять удалённые команды. Получайте доступ к информации о созданных оповещениях и их получателях.

API действий | API оповещений

Способы оповещения

Настройте способы оповещения и несколько способов получения пользователями предупреждений.

API способов оповещения

Скрипты

Настраивайте и выполняйте скрипты, чтобы упростить выполнение повседневных задач.

API скриптов

Пользователи API Zabbix позволяет управлять пользователями вашей системы мониторинга.

Пользователи и группы пользователей

Добавьте пользователей, которым будет предоставлен доступ к Zabbix, назначьте их в группы пользователей и предоставьте разрешения. Создайте роли для детального управления правами пользователей.

User API | User group API | User directory API | User role API

API-токены

Управление токенами авторизации.

API токенов

Аутентификация

Изменение параметров конфигурации аутентификации.

API аутентификации

Администрирование С помощью Zabbix API вы можете изменять административные настройки вашей системы мониторинга.

Общие

Изменение определённых глобальных параметров конфигурации.

API авторегистрации | API карт значков | API изображений | API настроек | API регулярных выражений | API модулей | API коннекторов

Журнал аудита

Отслеживайте изменения конфигурации, выполненные каждым пользователем.

API журнала аудита

Очистка истории

Настройте очистку истории.

Housekeeping API

Прокси и группы прокси

Управляйте прокси, используемыми в вашей распределённой среде мониторинга.

API прокси | API групп прокси

Макросы

Управление макросами.

API пользовательских макросов

Информация об API Получите версию API Zabbix, чтобы ваше приложение могло использовать функции, зависящие от версии.

API info API

Alert

Этот класс предназначен для работы с оповещениями.

Ссылки на объекты:

- ·Alert

Доступные методы:

- ·alert.get - получение предупреждений

Alert object

The following objects are directly related to the `alert` API.

Оповещение

> **Note:**
> Оповещения создаются сервером Zabbix и не могут быть изменены через API.

Объект оповещения содержит информацию о том, были ли определенные операции действия успешно выполнены. Он имеет следующие свойства.

Свойство	Тип	Описание
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 возникают 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 использовался для отправки сообщения.
message	text	Текст сообщения. Поведение свойства: - *поддерживается*, если `alerttype` установлено в ”message”
retries	integer	Количество попыток отправки сообщения, выполненных Zabbix.
sendto	string	Адрес, имя пользователя или другой идентификатор получателя. Поведение свойства: - *поддерживается*, если `alerttype` установлено в ”message”

Свойство	Тип	Описание
status	integer	Статус, указывающий, была ли операция действия выполнена успешно. Возможные значения, если `alerttype` установлено в ”message”: 0 - сообщение не отправлено; 1 - сообщение отправлено; 2 - ошибка после нескольких попыток; 3 - новое оповещение еще не обработано менеджером оповещений. Возможные значения, если `alerttype` установлено в ”remote command”: 0 - команда не выполнена; 1 - команда выполнена; 2 - была попытка выполнить команду на агенте Zabbix, но он был недоступен.
subject	string	Тема сообщения. Поведение свойства: - *поддерживается*, если `alerttype` установлено в ”message”
userid	ID	ID пользователя, которому было отправлено сообщение.
p_eventid	ID	ID события проблемы, которое сгенерировало оповещение.
acknowledgeid	ID	ID подтверждения, которое сгенерировало оповещение.

alert.get

Описание

`integer/array alert.get(parameters)`

Метод позволяет получать оповещения по заданным параметрам.

Note:
Этот метод доступен для пользователей любого типа. Права доступа вызова метода можно отменить в настройках роли пользователя. Дополнительную информацию см. в разделе Роли пользователей.

Параметры

`(object)` Параметры, определяющие желаемый вывод.

Метод поддерживает следующие параметры.

Параметр	Тип	Описание
alertids	ID/array	Возвращать только оповещения с указанными ID.
actionids	ID/array	Возвращать только оповещения, созданные указанными действиями.
eventids	ID/array	Возвращать только оповещения, созданные указанными событиями.
groupids	ID/array	Возвращать только оповещения, созданные объектами из указанных групп узлов сети.
hostids	ID/array	Возвращать только оповещения, созданные объектами из указанных узлов сети.
mediatypeids	ID/array	Возвращать только оповещения сообщений, в которых использовались указанные типы медиа.
objectids	ID/array	Возвращать только оповещения, созданные указанными объектами
userids	ID/array	Возвращать только оповещения сообщений, которые были отправлены указанным пользователям.

Параметр	Тип	Описание
eventobject	integer	Возвращать только оповещения, созданные событиями, связанными с объектами указанного типа. Список поддерживаемых типов объектов см. в `object` события. По умолчанию: 0 - триггер.
eventsource	integer	Возвращать только оповещения, созданные событиями указанного типа. Список поддерживаемых типов событий см. в `source` события. По умолчанию: 0 - события триггера.
time_from	timestamp	Возвращать только оповещения, которые были созданы после указанного времени.
time_till	timestamp	Возвращать только оповещения, которые были созданы до указанного времени.
selectHosts	query	Возвращать свойство `hosts` с данными узлов сети, которые запустили операцию действия.
selectMediatypes	query	Возвращать свойство `mediatypes` с массивом типов медиа, которые использовались для оповещения сообщения. Ограничения в зависимости от типа пользователя см. в `mediatype.get`.
selectUsers	query	Возвращать свойство `users` с массивом пользователей, которым было адресовано сообщение. Ограничения в зависимости от типа пользователя см. в `user.get`.
sortfield	string/array	Сортировать результат по указанным свойствам. Возможные значения: `alertid`, `clock`, `eventid`, `mediatypeid`, `sendto`, `status`.
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	
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   Linux-                5    : ",
            "message": "    :     Zabbix   Linux-                5    : \n          : PROBLEM\n          :
            "status": "0",
            "retries": "3",
            "error": "",
            "esc_step": "1",
            "alerttype": "0",
            "p_eventid": "0",
            "acknowledgeid": "0"
        }
    ],
    "id": 1
}
```

См. также

- Узел сети
- Способ оповещения
- 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 API Zabbix.

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*.

Connector

Этот класс предназначен для работы с коннекторами.

Ссылки на объекты:

- Connector
- Фильтр тегов

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	ID коннектора. Поведение свойства: - *только для чтения* - *обязательно* для операций обновления
name	string	Имя коннектора. Поведение свойства: - *обязательно* для операций создания

Property	Type	Description
url	string	URL конечной точки, то есть URL получателя. Поддерживаются пользовательские макросы. Поведение свойства: - *обязательно* для операций создания
protocol	integer	Протокол обмена данными. 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.
item_value_type	integer	Сумма типов значений элементов данных, которые будут отправлены. Возможные значения: 1 - Числовой (с плавающей точкой); 2 - Символьный; 4 - Журнал; 8 - Числовой (без знака); 16 - Текст; 32 - Бинарный; 64 - JSON. По умолчанию: 31 - Все типы элементов данных (кроме бинарного и JSON). Поведение свойства: - *поддерживается*, если `data_type` установлено в ”Значения элементов данных”.
max_records	integer	Максимальное количество событий или элементов данных, которое может быть отправлено в одном сообщении. Возможные значения: 0-2147483647 (максимальное значение 32-битного 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 значение в секундах. Возможные значения: 0s-10s. По умолчанию: 5s. Поведение свойства: - *поддерживается*, если `max_attempts` больше 1.

Property	Type	Description
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 HTTP(S) прокси в формате *[protocol]://[username[:password]@]proxy.example.com[:port]*. Поддерживаются пользовательские макросы.
authtype	integer	Метод HTTP-аутентификации. 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 `authtype` установлено в ”Basic”, ”NTLM”, ”Kerberos” или ”Digest”
password	string	Пароль. Поддерживаются пользовательские макросы. Поведение свойства: - *поддерживается*, если `authtype` установлено в ”Basic”, ”NTLM”, ”Kerberos” или ”Digest”
token	string	Токен Bearer. Поддерживаются пользовательские макросы. Поведение свойства: - *обязательно*, если `authtype` установлено в ”Bearer”
verify_peer	integer	Следует ли проверять подлинность сертификата узла сети. Возможные значения: 0 - Не проверять; 1 - *(по умолчанию)* Проверять.
verify_host	integer	Следует ли проверять, что имя узла сети для подключения совпадает с именем в сертификате узла сети. Возможные значения: 0 - Не проверять; 1 - *(по умолчанию)* Проверять.
ssl_cert_file	string	Путь к файлу открытого SSL-ключа. Поддерживаются пользовательские макросы.
ssl_key_file	string	Путь к файлу закрытого SSL-ключа. Поддерживаются пользовательские макросы.
ssl_key_password	string	Пароль для файла SSL-ключа. Поддерживаются пользовательские макросы.
description	text	Описание коннектора.
status	integer	Включен ли коннектор. Возможные значения: 0 - Отключен; 1 - *(по умолчанию)* Включен.

Property	Type	Description
tags_evaltype	integer	Метод вычисления 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 экспортировано всё. Объект фильтра тегов имеет следующие свойства.

Свойство	Тип	Описание
tag	string	Имя тега. Поведение свойства: - *обязательно*
operator	integer	Условный оператор. Возможные значения: 0 - *(по умолчанию)* 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 для типа пользователя *Super admin*. Разрешения на вызов метода можно отозвать в настройках роли пользователя. Чтобы получить больше информации смотрите Роли пользователя.

Параметры

`(object/array)` Объекты коннектора для создания.

В дополнение к стандартным свойствам коннектора, метод принимает следующие параметры.

Параметр	Type	Описание
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(    connectorids)`

Этот метод позволяет удалить записи коннектора.

> **Note:**
> Этот метод доступен только пользователю с типом доступа *Супер-администратор*. Разрешения на вызов метода можно отозвать в настройках ролей пользователя. Дополнительную информацию см. в разделе Роли пользователей.

Параметры

`(    )` ID удаляемых коннекторов.

Возвращаемые значения

`(object)` Возвращает объект, содержащий идентификаторы удалённых 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(          )`

Метод позволяет получать объекты коннектора по заданным параметрам.

> **Note:**
> Этот метод доступен только пользователю с типом доступа *Супер-администратор*. Разрешения на вызов метода можно отозвать в настройках ролей пользователя. Дополнительную информацию см. в разделе Роли пользователей.

Параметры

`(object)` Параметры, определяющие желаемый вывод.

Метод поддерживает следующие параметры.

Параметр	Type	Описание
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	

Параметр	Type	Описание
startSearch	boolean	

Возвращаемые значения

`(integer/array)` Возвращает либо:

- массив объектов;
- количество полученных объектов, если был использован параметр `countOutput`.

Примеры

Получение всех коннекторов

Получите 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(  /  connectors)`

Этот метод позволяет обновить существующие коннекторы.

> **Note:**
> Этот метод доступен только пользователю с типом доступа *Супер-администратор*. Разрешения на вызов метода можно отозвать в настройках ролей пользователя. Дополнительную информацию см. в разделе Роли пользователей.

Параметры

`(object/array)` Свойства коннектора, которые необходимо обновить.

Свойство `connectorid` должно быть определено для каждого коннектора, все остальные свойства являются необязательными. Будут обновлены только переданные свойства, все остальные останутся без изменений.

В дополнение к стандартным свойствам коннектора, метод принимает следующие параметры.

Параметр	Type	Описание
tags	array	Фильтр тегов коннектора для замены текущего фильтра тегов.

Возвращаемые значения

(object) Возвращает объект, содержащий идентификаторы обновлённых 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*.

MFA

Этот класс предназначен для работы с методами MFA (многофакторной аутентификации).

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 (многофакторная аутентификация) имеет следующие свойства.

Свойство	Type	Описание
mfaid	ID	ID метода MFA. Поведение свойства: - *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. Поведение свойства: - *обязательно* для операций создания
hash_function	integer	Тип хеш-функции для генерации кодов TOTP. Возможные значения: 1 - SHA-1; 2 - SHA-256; 3 - SHA-512. Поведение свойства: - *обязательно*, если `type` установлен в "TOTP"

Свойство	Type	Описание
code_length	integer	Длина кода подтверждения. Возможные значения: 6 - 6 цифр; 8 - 8 цифр. Поведение свойства: - *обязательно*, если type установлен в ”TOTP”
api_hostname	string	Имя узла API, предоставленное службой аутентификации Duo. Поведение свойства: - *обязательно*, если type установлен в ”Duo Universal Prompt”
clientid	string	ID клиента, предоставленный службой аутентификации Duo. Поведение свойства: - *обязательно*, если type установлен в ”Duo Universal Prompt”
client_secret	string	Секрет клиента, предоставленный службой аутентификации Duo. Поведение свойства: - *только для записи* - *обязательно*, если type установлен в ”Duo Universal Prompt”

mfa.create

Описание

`object mfa.create( / MFA methods)`

Этот метод позволяет создавать новые методы MFA.

> **Note:**
> Этот метод доступен только пользователю с типом доступа *Супер-администратор*. Разрешения на вызов метода можно отозвать в настройках ролей пользователя. Дополнительную информацию см. в разделе Роли пользователей.

Параметры

`(object/array)` Методы MFA для создания.

Метод принимает методы MFA со стандартными свойствами метода MFA.

Возвращаемые значения

`(object)` Возвращает объект, содержащий идентификаторы созданных методов MFA в свойстве `mfaids`. Порядок возвращаемых идентификаторов соответствует порядку переданных элементов данных.

Примеры

Создание методов MFA

Создайте метод MFA ”Zabbix TOTP”, использующий одноразовые пароли на основе времени (TOTP), с хеш-функцией для генерации кодов TOTP, установленной в SHA-1, и длиной кода подтверждения, 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
}
```

Исходный код

CMfa::create() в *ui/include/classes/api/services/CMfa.php*.

mfa.delete

Описание

`object mfa.delete(    mfaids)`

Этот метод позволяет удалять методы MFA.

> **Note:**
> Этот метод доступен только пользователю с типом доступа *Супер-администратор*. Разрешения на вызов метода можно отозвать в настройках ролей пользователя. Дополнительную информацию см. в разделе Роли пользователей.

Параметры

`(array)` Идентификаторы методов MFA для удаления.

Возвращаемые значения

`(object)` Возвращает объект, содержащий идентификаторы удалённых методов MFA в свойстве `mfaids`.

Примеры

Удаление методов MFA

Удалите 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(` `)`

Метод позволяет получать методы MFA по заданным параметрам.

Note:
Этот метод доступен только пользователю с типом доступа *Супер-администратор*. Разрешения на вызов метода можно отозвать в настройках ролей пользователя. Дополнительную информацию см. в разделе Роли пользователей.

Параметры

`(object)` Параметры, определяющие желаемый вывод.

Метод поддерживает следующие параметры.

Параметр	Тип	Описание
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 указаны, 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 одно из следующего:

- массив объектов;

- количество полученных объектов, если был использован параметр `countOutput`.

Примеры

Поиск методов MFA по имени

Получите все методы MFA, в имени которых содержится "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(    /     MFA methods)`

Этот метод позволяет обновлять существующие методы MFA.

> **Note:**
> Этот метод доступен только пользователю с типом доступа *Супер-администратор*. Разрешения на вызов метода можно отозвать в настройках ролей пользователя. Дополнительную информацию см. в разделе Роли пользователей.

Параметры

(object/array) Свойства метода MFA, которые требуется обновить.

Свойство mfaid должно быть определено для каждого элемента, все остальные свойства являются необязательными. 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 обновлённых методов MFA в свойстве mfaids.

Примеры

Обновление свойств метода

Обновите хеш-функцию для генерации TOTP-кодов и длину кода подтверждения 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
}
```

Исходный код

CMfa::update() в *ui/include/classes/api/services/CMfa.php*.

SLA

Этот класс предназначен для работы с объектами SLA (соглашение об уровне обслуживания), используемыми для оценки производительности ИТ-инфраструктуры и бизнес-сервисов.

Ссылки на объекты:

- SLA
- Расписание SLA
- Исключённое время простоя SLA
- Тег сервиса SLA

Доступные методы:

- sla.create - создание новых SLA
- sla.delete - удаление SLA
- sla.get - получение SLA
- sla.getsli - получение данных Service Level Indicator (SLI) для SLA
- sla.update - обновление SLA

Объект SLA

Следующие объекты напрямую связаны с API sla (соглашение об уровне обслуживания).

SLA

Объект SLA имеет следующие свойства.

Свойство	Тип	Описание
slaid	ID	ID SLA. Поведение свойства: - *только для чтения* - *обязательно* для операций обновления
name	string	Имя SLA. Поведение свойства: - *обязательно* для операций создания
period	integer	Отчетный период SLA. Возможные значения: 0 - ежедневно; 1 - еженедельно; 2 - ежемесячно; 3 - ежеквартально; 4 - ежегодно. Поведение свойства: - *обязательно* для операций создания
slo	float	Минимально допустимая целевая доступность сервиса (Service Level Objective, SLO), выраженная в процентах. Если индикатор уровня сервиса (Service Level Indicator, SLI) опускается ниже этого значения, SLA считается находящимся в проблемном/невыполненном состоянии. 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 для отчетности, например: `Europe/London`, `UTC`. Полный список поддерживаемых часовых поясов см. в документации PHP. Поведение свойства: - *обязательно* для операций создания
status	integer	Состояние SLA. Возможные значения: 0 - *(по умолчанию)* SLA отключен; 1 - SLA включен.
description	string	Описание SLA.

Расписание SLA

Объект расписания SLA определяет периоды, в течение которых подключённые сервисы должны находиться в рабочем состоянии. Он имеет следующие свойства.

Свойство	Type	Описание
period_from	integer	Время начала повторяющегося еженедельного периода времени (включительно). Возможные значения: количество секунд (отсчёт от воскресенья). Поведение свойства: - *обязательно*
period_to	integer	Время окончания повторяющегося еженедельного периода времени (не включительно). Возможные значения: количество секунд (отсчёт от воскресенья). Поведение свойства: - *обязательно*

Исключённое время простоя SLA

Объект исключённого времени простоя определяет периоды, в течение которых связанные сервисы планово находятся в нерабочем состоянии, без влияния на SLI, например, во время планового обслуживания. Он имеет следующие свойства.

Свойство	Type	Описание
name	string	Имя исключаемого периода простоя. Поведение свойства: - *обязательно*
period_from	integer	Время начала исключаемого периода простоя (включительно). Возможные значения: timestamp. Поведение свойства: - *обязательно*
period_to	integer	Время окончания исключаемого периода простоя (не включительно). Возможные значения: timestamp. Поведение свойства: - *обязательно*

Тег сервиса SLA

Объект тега сервиса SLA связывает сервисы, которые необходимо включить в расчёты SLA. Он имеет следующие свойства.

Свойство	Type	Описание
tag	string	Имя тега сервиса SLA. Поведение свойства: - *обязательно*
operator	integer	Оператор тега сервиса SLA. Возможные значения: 0 - *(по умолчанию)* Равно; 2 - Содержит.
value	string	Значение тега сервиса SLA.

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 этого метода могут быть отозваны в настройках роли пользователя. См. User roles для получения дополнительной информации.

Параметры

`(object/array)` Объекты SLA для создания.

В дополнение к стандартным свойствам SLA, метод принимает следующие параметры.

Параметр	Type	Описание
service_tags	array	Теги сервиса SLA, которые будут созданы для SLA. Поведение параметра: - *обязательный*
schedule	array	Расписание SLA, которое будет создано для SLA. Указание пустого параметра будет интерпретироваться как расписание 24x7. По умолчанию: расписание 24x7.
excluded_downtimes	array	Исключаемые 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 создать запись SLA для: * отслеживания времени доступности сервисов, связанных с SQL-движком; * пользовательского расписания для всех будних дней, исключая последний час в субботу; * даты вступления в силу — последнего дня 2022 года; * планового простоя длительностью 1 час 15 минут, начинающегося в полночь 4 июля; * включённого расчёта еженедельного отчёта SLA; * минимально допустимого SLO на уровне 99.9995%.

Запрос:

```
{
    "jsonrpc": "2.0",
    "method": "sla.create",
    "params": [
        {
            "name": "                    ",
            "slo": "99.9995",
            "period": "1",
            "timezone": "America/Toronto",
            "description": "                                                    .",
            "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": "                                    ",
                    "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(    slaids)`

Этот метод позволяет удалять записи SLA.

Note:
Этот метод доступен только пользователям с типом доступа *Администратор* и *Супер-администратор*. Разрешения на вызов метода можно отозвать в настройках ролей пользователя. Дополнительную информацию см. в разделе Роли пользователей.

Параметры

`(array)` Идентификаторы SLA для удаления.

Возвращаемые значения

`(object)` Возвращает объект, содержащий идентификаторы удалённых 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(          )`

Метод позволяет получать объекты SLA по заданным параметрам.

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)` Параметры, определяющие желаемый вывод.

Метод поддерживает следующие параметры.

Параметр	Тип	Описание
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 простоями 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 справочном комментарии.
editable	boolean	
excludeSearch	boolean	
filter	object	
limit	integer	
output	query	
preservekeys	boolean	
search	object	

Параметр	Тип	Описание
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 всех SLA

Получение всех данных 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": "                                    SQL-           .",
            "service_tags": [
                {
                    "tag": "database",
                    "operator": "0",
                    "value": "mysql"
                },
                {
                    "tag": "database",
                    "operator": "0",
                    "value": "postgresql"
                }
            ],
            "schedule": [
                {
                    "period_from": "0",
                    "period_to": "601200"
                }
            ],
```

```
            "excluded_downtimes": [
                {
                    "name": "                                    ",
                    "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(          )`

Этот метод позволяет рассчитать данные индикатора уровня обслуживания (SLI) для соглашения об уровне обслуживания (SLA).

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)` Параметры, содержащие ID SLA, отчетные периоды и, при необходимости, ID сервисов, для которых необходимо вычислить SLI.

Параметр	Тип	Описание
slaid	ID	ID SLA, для которого необходимо вернуть информацию о доступности. Поведение параметра: - *обязательный*
period_from	timestamp	Начальная временная метка (включительно), для которой необходимо сформировать отчет по SLI. Возможные значения: Unix timestamp.
period_to	timestamp	Конечная временная метка (включительно), для которой необходимо сформировать отчет по SLI. Возможные значения: Unix timestamp.
periods	integer	Количество периодов для вывода в отчете. Возможные значения: 1-100
serviceids	ID/array	ID сервисов, для которых необходимо вернуть SLI.

Разбиение периодов

В следующей таблице показано расположение возвращаемых срезов 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.

Параметры			Возвращаемые периоды
period_from	**period_to**	**periods**	
-	-	-	Последние 20 периодов, включая текущий.
-	-	N	Последние N периодов.
-	указан	-	Последние 20 периодов до `period_to`.
-	указан	N	Последние N периодов до `period_to`.
указан	-	-	Первые 20 периодов, начиная с `period_from`.
указан	-	N	Первые N периодов, начиная с `period_from`.
указан	указан	-	До 100 периодов в указанном диапазоне.
указан	указан	N	N периодов в указанном диапазоне.

Возвращаемые значения

`(object)` Возвращает результаты вычисления.

Свойство	Type	Описание
periods	array	Список отчётных периодов. Каждый отчётный период представлен объектом, состоящим из: - `period_from` - `(timestamp)` Начальная временная метка (включительно) отчётного периода. - `period_to` - `(timestamp)` Конечная временная метка (не включительно) отчётного периода. Периоды отсортированы по `period_from`, при этом самые ранние периоды идут первыми.
serviceids	array	Список ID сервисов в отчётных периодах. Порядок сортировки списка не определён. Даже если параметр `serviceids` был передан методу `sla.getsli`.
sli	array	Данные SLI (в виде **двумерного массива**) для каждого отчётного 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 отчетного периода и сервиса, включают:

Property	Type	Description
uptime	integer	Количество времени, в течение которого сервис находился в состоянии *OK* в рамках запланированного времени работы, за вычетом исключенных простоев.
downtime	integer	Количество времени, в течение которого сервис находился в состоянии *not OK* в рамках запланированного времени работы, за вычетом исключенных простоев.
sli	float	SLI (в процентах от общего времени работы), на основе времени работы и простоя.
error_budget	integer	Бюджет ошибок (в секундах), на основе SLI и SLO.

Property	Type	Description
excluded_downtimes	array	Массив исключенных простоев в этом отчетном периоде. Каждый объект содержит следующие параметры: - `name` - (`string`) Название исключенного простоя. - `period_from` - (`timestamp`) Временная метка начала (включительно) исключенного простоя. - `period_to` - (`timestamp`) Временная метка окончания (не включительно) исключенного простоя. Исключенные простои отсортированы по `period_from`, при этом самые ранние периоды отображаются первыми.

Примеры

Расчет SLI для ежедневного SLA

Получите данные SLI для сервисов с идентификаторами "1" и "4", связанных с SLA с идентификатором "1". Получите данные за один период до "1761861599" (30 окт. 2025 23:59:59 GMT+0200). Поскольку отчетный период SLA является ежедневным, данные SLI извлекаются с "1761775200" (30 окт. 2025 00:00:00 GMT+0200) до "1761861600" (31 окт. 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 сервисам с идентификаторами ”50”, ”60” и ”70”, которые связаны с SLA с идентификатором ”5”. Получите данные за три периода, начиная с ”1635724800” (Nov 01 2021 00:00:00 UTC). Поскольку отчетный период SLA является ежемесячным, данные SLI извлекаются для следующих трех месяцев:

- 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": []
                }
            ]
        ]
    },
    "id": 1
}
```


Источник

CSla::getSli() в *ui/include/classes/api/services/CSla.php*

sla.update

Описание

`object sla.update(  /  slaids)`

Этот метод позволяет обновлять существующие записи SLA.

Note:
Этот метод доступен только пользователям с типом доступа *Администратор* и *Супер-администратор*. Разрешения на вызов метода можно отозвать в настройках ролей пользователя. Дополнительную информацию см. в разделе Роли пользователей.

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.

Параметр	Тип	Описание
service_tags	array	Теги сервиса SLA для замены текущих тегов сервиса SLA.
schedule	array	Расписание SLA для замены текущего. Указание пустого значения параметра будет интерпретировано как расписание 24x7.
excluded_downtimes	array	Исключенные периоды простоя SLA для замены текущих.

Возвращаемые значения

`(object)` Возвращает объект, содержащий идентификаторы обновлённых SLA в свойстве `slaids`.

Примеры

Обновление тегов сервиса

Сделайте так, чтобы SLA с ID ”5” рассчитывался с ежемесячными интервалами для сервисов, связанных с NoSQL, не изменяя его расписание или исключённые простои; установите SLO равным 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 исключаемых периодов простоя для SLA

Добавьте запланированный 4-часовой простой для обновления RAM 6 апреля 2022 года, при этом сохранив (его необходимо определить заново) ранее существовавшее запланированное обновление программного обеспечения 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 токена имеет следующие свойства.

Свойство	Тип	Описание
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: *текущий пользователь*. Поведение свойства: - *постоянное*
lastaccess	timestamp	Последние дата и время аутентификации токена. ”0”, если токен никогда не использовался для аутентификации. Поведение свойства: - *только для чтения*
status	integer	Статус токена. Возможные значения: 0 - *(по умолчанию)* токен включен; 1 - токен отключен.

Свойство	Тип	Описание
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( / tokens)`

Этот метод позволяет создавать новые токены.

Note:
Для роли пользователя требуется разрешение *Управление токенами API*, чтобы управлять токенами для других пользователей.

Attention:
Токен, созданный этим методом, также должен быть сгенерирован перед использованием.

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, который никогда не истекает и аутентифицирует 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 аутентифицировать 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(    tokenids)`

Этот метод позволяет удалять токены.

> **Note:**
> Для роли пользователя требуется разрешение *Управление токенами API*, чтобы управлять токенами для 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(    tokenids)`

Этот метод позволяет генерировать токены.

Note:
Для роли пользователя требуется разрешение *Управление токенами API*, чтобы управлять токенами для 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 идентификатор сгенерированного токена в свойстве `tokenid` и сгенерированную строку авторизации в свойстве `token`.

Свойство	Type	Описание
tokenid	ID	Идентификатор токена.
token	string	Сгенерированная строка авторизации для этого токена.

Примеры

Сгенерировать несколько токенов

Сгенерируйте 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

```
/    token.get(          )
```

Этот метов позволяет получать токены в соответствии с заданными параметрами.

> **Note:**
> Только пользователю типа *Super admin* разрешено видеть токены других пользователей.

Параметры

`(object)` Параметры, определяющие желаемый вывод.

Метод поддерживает следующие параметры.

Параметр	Тип	Описание
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 указанные дату и время.
expired_at	timestamp	Возвращать только токены, срок действия которых истёк (они недействительны) на указанные 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	
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( / tokens)`

Этот метод позволяет обновлять существующие токены.

> **Note:**
> Для роли пользователя требуется разрешение *Управление токенами API*, чтобы управлять токенами для других пользователей.

Параметры

`(object/array)` Свойства токена, которые необходимо обновить.

Свойство `tokenid` должно быть определено для каждого токена, все остальные свойства являются необязательными. 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*.

Авторегистрация

Этот класс предназначен для работы с авторегистрацией.

Ссылки на объекты:

- Autoregistration

Доступные методы:

- autoregistration.get — получить авторегистрацию
- autoregistration.update — обновить авторегистрацию

Объект авторегистрации

Следующие объекты напрямую связаны с API .

Авторегистрация

Объект авторегистрации имеет следующие свойства.

Свойство	Тип	Описание
tls_accept	integer	Тип разрешённых входящих соединений для авторегистрации. Возможные значения: 1 — разрешить незашифрованные соединения; 2 — разрешить TLS с PSK; 3 — разрешить как незашифрованные соединения, так и TLS с PSK.

Свойство	Тип	Описание
tls_psk_identity	string	Идентификатор PSK; должен быть связан только с одним PSK (среди авторегистрации, узлов сети и прокси). Не включайте конфиденциальную информацию в идентификатор PSK, так как он передаётся по сети в незашифрованном виде, 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( )`

Метод позволяет получить объект авторегистрации по заданным параметрам.

> **Note:**
> Этот метод доступен только пользователю с типом доступа *Супер-администратор*. Разрешения на вызов метода можно отозвать в настройках ролей пользователя. Дополнительную информацию см. в разделе Роли пользователей.

Параметры

`(object)` Параметры, определяющие желаемый вывод.

Метод поддерживает только один параметр.

Параметр	Type	Описание
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( )`

Этот метод позволяет обновить существующую авторегистрацию.

> **Note:**
> Этот метод доступен только пользователю с типом доступа *Супер-администратор*. Разрешения на вызов метода можно отозвать в настройках ролей пользователя. Дополнительную информацию см. в разделе Роли пользователей.

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	HTTP-аутентификация. Возможные значения: 0 - *(по умолчанию)* Отключена; 1 - Включена. Поведение свойства: - *поддерживается*, если `$ALLOW_HTTP_AUTH` включён в файле конфигурации веб-интерфейса (*zabbix.conf.php*).
http_login_form	integer	Форма входа по умолчанию. Возможные значения: 0 - *(по умолчанию)* Форма входа Zabbix; 1 - HTTP-форма входа. Поведение свойства: - *поддерживается*, если `$ALLOW_HTTP_AUTH` включён в файле конфигурации веб-интерфейса (*zabbix.conf.php*).
http_strip_domains	string	Имя домена для удаления. Поведение свойства: - *поддерживается*, если `$ALLOW_HTTP_AUTH` включён в файле конфигурации веб-интерфейса (*zabbix.conf.php*).
http_case_sensitive	integer	HTTP-вход с учётом регистра. Возможные значения: 0 - Выкл.; 1 - *(по умолчанию)* Вкл. Поведение свойства: - *поддерживается*, если `$ALLOW_HTTP_AUTH` включён в файле конфигурации веб-интерфейса (*zabbix.conf.php*).
ldap_auth_enabled	integer	LDAP-аутентификация. Возможные значения: 0 - *(по умолчанию)* Отключена; 1 - Включена.
ldap_case_sensitive	integer	LDAP-вход с учётом регистра. Возможные значения: 0 - Выкл.; 1 - *(по умолчанию)* Вкл.
ldap_userdirectoryid	ID	ID каталога пользователей по умолчанию для LDAP-аутентификации. Используется для групп пользователей, у которых `gui_access` установлен в LDAP или System default. Поведение свойства: - *обязательно*, если `ldap_auth_enabled` установлено в ”Enabled”

Свойство	Тип	Описание
saml_auth_enabled	integer	SAML-аутентификация. Возможные значения: 0 - *(по умолчанию)* Отключена; 1 - Включена.
saml_case_sensitive	integer	SAML-вход с учётом регистра. Возможные значения: 0 - Выкл.; 1 - *(по умолчанию)* Вкл.
passwd_min_length	integer	Требование к минимальной длине пароля. Возможные значения находятся в диапазоне от 1 до 70. По умолчанию: 8.
passwd_check_rules	integer	Правила проверки пароля. Возможные значения битовой маски: 0 - Проверять длину пароля; 1 - Проверять, использует ли пароль латинские буквы верхнего и нижнего регистра; 2 - Проверять, использует ли пароль цифры; 4 - Проверять, использует ли пароль специальные символы; 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 битовой маски; допустима любая сумма возможных значений битовой маски (например, 15 для проверки всех правил).
ldap_jit_status	integer	Статус LDAP-подготовки учётных записей. Возможные значения: 0 - Отключена для настроенных LDAP IdP; 1 - Включена для настроенных LDAP IdP.
saml_jit_status	integer	Статус SAML-подготовки учётных записей. Возможные значения: 0 - Отключена для настроенных SAML IdP; 1 - Включена для настроенных SAML IdP.
jit_provision_interval	string	Интервал времени между запросами JIT-подготовки учётных записей для вошедшего пользователя. Принимает секунды и единицу времени с суффиксом, с поддержкой месяцев и лет (3600s,60m,1h,1d,1M,1y). Минимальное значение: 1h. По умолчанию: 1h. Доступно только для LDAP-подготовки учётных записей.
disabled_usrgrpid	ID	ID группы пользователей, в которую будет назначен пользователь после отмены подготовки учётной записи. Группа пользователей должна быть отключена и не может быть включена или удалена, пока она настроена. Поведение свойства: - *обязательно*, если `ldap_jit_status` установлено в ”Enabled for configured LDAP IdPs” или `saml_jit_status` установлено в ”Enabled for configured SAML IdPs”
mfa_status	integer	Многофакторная аутентификация. Возможные значения: 0 - Отключена (для всех настроенных методов MFA); 1 - Включена (для всех настроенных методов MFA).

Свойство	Тип	Описание
mfaid	ID	Метод MFA по умолчанию для групп пользователей с включённой MFA. Поведение свойства: - *обязательно*, если `mfa_status` установлено в "Enabled"

authentication.get

Описание

`object authentication.get(parameters)`

Метод позволяет получить объект аутентификации 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. Чтобы получить больше информации смотрите Роли пользователя.

Параметры

`(object)` Параметры, определяющие желаемый вывод.

Метод поддерживает только один параметр.

Параметр	Тип	Описание
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

`authentication.update( authentication)`

Этот метод позволяет обновить 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, чтобы получить больше информации.

Параметры

`(object)` Свойства аутентификации подлежат обновлению.

Возвращаемые значения

`(array)` Возвращает массив с именами обновлённых 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
  - HTTP-поле
- 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	Type	Описание
httptestid	ID	ID веб-сценария. Поведение свойства: - *только для чтения* - *обязательно* для операций обновления
hostid	ID	ID узла сети, которому принадлежит веб-сценарий. Поведение свойства: - *постоянное* - *обязательно* для операций создания
name	string	Имя веб-сценария. Поведение свойства: - *обязательно* для операций создания
agent	string	Строка user agent, которая будет использоваться веб-сценарием. По умолчанию: Zabbix
authentication	integer	Метод аутентификации, который будет использоваться веб-сценарием. Возможные значения: 0 - *(по умолчанию)* нет; 1 - базовая HTTP-аутентификация; 2 - NTLM-аутентификация; 3 - Kerberos-аутентификация; 4 - Digest-аутентификация.
delay	string	Интервал выполнения веб-сценария. Допускаются секунды или единица времени с суффиксом (например, 30s, 1m, 2h, 1d), либо пользовательский макрос. По умолчанию: 1m.
headers	array	HTTP-заголовки, которые будут отправлены при выполнении запроса.
http_password	string	Пароль, используемый для базовой HTTP-, NTLM-, Kerberos- или Digest-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-, NTLM-, Kerberos- или Digest-аутентификации.

Свойство	Type	Описание
retries	integer	Количество попыток выполнения каждого шага веб-сценария перед ошибкой. По умолчанию: 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 SSL-ключа, используемого для аутентификации клиента (должен быть в формате PEM).
ssl_key_password	string	Пароль закрытого SSL-ключа.
status	integer	Включен ли веб-сценарий. Возможные значения: 0 - *(по умолчанию)* включен; 1 - отключен.
templateid	ID	ID родительского веб-сценария шаблона. Поведение свойства: - *только для чтения*
variables	array	Переменные веб-сценария.
verify_host	integer	Следует ли проверять, что имя узла сети для соединения совпадает с именем в сертификате узла сети. Возможные значения: 0 - *(по умолчанию)* пропустить проверку узла сети; 1 - проверять узел сети.
verify_peer	integer	Следует ли проверять подлинность сертификата узла сети. Возможные значения: 0 - *(по умолчанию)* пропустить проверку peer; 1 - проверять peer.
uuid	string	Глобальный уникальный идентификатор, используемый для связывания импортированных веб-сценариев с уже существующими. Генерируется 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.

Свойство	Тип	Описание
name	string	Имя шага сценария. Поведение свойства: - *обязательно*
no	integer	Порядковый номер шага в веб-сценарии. Поведение свойства: - *обязательно*
url	string	URL для проверки. Поведение свойства: - *обязательно*
follow_redirects	integer	Следовать ли HTTP-перенаправлениям. Возможные значения: 0 - не следовать перенаправлениям; 1 - *(по умолчанию)* следовать перенаправлениям.

Свойство	Тип	Описание
headers	array	HTTP-заголовки, которые будут отправлены при выполнении запроса. Заголовки шага сценария перезаписывают заголовки, указанные для веб-сценария.
posts	string/array	Переменные HTTP POST в виде строки (необработанные данные post-запроса) или в виде массива HTTP-полей (данные полей формы).
required	string	Текст, который должен присутствовать в ответе.
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 HTTP-кодов состояния, разделенные запятыми.
timeout	string	Тайм-аут запроса в секундах. Допускаются секунды, единица времени с суффиксом или пользовательский макрос. По умолчанию: 15s. Максимум: 1h. Минимум: 1s.
variables	array	Переменные шага сценария.
query_fields	array	Поля запроса — массив HTTP-полей, которые будут добавлены к URL при выполнении запроса.

Поле HTTP

Объект поля HTTP определяет имя и значение, используемые для указания переменных веб-сценария, HTTP-заголовков, а также полей POST или полей запроса. Он имеет следующие свойства.

Свойство	Тип	Описание
name	string	Имя заголовка/переменной/поля POST или GET. Поведение свойства: - *обязательно*
value	string	Значение заголовка/переменной/поля POST или GET. Поведение свойства: - *обязательно*

Тег веб-сценария

Объект тега веб-сценария имеет следующие свойства.

Свойство	Тип	Описание
tag	string	Имя тега веб-сценария. Поведение свойства: - *обязательно*
value	string	Значение тега веб-сценария. Поведение свойства: - *только для чтения*.
object	integer	Тип объекта, от которого был унаследован тег. Возможные значения: 0 - шаблон; 1 - узел сети. Поведение свойства: - *только для чтения*.

Свойство	Тип	Описание
objectid	ID	ID объекта, от которого был унаследован тег. Поведение свойства: - *только для чтения*.

httptest.create

Описание

`object httptest.create(  /  webScenarios)`

Этот метод позволяет создавать новые веб-сценарии.

Note:
При создании веб-сценария также будет автоматически создан набор элементов данных веб-мониторинга.

Note:
Этот метод доступен только пользователям с типом доступа *Администратор* и *Супер-администратор*. Разрешения на вызов метода можно отозвать в настройках ролей пользователя. Дополнительную информацию см. в разделе Роли пользователей.

Параметры

`(object/array)` Веб-сценарии для создания.

В дополнение к стандартным свойствам веб-сценария, метод принимает следующие параметры.

Параметр	Тип	Описание
steps	array	Шаги сценария. Поведение параметра: - *обязательный*
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 для проверки домашней страницы и страницы "About", а также для подтверждения того, что они возвращают код состояния HTTP 200.

Запрос:

```
{
    "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
}
```

Ответ:

```
{
    "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(    webScenarioIds)`

Этот метод позволяет удалять веб-сценарии.

Note:
Этот метод доступен только пользователям с типом доступа *Администратор* и *Супер-администратор*. Разрешения на вызов метода можно отозвать в настройках ролей 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

Удалите два веб-сценария.

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)`

Этот метод позволяет получать веб-сценарии 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 смотрите в разделе Роли пользователей.

Параметры

`(object)` Параметры, определяющие желаемый вывод.

Метод поддерживает следующие параметры.

Параметр	Тип	Описание
groupids	ID/array	Возвращать только веб-сценарии, принадлежащие указанным группам узлов сети.
hostids	ID/array	Возвращать только веб-сценарии, принадлежащие 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 включённые веб-сценарии, принадлежащие отслеживаемым узлам сети.
selectInheritedTags	query	Возвращать свойство `inheritedTags` с тегами, которые присутствуют в шаблоне/узле сети/связанных шаблонах.
templated	boolean	Если установлено значение `true`, возвращать только веб-сценарии, принадлежащие шаблонам.
templateids	ID/array	Возвращать только веб-сценарии, принадлежащие указанным шаблонам.
expandName	flag	Разворачивать макросы в имени веб-сценария.
expandStepName	flag	Разворачивать макросы в именах шагов сценария.
evaltype	integer	Метод вычисления тегов. Возможные значения: 0 — *(по умолчанию)* И/Или; 2 — Или.

Параметр	Тип	Описание
tags	array	Возвращать только веб-сценарии с указанными тегами. Формат: `[{"tag": "<tag>", "value": "<value>", "operator": "<operator>"}, ...]`. Пустой массив возвращает все веб-сценарии. Возможные значения operator: 0 — *(по умолчанию)* Содержит; 1 — Равно; 2 — Не содержит; 3 — Не равно; 4 — Существует; 5 — Не существует.
selectHosts	query	Возвращать узлы сети, 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 одно из следующего:

- массив объектов;
- количество полученных объектов, если был 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(  /  webScenarios)`

Этот метод позволяет обновлять существующие веб-сценарии.

Note:
Этот метод доступен только пользователям с типом доступа *Администратор* и *Супер-администратор*. Разрешения на вызов метода можно отозвать в настройках ролей пользователя. Дополнительную информацию см. в разделе Роли пользователей.

Параметры

`(object/array)` Свойства веб-сценария, которые необходимо обновить.

Свойство `httptestid` должно быть определено для каждого веб-сценария, все остальные свойства являются необязательными. Будут обновлены только переданные свойства, все остальные останутся без изменений.

В дополнение к стандартным свойствам веб-сценария, метод принимает следующие параметры.

Параметр	Type	Описание
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 его статус в "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*.

График

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.

Свойство	Тип	Описание
graphid	ID	ID графика. Поведение свойства: - *только для чтения* - *обязательно* для операций обновления
height	integer	Высота графика в пикселях. Поведение свойства: - *обязательно* для операций создания
name	string	Имя графика. Поведение свойства: - *обязательно* для операций создания
width	integer	Ширина графика в пикселях. Поведение свойства: - *обязательно* для операций создания
flags	integer	Происхождение графика. Возможные значения: 0 - *(по умолчанию)* обычный график; 4 - график, преобразованный из прототипа. Поведение свойства: - *только для чтения*

Свойство	Тип	Описание
graphtype	integer	Тип макета графика. Возможные значения: 0 - *(по умолчанию)* обычный; 1 - с накоплением; 2 - круговой; 3 - разнесённый.
percent_left	float	Левый процентиль. По умолчанию: 0.
percent_right	float	Правый процентиль. По умолчанию: 0.
show_3d	integer	Показывать ли круговые и разнесённые графики в 3D. Возможные значения: 0 - *(по умолчанию)* показывать в 2D; 1 - показывать в 3D.
show_legend	integer	Показывать ли легенду на графике. Возможные значения: 0 - скрыть; 1 - *(по умолчанию)* показывать.
show_work_period	integer	Показывать ли рабочее время на графике. Возможные значения: 0 - скрыть; 1 - *(по умолчанию)* показывать.
show_triggers	integer	Показывать ли линию триггера на графике. Возможные значения: 0 - скрыть; 1 - *(по умолчанию)* показывать.
templateid	ID	ID графика родительского шаблона. Поведение свойства: *- только для чтения*
yaxismax	float	Фиксированное максимальное значение для оси Y. По умолчанию: 100.
yaxismin	float	Фиксированное минимальное значение для оси Y. По умолчанию: 0.
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”.

Свойство	Тип	Описание
ymin_type	integer	Метод вычисления минимального значения для оси Y. Возможные значения: 0 - *(по умолчанию)* calculated; 1 - fixed; 2 - item.
uuid	string	Универсальный уникальный идентификатор, используемый для связывания импортированных графиков с уже существующими. Генерируется автоматически, если не указан. Поведение свойства: - *поддерживается*, если график принадлежит шаблону

graph.create

Описание

`object graph.create( / graphs)`

Этот метод позволяет создавать новые графики.

Note:
Этот метод доступен только пользователям с типом доступа *Администратор* и *Супер-администратор*. Разрешения на вызов метода можно отозвать в настройках ролей пользователя. Дополнительную информацию см. в разделе Роли пользователей.

Параметры

`(object/array)` Графики для создания.

В дополнение к стандартным свойствам графика, метод принимает следующие параметры.

Параметр	Тип	Описание
gitems	array	Элементы графика, которые будут созданы для графика. Поведение параметра: - *обязательный*

Возвращаемые значения

`( )` 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(    graphIds)`

Этот метод позволяет удалять графики.

Note:
Этот метод доступен только пользователям с типом доступа *Администратор* и *Супер-администратор*. Разрешения на вызов метода можно отозвать в настройках ролей 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

Удалите два графика.

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(          )`

Этот метод позволяет получать графики в соответствии с заданными параметрами.

> **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)` Параметры, определяющие желаемый вывод.

Метод поддерживает следующие параметры.

Параметр	Тип	Описание
graphids	ID/array	Возвращать только графики с указанными ID.
groupids	ID/array	Возвращать только графики, принадлежащие узлам сети или шаблонам в указанных группах узлов сети или группах шаблонов.
templateids	ID/array	Возвращать только графики, принадлежащие указанным шаблонам.
hostids	ID/array	Возвращать только графики, принадлежащие указанным узлам сети.
itemids	ID/array	Возвращать только графики, содержащие указанные элементы данных.
templated	boolean	Если установлено в `true`, возвращать только графики, принадлежащие шаблонам.
inherited	boolean	Если установлено в `true`, возвращать только графики, унаследованные от шаблона.
expandName	flag	Развернуть макросы в имени графика.
selectDiscoveryData	query	Возвращать свойство `discoveryData` с данными объекта обнаружения графика. Объект обнаружения графика связывает обнаруженный график с прототипом графика, на основе которого он был обнаружен. Имеет следующие свойства: `parent_graphid` - `(ID)` ID прототипа графика, на основе которого был создан график; `status` - `(int)` статус обнаружения графика: 0 - *(по умолчанию)* график обнаружен, 1 - график больше не обнаруживается; `ts_delete` - `(timestamp)` время, когда график, который больше не обнаруживается, будет удалён.
selectDiscoveryRule	query	Возвращать свойство `discoveryRule` с правилом низкоуровневого обнаружения, создавшим график.

Параметр	Тип	Описание
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 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 график; `hostid` - ID узла сети, которому принадлежит график.
sortfield	string/array	Сортировать результат по указанным свойствам. Возможные значения: `graphid`, `name`, `graphtype`.
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	
selectGraphDiscovery	query	Возвращать свойство `graphDiscovery` с объектом обнаружения графика. Объекты обнаружения графика связывают график с прототипом графика, на основе которого он был создан. Этот запрос **устарел**, используйте вместо него `selectDiscoveryData`.

Возвращаемые значения

(/) Возвращает либо:

- ·массив объектов;
- ·количество извлеченных объектов, если использовался параметр `countOutput`.

Примеры

Получение графиков с узлов сети

Получите все графики с узла сети ”10107” и отсортируйте 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(  /  graphs)`

Этот метод позволяет обновлять существующие графики.

Note:
Этот метод доступен только пользователям с типом доступа *Администратор* и *Супер-администратор*. Разрешения на вызов метода можно отозвать в настройках ролей пользователя. Дополнительную информацию см. в разделе Роли пользователей.

Параметры

(object/array) Свойства графика, которые необходимо обновить.

Свойство graphid должно быть определено для каждого графика, все остальные свойства являются необязательными. Будут обновлены только переданные свойства, все остальные останутся без изменений.

В дополнение к стандартным свойствам графика метод принимает следующие параметры.

Параметр	Type	Описание
gitems	array	Элементы графика для замены существующих элементов графика. Если для элемента графика определено свойство gitemid, он будет обновлён, 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 на 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.

Свойство	Тип	Описание
usrgrpid	ID	ID группы пользователей. Поведение свойства: - *только для чтения* - *обязательно* для операций обновления
name	string	Имя группы пользователей. Поведение свойства: - *обязательно* для операций создания
debug_mode	integer	Включён или отключён режим отладки. Возможные значения: 0 - *(по умолчанию)* отключён; 1 - включён.
gui_access	integer	Метод аутентификации в веб-интерфейсе для пользователей в группе. Возможные значения: 0 - *(по умолчанию)* использовать системный метод аутентификации по умолчанию; 1 - использовать внутреннюю аутентификацию; 2 - использовать LDAP-аутентификацию; 3 - отключить доступ к веб-интерфейсу.
mfa_status	integer	Включена или отключена MFA для пользователей в группе. Возможные значения: 0 - отключена (для всех настроенных методов MFA); 1 - включена (для всех настроенных методов MFA).
mfaid	ID	Метод MFA, используемый для пользователей в группе. Поведение свойства: - *поддерживается* для пользователей типа *Супер-администратор* - *поддерживается*, если `mfa_status` объекта Authentication object установлен в значение ”Enabled”
users_status	integer	Включена или отключена группа пользователей. Для деавтоматизированных пользователей группа пользователей не может быть включена. Возможные значения: 0 - *(по умолчанию)* включена; 1 - отключена.
userdirectoryid	ID	ID каталога пользователей, используемого для аутентификации. Поведение свойства: - *поддерживается* для пользователей типа *Супер-администратор* - *поддерживается*, если `gui_access` установлено в значение ”use the system default authentication method” или ”use LDAP authentication”

Разрешение

Объект разрешения имеет следующие свойства.

Свойство	Тип	Описание
id	ID	ID группы узлов сети или группы шаблонов, для которой добавляется разрешение. Поведение свойства: - *обязательно*
permission	integer	Уровень доступа к группе узлов сети или группе шаблонов. Возможные значения: 0 - доступ запрещён; 2 - доступ только для чтения; 3 - доступ на чтение и запись. Поведение свойства: - *обязательно* для операций создания

Разрешение на основе тегов

Объект разрешения на основе тегов имеет следующие свойства.

Свойство	Тип	Описание
groupid	ID	ID группы узлов сети, для которой добавляется разрешение. Поведение свойства: - *обязательно*
tag	string	Имя тега.
value	string	Значение тега.

usergroup.create

Описание

`object usergroup.create(object/array userGroups)`

Этот метод позволяет создавать новые группы пользователей.

Note:
Этот метод доступен только для типа пользователя *Супер-администратор*. Права на вызов этого метода могут быть отозваны в настройках роли пользователя. Дополнительную информацию смотрите в разделе Роли пользователей.

Параметры

`(object/array)` Группы пользователей для создания.

В дополнение к стандартным свойствам группы пользователей, метод принимает следующие параметры.

Параметр	Тип	Описание
hostgroup_rights	object/array	Права доступа к группам узлов сети, назначаемые группе пользователей.
templategroup_rights	object/array	Права доступа к группам шаблонов, назначаемые группе пользователей.
tag_filters	array	Права доступа на основе тегов, назначаемые группе пользователей.
users	object/array	Пользователи, добавляемые в группу пользователей. Для пользователя должно быть определено 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 запрещённым доступом к группе узлов сети "2" и добавьте в неё 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(    userGroupIds)`

Этот метод позволяет удалять группы пользователей.

> **Note:**
> Этот метод доступен только пользователю с типом доступа *Супер-администратор*. Разрешения на вызов метода можно отозвать в настройках ролей пользователя. Дополнительную информацию см. в разделе Роли пользователей.

Параметры

( ) ID удаляемых групп пользователей.

Возвращаемые значения

( ) Возвращает объект, который содержит ID удалённых групп пользователей, указанных в свойстве usrgrpids.

Примеры

Удаление нескольких групп пользователей

Удалите две группы 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(  )`

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 ролей пользователя. Дополнительную информацию см. в разделе Роли пользователей.

Параметры

`(object)` Параметры, определяющие желаемый вывод.

Метод поддерживает следующие параметры.

Параметр	Тип	Описание
mfaids	ID/array	Возвращать только группы пользователей с указанными методами MFA.
mfa_status	integer	Поведение параметра: - *поддерживается* для пользователей типа *Super admin* Возвращать только группы пользователей с указанным статусом MFA. Список поддерживаемых статусов см. на странице группы пользователей.

Параметр	Тип	Описание
status	integer	Возвращать только группы пользователей с указанным статусом. Список поддерживаемых статусов см. на странице группы пользователей.
userids	ID/array	Возвращать только группы пользователей, содержащие указанных пользователей.
usrgrpids	ID/array	Возвращать только группы пользователей с указанными ID.
selectTagFilters	query	Возвращать разрешения группы 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 `user.get`.
selectHostGroupRights	query	Возвращать разрешения группы пользователей для групп узлов сети в свойстве `hostgroup_rights`. Список уровней доступа к группам узлов сети см. на странице группы пользователей.
selectTemplateGroupRights	query	Возвращать разрешения группы пользователей для групп шаблонов в свойстве `templategroup_rights`. Список уровней доступа к группам шаблонов см. на странице группы пользователей.
limitSelects	integer	Ограничивает количество записей, возвращаемых вложенными выборками.
output	query	Свойства объекта группы пользователей, которые должны быть возвращены. Пользователи типа *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 комментарии.
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 включённых групп пользователей

Получить все включённые 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

- Пользователь

Источник

CUserGroup::get() в *ui/include/classes/api/services/CUserGroup.php*.

usergroup.update

Описание

`object usergroup.update(  /  userGroups)`

Этот метод позволяет обновлять существующие группы пользователей.

> **Note:**
> Этот метод доступен только пользователю с типом доступа *Супер-администратор*. Разрешения на вызов метода можно отозвать в настройках ролей пользователя. Дополнительную информацию см. в разделе Роли пользователей.

Параметры

`(object/array)` Свойства группы 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.

В дополнение к стандартным свойствам группы пользователей, метод принимает следующие параметры.

Параметр	Тип	Описание
hostgroup_rights	object/array	Разрешения группы узлов сети, которые заменят текущие разрешения, назначенные группе пользователей.
templategroup_rights	object/array	Разрешения группы шаблонов, которые заменят текущие разрешения, назначенные группе пользователей.
tag_filters	array	Разрешения на основе тегов, которые заменят текущие разрешения, назначенные группе пользователей.
users	object/array	Пользователи, которые заменят текущих пользователей, назначенных группе пользователей. Для пользователя должно быть определено 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.

Свойство	Тип	Описание
proxy_groupid	ID	ID группы прокси. Поведение свойства: - *только для чтения* - *обязательно* для операций обновления
name	string	Имя группы прокси. Поведение свойства: - *обязательно* 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. По умолчанию: 1m.
min_online	string	Минимальное количество доступных прокси, необходимое для того, чтобы группа прокси считалась доступной. Поддерживаются пользовательские макросы. Диапазон возможных значений: 1-1000. По умолчанию: 1.
state	integer	Состояние группы прокси. Возможные значения: 0 - Неизвестно; 1 - Недоступна; 2 - Восстановление; 3 - Доступна; 4 - Деградация. Поведение свойства: - *только для чтения*

proxygroup.create

Описание

`object proxygroup.create( / proxyGroups)`

Этот метод позволяет создавать новые группы прокси.

Note:
Этот метод доступен только пользователю с типом доступа *Супер-администратор*. Разрешения на вызов метода можно отозвать в настройках ролей 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
}
```

Исходный код

CProxyGroup::create() в *ui/include/classes/api/services/CProxyGroup.php*.

proxygroup.delete

Описание

`object proxygroup.delete(    proxyGroups)`

Этот метод позволяет удалять группы прокси.

> **Note:**
> Этот метод доступен только пользователю с типом доступа *Супер-администратор*. Разрешения на вызов метода можно отозвать в настройках ролей пользователя. Дополнительную информацию см. в разделе Роли пользователей.

Параметры

`(array)` Идентификаторы групп прокси для удаления.

Возвращаемые значения

`(object)` Возвращает объект, содержащий идентификаторы удалённых групп прокси в свойстве `proxy_groupids`.

Примеры

Удаление нескольких групп прокси

Удалите две группы прокси.

Запрос:

```
{
    "jsonrpc": "2.0",
    "method": "proxygroup.delete",
    "params": [
        "5",
        "10"
    ],
```

```
    "id": 1
}
```

Ответ:

```
{
    "jsonrpc": "2.0",
    "result": {
        "proxy_groupids": [
            "5",
            "10"
        ]
    },
    "id": 1
}
```

Исходный код

CProxyGroup::delete() в *ui/include/classes/api/services/CProxyGroup.php*.

proxygroup.get

Описание

`integer/array proxygroup.get(          )`

Метод позволяет получать группы прокси по заданным параметрам.

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)` Параметры, определяющие желаемый вывод.

Метод поддерживает следующие параметры.

Параметр	Тип	Описание
proxy_groupids	ID/array	Возвращать только группы прокси с указанными ID.
proxyids	ID/array	Возвращать только группы прокси, содержащие указанные прокси.
selectProxies	query	Возвращать свойство `proxies` с прокси, принадлежащими 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	
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 одно из следующего:

- массив объектов;
- количество полученных объектов, если использовался параметр `countOutput`.

Примеры

Получение всех групп прокси

Получение всех настроенных групп прокси вместе 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": "Proxy group 1",
            "failover_delay": "1m",
            "min_online": "3",
            "description": "",
            "state": "1",
            "proxies": [
                {
                    "proxyid": "1",
                    "name": "proxy 1"
                },
                {
                    "proxyid": "2",
                    "name": "proxy 2"
                }
            ]
        },
        {
            "proxy_groupid": "2",
            "name": "Proxy group 2",
            "failover_delay": "10m",
            "min_online": "3",
            "description": "",
            "state": "3",
            "proxies": [
                {
                    "proxyid": "3",
                    "name": "proxy 3"
                },
                {
                    "proxyid": "4",
                    "name": "proxy 4"
                },
                {
                    "proxyid": "5",
                    "name": "proxy 5"
                }
            ]
        }
```

```
    ],
    "id": 1
}
```

См. также

- Прокси

Исходный код

CProxyGroup::get() в *ui/include/classes/api/services/CProxyGroup.php*.

proxygroup.update

Описание

`object proxygroup.update(  /  proxyGroups)`

Этот метод позволяет обновлять существующие группы прокси.

Note:
Этот метод доступен только пользователю с типом доступа *Супер-администратор*. Разрешения на вызов метода можно отозвать в настройках ролей пользователя. Дополнительную информацию см. в разделе Роли пользователей.

Параметры

`(object/array)` Свойства группы прокси, которые требуется обновить.

Свойство `proxy_groupid` должно быть определено для каждой группы прокси, все остальные свойства являются необязательными. 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 обновлённых групп прокси в свойстве `proxy_groupids`.

Примеры

Изменение минимального количества прокси в сети

Измените минимальное количество прокси в сети, которые должны быть в сети, чтобы группа считалась находящейся в сети.

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.

Свойство	Тип	Описание
groupid	ID	ID группы узлов сети. Поведение свойства: - *только для чтения* - *обязательно* для операций обновления
name	string	Имя группы узлов сети. Поведение свойства: - *обязательно* для операций создания
flags	integer	Происхождение группы узлов сети. Возможные значения: 0 - обычная группа узлов сети; 4 - обнаруженная группа узлов сети. Поведение свойства: - *только для чтения*
uuid	string	Универсальный уникальный идентификатор, используемый для связывания импортированных групп узлов сети с уже существующими. Генерируется автоматически, если не указан.

hostgroup.create

Описание

`object hostgroup.create( / hostGroups)`

Этот метод позволяет создавать новые группы узлов сети.

> **Note:**
> Этот метод доступен только пользователю с типом доступа *Супер-администратор*. Разрешения на вызов метода можно отозвать в настройках ролей пользователя. Дополнительную информацию см. в разделе Роли пользователей.

Параметры

(object/array) Группы узлов сети для создания.

Метод принимает группы узлов сети со 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 именем "Linux servers".

Запрос:

```
{
    "jsonrpc": "2.0",
    "method": "hostgroup.create",
    "params": {
        "name": "Linux servers"
    },
    "id": 1
}
```

Ответ:

```
{
    "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(hostGroupIds)

Этот метод позволяет удалять группы узлов сети.

Группу узлов сети нельзя удалить в случае, если:

- она содержит узлы сети, которые принадлежат только этой группе узлов сети;
- она отмечена как внутренняя;
- она используется прототипом узлов сети;
- она используется в глобальном скрипте;
- она используется в условии корреляции.

Note:
Этот метод доступен только пользователям с типом доступа *Администратор* и *Супер-администратор*. Разрешения на вызов метода можно отозвать в настройках ролей пользователя. Дополнительную информацию см. в разделе Роли пользователей.

Параметры

( ) ID удаляемых групп узлов сети.

Возвращаемые значения

( ) Возвращает объект, который содержит ID удаленных групп узлов сети, указанных в свойствеgroupids.

Примеры

Удаление нескольких групп узлов сети

Удалите две группы 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)`

Этот метод позволяет получать группы узлов сети в соответствии с заданными параметрами.

> **Note:**
> Этот метод доступен пользователям любого типа. Права на вызов метода могут быть отозваны в настройках роли пользователя. Дополнительную информацию смотрите в разделе Роли пользователей.

Параметры

`(object)` Параметры, определяющие желаемый вывод.

Метод поддерживает следующие параметры.

Параметр	Тип	Описание
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 ID 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.
with_graph_prototypes	boolean	Возвращать только группы узлов сети, которые содержат узлы сети с прототипами графиков.

Параметр	Тип	Описание
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, включёнными для создания и имеющими числовой тип информации.
with_monitored_httptests	boolean	Возвращать только группы узлов сети, которые содержат узлы сети с включёнными веб-проверками.
with_monitored_hosts	boolean	Возвращать только группы узлов сети, которые содержат отслеживаемые узлы сети.
with_monitored_items	boolean	Возвращать только группы узлов сети, которые содержат узлы сети с включёнными элементами данных. Переопределяет параметр `with_simple_graph_items`.
with_monitored_triggers	boolean	Возвращать только группы узлов сети, которые содержат узлы сети с включёнными 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.
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)` ID прототипа группы узлов сети, из которого была обнаружена группа узлов сети; `name` - `(string)` имя прототипа группы узлов сети; `ts_delete` - `(timestamp)` время, когда группа узлов сети, которая больше не обнаруживается, будет удалена; `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, принадлежащими группе узлов сети. Поддерживает `count`.
limitSelects	integer	Ограничивает количество записей, возвращаемых подзапросами. Применяется к следующим подзапросам: `selectHosts` - результаты будут отсортированы по `host`.

Параметр	Тип	Описание
sortfield	string/array	Сортировать результат по указанным свойствам. Возможные значения: `groupid`, `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	
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**, используйте вместо него `selectDiscoveryData`.

Возвращаемые значения

`(integer/array)` Возвращает одно из следующего:

- массив объектов
- количество полученных объектов, если был использован параметр `countOutput`

Примеры

Получение данных по имени

Получите 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(          )`

Этот метод позволяет добавить одновременно несколько связанных объектов во все заданные группы узлов сети.

> **Note:**
> Этот метод доступен только пользователям с типом доступа *Администратор* и *Супер-администратор*. Разрешения на вызов метода можно отозвать в настройках ролей пользователя. Дополнительную информацию см. в разделе Роли пользователей.

Параметры

`(object)` Параметры, содержащие идентификаторы групп узлов сети для обновления и объекты для добавления во все группы узлов сети.

Метод принимает следующие параметры.

Параметр	Type	Описание
groups	object/array	Группы узлов сети, которые необходимо обновить. Для групп узлов сети должно быть определено только свойство `groupid`. Поведение параметров: - *обязательный*
hosts	object/array	Узлы сети для добавления во все группы узлов сети. Для узлов сети должно быть определено 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 идентификаторами 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(          )`

Этот метод позволяет удалить связанные объекты из нескольких групп узлов сети.

> **Note:**
> Этот метод доступен только пользователям с типом доступа *Администратор* и *Супер-администратор*. Разрешения на вызов метода можно отозвать в настройках ролей пользователя. Дополнительную информацию см. в разделе Роли пользователей.

Параметры

`(object)` Параметры, содержащие идентификаторы групп узлов сети для обновления и объекты, которые следует удалить.

Параметр	Type	Описание
groupids	ID/array	Идентификаторы групп узлов сети, которые должны быть обновлены. Поведение параметра: - *обязательный*

Параметр	Type	Описание
hostids	ID/array	Идентификаторы узлов сети, которые следует 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( )`

Этот метод позволяет применять разрешения и фильтры тегов ко всем подгруппам группы хостов.

> **Note:**
> Этот метод доступен только пользователю с типом доступа *Супер-администратор*. Разрешения на вызов метода можно отозвать в настройках ролей пользователя. Дополнительную информацию см. в разделе Роли пользователей.

Параметры

`(object)` Параметры, определяющие желаемый вывод.

Метод поддерживает следующие параметры.

|Параметр|Type|Описание|

|--|--|------|

|groups|object/array|Группы узлов сети для распространения.

Для групп узлов сети должно быть определено свойство `groupid`.

Поведение параметра:
- *обязателен*|

|permissions|boolean|Установите значение "true", чтобы распространить разрешения.

Поведение параметра:
- *обязателен*, если `tag_filters` не задан|

|tag_filters|boolean|Установите значение "true", чтобы распространить фильтры тегов.

Поведение параметра:
- *обязателен*, если `permissions` не задан|

Возвращаемые значения

`(object)` Возвращает объект, содержащий идентификаторы распространённых групп узлов сети в свойстве `groupids`.

Примеры

Распространение прав доступа к группе узлов сети и фильтров тегов на её подгруппы.

Распространите права доступа к группе узлов сети и фильтры тегов на её 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

Исходный код

CHostGroup::propagate() в

ui/include/classes/api/services/CHostGroup.php.

hostgroup.update

Описание

`object hostgroup.update(  /  hostGroups)`

Этот метод позволяет обновлять существующие группы узлов сети.

> **Note:**
> Этот метод доступен только пользователям с типом доступа *Администратор* и *Супер-администратор*. Разрешения на вызов метода можно отозвать в настройках ролей 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 являются необязательными. 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.

Свойство	Тип	Описание
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( / templateGroups)`

Этот метод позволяет создавать новые группы шаблонов.

Note:
Этот метод доступен только пользователю с типом доступа *Супер-администратор*. Разрешения на вызов метода можно отозвать в настройках ролей пользователя. Дополнительную информацию см. в разделе Роли пользователей.

Параметры

`(object/array)` Группы шаблонов для создания. Метод принимает группы шаблонов со 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
}
```

Исходный код

CTemplateGroup::create() в *ui/include/classes/api/services/CTemplateGroup.php*.

templategroup.delete

Описание

`object templategroup.delete(array templateGroupIds)`

Этот метод позволяет удалять группы шаблонов.

Группа шаблонов не может быть удалена, если она содержит шаблоны, которые принадлежат только этой группе.

> **Note:**
> Этот метод доступен только типам пользователей *Admin* и *Super admin*. Разрешения на вызов метода могут быть отозваны в настройках роли пользователя. См. Роли пользователей для получения дополнительной информации.

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(          )`

Метод позволяет получить группы шаблонов в соответствии с заданными параметрами.

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)` Параметры, определяющие желаемый вывод.

Метод поддерживает следующие параметры.

Параметр	Тип	Описание
graphids	ID/array	Возвращать только группы шаблонов, содержащие шаблоны с указанными графиками.
groupids	ID/array	Возвращать только группы шаблонов с указанными ID групп шаблонов.
templateids	ID/array	Возвращать только группы шаблонов, содержащие указанные шаблоны.
triggerids	ID/array	Возвращать только группы шаблонов, содержащие шаблоны с указанными триггерами.
with_graphs	boolean	Возвращать только группы шаблонов, содержащие шаблоны с графиками.
with_graph_prototypes	boolean	Возвращать только группы шаблонов, содержащие шаблоны с прототипами графиков.
with_httptests	boolean	Возвращать только группы шаблонов, содержащие шаблоны с веб-проверками.
with_items	boolean	Возвращать только группы шаблонов, содержащие шаблоны с элементами данных. Переопределяет параметры `with_simple_graph_items`.
with_item_prototypes	boolean	Возвращать только группы шаблонов, содержащие шаблоны с прототипами элементов данных. Переопределяет параметр `with_simple_graph_item_prototypes`.
with_simple_graph_item_prototypes	boolean	Возвращать только группы шаблонов, содержащие шаблоны с прототипами элементов данных, которые включены для создания и имеют числовой тип информации.
with_simple_graph_items	boolean	Возвращать только группы шаблонов, содержащие шаблоны с числовыми элементами данных.
with_templates	boolean	Возвращать только группы шаблонов, содержащие шаблоны.
with_triggers	boolean	Возвращать только группы шаблонов, содержащие шаблоны с триггерами.
selectTemplates	query	Возвращать свойство `templates` с шаблонами, принадлежащими 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 справочном комментарии.
editable	boolean	
excludeSearch	boolean	
filter	object	
limit	integer	
output	query	
preservekeys	boolean	
search	object	
searchByAny	boolean	

Параметр	Тип	Описание
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 данных по имени

Получите 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(          )`

Этот метод позволяет одновременно добавить несколько связанных объектов во все указанные группы шаблонов.

> **Note:**
> Этот метод доступен только пользователям с типом доступа *Администратор* и *Супер-администратор*. Разрешения на вызов метода можно отозвать в настройках ролей пользователя. Дополнительную информацию см. в разделе Роли пользователей.

Параметры

`(object)` Параметры, содержащие идентификаторы групп шаблонов для обновления и объекты для добавления во все группы шаблонов.

Метод принимает следующие параметры.

Параметр	Тип	Описание
groups	object/array	Группы шаблонов, которые необходимо обновить. Для групп шаблонов должно быть определено только свойство `groupid`. Поведение параметра: - *обязательный*
templates	object/array	Шаблоны, которые необходимо добавить во все группы шаблонов. Для шаблонов должно быть определено только свойство `templateid`. Поведение параметра: - *обязательный*

Возвращаемые значения

`(object)` Возвращает объект, содержащий идентификаторы обновлённых 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

Исходный код

CTemplateGroup::massAdd() в *ui/include/classes/api/services/CTemplateGroup.php*.

templategroup.massremove

Описание

`object templategroup.massremove(          )`

Этот метод позволяет удалить связанные объекты из нескольких групп шаблонов.

Note:
Этот метод доступен только пользователям с типом доступа *Администратор* и *Супер-администратор*. Разрешения на вызов метода можно отозвать в настройках ролей пользователя. Дополнительную информацию см. в разделе Роли пользователей.

Параметры

`(object)` Параметры, содержащие идентификаторы групп шаблонов для обновления и объекты, которые должны быть удалены.

Параметр	Тип	Описание
groupids	ID/array	Идентификаторы групп шаблонов, которые должны быть обновлены. Поведение параметра: - *обязательный*
templateids	ID/array	Идентификаторы шаблонов, которые должны быть удалены из всех групп шаблонов. Поведение параметра: - *обязательный*

Возвращаемые значения

`(object)` Возвращает объект, содержащий идентификаторы обновлённых 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
}
```

Исходный код

CTemplateGroup::massRemove() в *ui/include/classes/api/services/CTemplateGroup.php*.

templategroup.propagate

Описание

`object templategroup.propagate(          )`

Этот метод позволяет применить права ко всем подгруппам групп шаблонов.

> **Note:**
> Этот метод доступен только пользователю с типом доступа *Супер-администратор*. Разрешения на вызов метода можно отозвать в настройках ролей пользователя. Дополнительную информацию см. в разделе Роли пользователей.

Параметры

`(object)` Параметры, определяющие желаемый вывод.

Метод поддерживает следующие параметры.

|Параметр|Type|Описание|

|--|--|------|

|groups|object/array|Группы шаблонов, которые необходимо распространить.

Для групп шаблонов должно быть определено только свойство `groupid`.

Поведение параметра:
- *обязательный*|

|permissions|boolean|Установите `true`, если необходимо распространить разрешения.

Поведение параметра:
- *обязательный*|

Возвращаемые значения

`(object)` Возвращает объект, содержащий идентификаторы распространённых групп шаблонов в свойстве `groupids`.

Примеры

Распространение прав доступа группы шаблонов на её подгруппы.

Распространите права доступа группы шаблонов на её 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(  /  templateGroups)`

Этот метод позволяет обновлять существующие группы шаблонов.

> **Note:**
> Этот метод доступен только пользователям с типом доступа *Администратор* и *Супер-администратор*. Разрешения на вызов метода можно отозвать в настройках ролей 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 являются необязательными. Будут обновлены только указанные свойства, все остальные останутся без изменений.

Возвращаемые значения

`(object)` Возвращает объект, содержащий идентификаторы обновлённых групп шаблонов в свойстве `groupids`.

Примеры

Переименование группы шаблонов

Переименуйте 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*.

Действие

Этот класс предназначен для работы с действиями.

Ссылки на объекты:

- Действие
- Операция действия
  - Сообщение операции действия
  - Условие операции действия
- Операция восстановления действия
- Операция обновления действия
- Фильтр действия
  - Условие фильтра действия

Доступные методы:

- action.create - создать новые действия

- action.delete - удалить действия
- action.get - получить действия
- action.update - обновить действия

Action object

The following objects are directly related to the `action` API.

Действие

Объект действия имеет следующие свойства.

Свойство	Тип	Описание
actionid	ID	ID действия. Поведение свойства: - *только для чтения* - *обязательно* для операций обновления
esc_period	string	Длительность шага операции по умолчанию. Должна составлять не менее 60 секунд. Допускаются секунды, единица времени с суффиксом или пользовательский макрос. Поведение свойства: - *поддерживается*, если `eventsource` установлено в значение ”событие, созданное триггером”, ”внутреннее событие” или ”событие, созданное при обновлении состояния сервиса”
eventsource	integer	Тип событий, которые будет обрабатывать действие. Список поддерживаемых типов событий см. в свойстве `source` события. Поведение свойства: - *константа* - *обязательно* для операций создания
name	string	Имя действия. Поведение свойства: - *обязательно* для операций создания
status	integer	Указывает, включено или отключено действие. Возможные значения: 0 - *(по умолчанию)* включено; 1 - отключено.
pause_symptoms	integer	Следует ли приостанавливать эскалацию, если событие является симптомным событием. Возможные значения: 0 - Не приостанавливать эскалацию для симптомных проблем; 1 - *(по умолчанию)* Приостанавливать эскалацию для симптомных проблем. Поведение свойства: - *поддерживается*, если `eventsource` установлено в значение ”событие, созданное триггером”

Свойство	Тип	Описание
pause_suppressed	integer	Следует ли приостанавливать эскалацию во время периодов обслуживания. Возможные значения: 0 - Не приостанавливать эскалацию; 1 - *(по умолчанию)* Приостанавливать эскалацию. Поведение свойства: - *поддерживается*, если `eventsource` установлено в значение ”событие, созданное триггером”
notify_if_canceled	integer	Следует ли отправлять 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` установлено в значение ”событие, созданное триггером”

Операция действия

Объект операции действия определяет операцию, которая будет выполнена при выполнении действия. Он имеет следующие свойства.

Свойство	Тип	Описание
operationtype	integer	Тип операции. Возможные значения: 0 - отправить сообщение; 1 - глобальный скрипт; 2 - добавить узел сети; 3 - удалить узел сети; 4 - добавить в группу узлов сети; 5 - удалить из группы узлов сети; 6 - привязать шаблон; 7 - отвязать шаблон; 8 - включить узел сети; 9 - отключить узел сети; 10 - установить режим инвентаризации узла сети; 13 - добавить теги узла сети; 14 - удалить теги узла сети. Возможные значения, если `eventsource` объекта Action object установлен в ”event created by a trigger” или ”event created on service status update”: 0 - ”send message”; 1 - ”global script”. Возможные значения, если `eventsource` объекта Action object установлен в ”internal event”: 0 - ”send message”. Поведение свойства: - *обязательно*

Свойство	Тип	Описание
esc_period	string	Длительность шага эскалации в секундах. Должна быть больше 60 секунд. Допускаются секунды, единица времени с суффиксом или пользовательский макрос. Если установлено значение 0 или 0s, будет использован период эскалации действия по умолчанию. По умолчанию: 0s. Поведение свойства: - *поддерживается*, если `eventsource` объекта Action object установлен в ”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 object установлен в ”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 object установлен в ”event created by a trigger”, ”internal event” или ”event created on service status update”
evaltype	integer	Метод вычисления условия 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: - *обязательно*, если `operationtype` установлен в ”global script”
opcommand_grp	array	Группы узлов сети, на которых будут выполняться глобальные скрипты. Для групп узлов сети должно быть определено свойство `groupid`. Поведение свойства: - *обязательно*, если `operationtype` установлен в ”global script” и `opcommand_hst` не задан
opcommand_hst	array	Узел сети, на котором будут выполняться глобальные скрипты. Для узлов сети должно быть определено свойство `hostid`. Поведение свойства: - *обязательно*, если `operationtype` установлен в ”global script” и `opcommand_grp` не задан
opconditions	array	Условия операции, используемые для действий триггеров. Объект условия операции подробно описан ниже.

Свойство	Тип	Описание
opgroup	array	Группы узлов сети, в которые добавляются узлы сети. Для групп узлов сети должно быть определено свойство `groupid`. Поведение свойства: - *обязательно*, если `operationtype` установлен в ”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: - *обязательно*, если `operationtype` установлен в ”send message”
opmessage_grp	array	Группы пользователей, которым отправляются сообщения. Для групп пользователей должно быть определено свойство `usrgrpid`. Поведение свойства: - *обязательно*, если `operationtype` установлен в ”send message” и `opmessage_usr` не задан
opmessage_usr	array	Пользователи, которым отправляются сообщения. Для пользователей должно быть определено свойство `userid`. Поведение свойства: - *обязательно*, если `operationtype` установлен в ”send message” и `opmessage_grp` не задан
optemplate	array	Шаблоны для привязки к узлам сети. Для шаблонов должно быть определено свойство `templateid`. Поведение свойства: - *обязательно*, если `operationtype` установлен в ”link template” или ”unlink template”
opinventory	object	Режим инвентаризации, который устанавливается для узла сети. Для инвентаризации должно быть определено свойство `inventory_mode`. Поведение свойства: - *обязательно*, если `operationtype` установлен в ”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 `operationtype` установлен в ”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.

Свойство	Тип	Описание
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 средства оповещения.
mediatypeid	ID	ID типа средства оповещения, который будет использоваться для отправки сообщения. Поведение свойства: - *поддерживается*, если `operationtype` объекта Объект операции действия, Объект операции восстановления действия или Объект операции обновления действия установлен в ”отправить сообщение”, либо если `operationtype` объекта Объект операции обновления действия установлен в ”уведомить всех вовлечённых”
message	string	Текст сообщения операции. Поведение свойства: - *поддерживается*, если `default_msg` установлен в ”использовать данные из операции”
subject	string	Тема сообщения операции. Поведение свойства: - *поддерживается*, если `default_msg` установлен в ”использовать данные из операции”

Условие операции действия

Объект условия операции действия определяет условие, которое должно быть выполнено для выполнения текущей операции. Он имеет следующие свойства.

Свойство	Type	Описание
conditiontype	integer	Тип условия. Возможные значения: 14 - событие подтверждено. Поведение свойства: - *обязательно*
value	string	Значение для сравнения. Поведение свойства: - *обязательно*
operator	integer	Оператор условия. Возможные значения: 0 - *(по умолчанию)* =

Для каждого типа условия операции поддерживаются следующие операторы и значения.

Условие	Название условия	Поддерживаемые операторы	Ожидаемое значение
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 при устранении проблемы. Операции восстановления возможны **только** для действий по триггерам, внутренних действий и действий служб. Он имеет следующие свойства.

Свойство	Type	Описание
operationtype	integer	Тип операции. Возможные значения, если `eventsource` объекта Action object установлен в ”событие, созданное триггером” или ”событие, созданное при обновлении состояния службы”: 0 - отправить сообщение; 1 - глобальный скрипт; 11 - уведомить всех вовлечённых. Возможные значения, если `eventsource` объекта Action object установлен в ”внутреннее событие”: 0 - отправить сообщение; 11 - уведомить всех вовлечённых. Поведение свойства: - *обязательно*
opcommand	object	Глобальный скрипт для выполнения. Для глобального скрипта должно быть определено свойство `scriptid`. Поведение свойства: - *обязательно*, если `operationtype` установлен в ”глобальный скрипт”
opcommand_grp	array	Группы узлов сети, на которых будут выполняться глобальные скрипты. Для групп узлов сети должно быть определено свойство `groupid`. Поведение свойства: - *обязательно*, если `eventsource` объекта Action object установлен в ”событие, созданное триггером”, `operationtype` установлен в ”глобальный скрипт”, и `opcommand_hst` не задан
opcommand_hst	array	Узел сети, на котором будут выполняться глобальные скрипты. Для узлов сети должно быть определено свойство `hostid`. Поведение свойства: - *обязательно*, если `eventsource` объекта Action object установлен в ”событие, созданное триггером”, `operationtype` установлен в ”глобальный скрипт”, и `opcommand_grp` не задан
opmessage	object	Объект, содержащий данные о сообщении, отправляемом операцией восстановления. Объект сообщения операции подробно описан выше. Поведение свойства: - *обязательно*, если `operationtype` установлен в ”отправить сообщение”
opmessage_grp	array	Группы пользователей, которым будут отправлены сообщения. Для групп пользователей должно быть определено свойство `usrgrpid`. Поведение свойства: - *обязательно*, если `operationtype` установлен в ”отправить сообщение” и `opmessage_usr` не задан

Свойство	Type	Описание
opmessage_usr	array	Пользователи, которым будут отправлены сообщения. Для пользователей должно быть определено свойство `userid`. Поведение свойства: - *обязательно*, если `operationtype` установлен в "отправить сообщение" и `opmessage_grp` не задан

Операция обновления действия

Объект операции обновления действия определяет операцию, которая будет выполнена при обновлении проблемы (добавлении комментария, подтверждении, изменении важности или закрытии вручную). Операции обновления возможны **только** для действий триггеров и сервисов. Он имеет следующие свойства.

Свойство	Type	Описание
operationtype	integer	Тип операции. Возможные значения: 0 - отправить сообщение; 1 - глобальный скрипт; 12 - уведомить всех вовлечённых. Поведение свойства: - *required*
opcommand	object	Глобальный скрипт для выполнения. Для глобального скрипта должно быть определено свойство `scriptid`. Поведение свойства: - *required* если `operationtype` установлено в "global script"
opcommand_grp	array	Группы узлов сети, на которых будут выполняться глобальные скрипты. Для групп узлов сети должно быть определено свойство `groupid`. Поведение свойства: - *required* если `eventsource` объекта Action object установлено в "event created by a trigger", `operationtype` установлено в "global script", и `opcommand_hst` не задано
opcommand_hst	array	Узел сети, на котором будут выполняться глобальные скрипты. Для узлов сети должно быть определено свойство `hostid`. Поведение свойства: - *required* если `eventsource` объекта Action object установлено в "event created by a trigger", `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 будут отправлены сообщения. Для групп пользователей должно быть определено свойство `usrgrpid`. Поведение свойства: - *required* если `operationtype` установлено в "send message" и `opmessage_usr` не задано

Свойство	Type	Описание
opmessage_usr	array	Пользователи, которым будут отправлены сообщения. Для 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.

Свойство	Type	Описание
conditions	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*
eval_formula	string	Сгенерированное выражение, которое будет использоваться для вычисления условий фильтра. Выражение содержит идентификаторы, которые ссылаются 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	Определяемое пользователем выражение, используемое для вычисления условий фильтров с пользовательским выражением. Выражение должно содержать идентификаторы, которые ссылаются 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 остаться неиспользованным или пропущенным. Поведение свойства: - *обязательно*, 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 object установлен в значение «событие, созданное триггером»: 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 object установлен в значение «событие, созданное правилом обнаружения»: 7 - IP-адрес узла сети; 8 - тип обнаруженного сервиса; 9 - порт обнаруженного сервиса; 10 - статус обнаружения; 11 - длительность доступности или недоступности; 12 - полученное значение; 18 - правило обнаружения; 19 - проверка обнаружения; 20 - прокси; 21 - объект обнаружения. Возможные значения, если `eventsource` объекта Action object установлен в значение «событие, созданное активной авторегистрацией агента»: 20 - прокси; 22 - имя узла сети; 24 - метаданные узла сети. Возможные значения, если `eventsource` объекта Action object установлен в значение «внутреннее событие»: 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 object установлен в значение «событие, созданное при обновлении состояния сервиса»: 25 - тег события; 26 - значение тега события; 27 - сервис; 28 - имя сервиса. Поведение свойства: - *обязательно*
value	string	Значение для сравнения. Поведение свойства: - *обязательно*

Свойство	Тип	Описание
value2	string	Вторичное значение для сравнения. Поведение свойства: - *обязательно*, если `eventsource` объекта Action object установлен в значение «событие, созданное триггером», `conditiontype` установлен в любое возможное значение для действий по триггерам, а тип условия (см. ниже) — «26» - *обязательно*, если `eventsource` объекта Action object установлен в значение «внутреннее событие», `conditiontype` установлен в любое возможное значение для внутренних действий, а тип условия (см. ниже) — «26» - *обязательно*, если `eventsource` объекта Action object установлен в значение «событие, созданное при обновлении состояния сервиса», `conditiontype` установлен в любое возможное значение для действий сервиса, а тип условия (см. ниже) — «26»
formulaid	string	Произвольный уникальный идентификатор, который используется 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.
operator	integer	Оператор условия. Возможные значения: 0 - *(по умолчанию)* равно; 1 - не равно; 2 - содержит; 3 - не содержит; 4 - в; 5 - больше или равно; 6 - меньше или равно; 7 - не в; 8 - соответствует; 9 - не соответствует; 10 - Да; 11 - Нет.

Note:
Чтобы лучше понять, как использовать фильтры с различными типами выражений, смотрите примеры на страницах методов action.get и action.create.

Для каждого типа условия поддерживаются следующие операторы и значения.

Условие	Название условия	Поддерживаемые операторы	Ожидаемое значение
0	Группа узлов сети	равно, не равно	ID группы узлов сети.
1	Узел сети	равно, не равно	ID узла сети.
2	Триггер	равно, не равно	ID 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 уровней важности триггера см. в свойстве `severity` триггера.
5	Значение триггера	равно	Значение триггера. Список поддерживаемых значений триггера см. в свойстве `value` триггера.
6	Период времени	в, не в	Время, когда было сгенерировано событие, в формате периода времени.

Условие	Название условия	Поддерживаемые операторы	Ожидаемое значение
7	IP-адрес узла сети	равно, не равно	Один или несколько диапазонов IP-адресов, разделённых запятыми. Дополнительную информацию о поддерживаемых форматах диапазонов IP-адресов см. в разделе настройки сетевого обнаружения.
8	Тип обнаруженного сервиса	равно, не равно	Тип обнаруженного сервиса. Тип сервиса соответствует типу проверки обнаружения, использованной для обнаружения сервиса. Список поддерживаемых типов см. в свойстве `type` проверки обнаружения.
9	Порт обнаруженного сервиса	равно, не равно	Один или несколько диапазонов портов, разделённых запятыми.
10	Статус обнаружения	равно	Статус обнаруженного объекта. Возможные значения: 0 - узел сети или сервис доступен; 1 - узел сети или сервис недоступен; 2 - узел сети или сервис обнаружен; 3 - узел сети или сервис потерян.
11	Длительность доступности или недоступности	больше или равно, меньше или равно	Время в секундах, показывающее, как долго обнаруженный объект находится в текущем состоянии.
12	Полученные значения	равно, не равно, больше или равно, меньше или равно, содержит, не содержит	Значение, возвращаемое при выполнении проверки обнаружения Zabbix агент, SNMPv1, SNMPv2 или SNMPv3.
13	Шаблон узла сети	равно, не равно	ID связанного шаблона.
16	Проблема подавлена	Да, Нет	Значение не требуется: использование оператора «Да» означает, что проблема должна быть подавлена, «Нет» — не подавлена.
18	Правило обнаружения	равно, не равно	ID правила обнаружения.
19	Проверка обнаружения	равно, не равно	ID проверки обнаружения.
20	Прокси	равно, не равно	ID прокси.
21	Объект обнаружения	равно	Тип объекта, который вызвал событие обнаружения. Возможные значения: 1 - обнаруженный узел сети; 2 - обнаруженный сервис.
22	Имя узла сети	содержит, не содержит, соответствует, не соответствует	Имя узла сети. Для операторов *соответствует* и *не соответствует* в условиях авторегистрации поддерживается использование регулярного выражения.
23	Тип события	равно	Конкретное внутреннее событие. Возможные значения: 0 - элемент данных в состоянии «не поддерживается»; 1 - элемент данных в состоянии «нормальное»; 2 - правило LLD в состоянии «не поддерживается»; 3 - правило LLD в состоянии «нормальное»; 4 - триггер в состоянии «неизвестно»; 5 - триггер в состоянии «нормальное».
24	Метаданные узла сети	содержит, не содержит, соответствует, не соответствует	Метаданные автоматически зарегистрированного узла сети. Поддерживается использование регулярного выражения 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	Сервис	равно, не равно	ID сервиса.
28	Имя сервиса	равно, не равно	Имя сервиса.

action.create

Описание

`object action.create( / actions)`

Этот метод позволяет создавать новые действия.

Note:
Этот метод доступен только типам пользователей *Администратор* и *Супер-администратор*. Разрешение на использование данного метода можно отозвать в настройках роли пользователя. Дополнительную информацию см. в разделе Роли пользователей

Параметры

`(object/array)` Действия для создания.

В дополнение к стандартным свойствам действия, метод принимает следующие параметры.

Параметр	Тип	Описание
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 авторегистрации агента с тегами узла сети

Создайте действие авторегистрации, 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( actionIds)`

Этот метод позволяет удалять действия.

> **Note:**
> Этот метод доступен только пользователям с типом доступа *Администратор* и *Супер-администратор*. Разрешения на вызов метода могут быть отключены в настройках роли пользователя. Дополнительную информацию см. в разделе Роли пользователей.

Параметры

`( )` 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( parameters)`

Метод позволяет получать действия в соответствии с заданными параметрами.

Note:
Этот метод доступен всем типам пользователей. Разрешение на использование данного метода можно отозвать в настройках роли пользователя. Дополнительную информацию см. в разделе Роли пользователей.

Параметры

`(object)` Параметры, определяющие желаемый вывод.

Метод поддерживает следующие параметры.

Параметр	Тип	Описание
actionids	ID/array	Возвращать только действия с указанными ID.
groupids	ID/array	Возвращать только действия, использующие указанные группы узлов сети в условиях действия.
hostids	ID/array	Возвращать только действия, использующие указанные узлы сети в условиях действия.
triggerids	ID/array	Возвращать только действия, использующие указанные триггеры в условиях действия.
mediatypeids	ID/array	Возвращать только действия, использующие указанные типы медиа для отправки сообщений.
usrgrpids	ID/array	Возвращать только действия, настроенные на отправку сообщений указанным группам пользователей.
userids	ID/array	Возвращать только действия, настроенные на отправку сообщений указанным пользователям.
scriptids	ID/array	Возвращать только действия, настроенные 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

Получение всех настроенных действий триггеров вместе с условиями действий 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": "                        Zabbix",
            "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.NAME}\r\n          : {TRIGGER.STATUS}\r\n          : {TR
                        "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": "     : {TRIGGER.NAME}",
                        "message": "{USER.FULLNAME}             {EVENT.UPDATE.DATE} {EVENT.UPDATE.TIME}
                        "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(  /  actions)`

Этот метод позволяет обновлять существующие действия.

> **Note:**
> Этот метод доступен только для пользователей, имеющих типы *Администратор* и *Супер-администратор*. Права вызывать этот метод могут быть отменены в настройках роли пользователя. Дополнительную информацию см. в разделе Роли пользователей.

Параметры

`(object/array)` Свойства действия для обновления.

Свойство `actionid` должно быть определено для каждого действия, все остальные свойства являются необязательными. Будут обновлены только переданные свойства, все остальные останутся без изменений.

В дополнение к стандартным свойствам действия, метод принимает следующие параметры.

Параметр	Тип	Описание
filter	object	Объект фильтра действия для замены текущего фильтра.
operations	array	Операции действия для замены существующих операций.

Параметр	Тип	Описание
recovery_operations	array	Операции восстановления действия для замены существующих операций восстановления. Поведение параметра: - *поддерживается*, если `eventsource` объекта действия установлен в значение ”событие, созданное триггером”, ”внутреннее событие” или ”событие, созданное при обновлении состояния сервиса”
update_operations	array	Операции обновления действия для замены существующих операций обновления. Поведение параметра: - *поддерживается*, если `eventsource` объекта действия установлен в значение ”событие, созданное триггером” или ”событие, созданное при обновлении состояния сервиса”

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*.

Журнал аудита

Этот класс предназначен для работы с журналом аудита.

Ссылки на объекты:

- Журнал аудита

Доступные методы:

- auditlog.get - получить записи журнала аудита

Audit log object

Следующие объекты напрямую связаны с методом API `auditlog`.

Журнал аудита

Объект журнала аудита содержит информацию о действиях пользователя. Он имеет следующие свойства.

Свойство	Тип	Описание
auditid	ID	ID записи журнала аудита. Генерируется с использованием алгоритма CUID.
userid	ID	userid автора записи журнала аудита.
username	string	имя пользователя автора записи журнала аудита.
clock	timestamp	метка времени создания записи журнала аудита.
ip	string	IP-адрес автора записи журнала аудита.
action	integer	действие записи журнала аудита. Возможные значения: 0 - Добавление; 1 - Обновление; 2 - Удаление; 4 - Выход из системы; 7 - Выполнение; 8 - Вход в систему; 9 - Неудачный вход в систему; 10 - Очистка истории; 11 - Обновление конфигурации; 12 - Отправка.

Свойство	Тип	Описание
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 - Авторегистрация; 39 - Модуль; 40 - Настройки; 41 - Очистка устаревших данных; 42 - Аутентификация; 43 - Панель мониторинга шаблона; 44 - Роль пользователя; 45 - API-токен; 46 - Отчёт по расписанию; 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	понятное человеку имя ресурса записи журнала аудита.
recordsetid	ID	ID набора записей журнала аудита. Записи журнала аудита, созданные в ходе одной и той же операции, будут иметь одинаковый ID набора записей. Генерируется с использованием алгоритма CUID.

Свойство	Тип	Описание
details	text	подробности записи журнала аудита. Подробности хранятся как объект JSON, где каждое имя свойства является путём к свойству или вложенному объекту, в котором произошло изменение, а каждое значение содержит данные (в формате массива) об изменении этого свойства или вложенного объекта. Возможные форматы значений: ["add"] - Вложенный объект был добавлен; ["add", "<value>"] - Свойство добавленного объекта равно <value>; ["update"] - Вложенный объект был обновлён; ["update", "<new value>", "<old value>"] - Свойство обновлённого объекта было изменено с <old value> на <new value>; ["delete"] - Вложенный объект был удалён.

auditlog.get

Описание

`integer/array auditlog.get(object parameters)`

Этот метод позволяет получать записи журнала аудита в соответствии с указанными параметрами.

> **Note:**
> Этот метод доступен только для типа пользователя *Супер-администратор*. Права на вызов этого метода могут быть отозваны в настройках роли пользователя. Дополнительную информацию смотрите в разделе Роли пользователей.

Параметры

`(object)` Параметры, определяющие желаемый вывод.

Метод поддерживает следующие параметры.

Параметр	Тип	Описание
auditids	ID/array	Возвращать только журнал аудита с указанными ID.
userids	ID/array	Возвращать только журнал аудита, созданный указанными пользователями.
time_from	timestamp	Возвращает только записи журнала аудита, созданные после указанного времени или в указанный момент времени.
time_till	timestamp	Возвращает только записи журнала аудита, созданные до указанного времени или в указанный момент времени.
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 одно из следующего:

- массив объектов;

- количество полученных объектов, если был использован параметр `countOutput`.

Примеры

Получение журнала аудита

Получение двух последних записей 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, проверки элементов данных или правил низкоуровневого обнаружения без перезагрузки конфигурации).

Ссылки на объекты:

- Задача
  - Объект запроса 'Выполнить сейчас'
  - Объект запроса 'Обновить конфигурацию прокси'
  - Объект запроса 'Диагностическая информация'
    * 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:

Свойство	Тип	Описание
taskid	ID	ID задачи. Поведение свойства: - *только для чтения*
type	integer	Тип задачи. Возможные значения: 1 - Диагностическая информация; 2 - Обновить конфигурацию прокси; 6 - Выполнить сейчас. Пользователи типа *Admin* и *User* могут создавать только задачи «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 истёк. Поведение свойства: - *только для чтения*
clock	timestamp	Время создания задачи. Поведение свойства: - *только для чтения*

Свойство	Тип	Описание
ttl	integer	Время в секундах, по истечении которого срок действия задачи истекает. Поведение свойства: - *только для чтения*
proxyid	ID	ID прокси, по которому собирается статистика диагностической информации. Поведение свойства: - *поддерживается*, если `type` установлен в ”Diagnostic information” или ”Refresh proxy configuration”
request	object	Объект запроса задачи в соответствии с типом задачи: Объект задачи «Выполнить сейчас» подробно описан ниже; Объект задачи «Обновить конфигурацию 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 ещё не готов. Объект результата подробно описан ниже. Поведение свойства: - *только для чтения*

Объект запроса ’Выполнить сейчас’

Объект запроса задачи ’Выполнить сейчас’ имеет следующие свойства.

Свойство	Тип	Описание
itemid	ID	ID элемента данных и правил обнаружения низкого уровня. Пользователи типа *Admin* и *User* могут выполнять действие ’Выполнить сейчас’ для элементов данных на узлах сети, для которых у них есть разрешение *чтение-запись* или разрешение *чтение* и для их роли включено действие `invoke_execute_now`. То же самое относится к пользователям типа *Admin* для правил обнаружения низкого уровня (LLD).

Объект запроса ’Refresh proxy configuration’

Объект запроса задачи ’Refresh proxy configuration’ имеет следующие свойства.

Свойство	Тип	Описание
proxyids	array	ID прокси.

Объект запроса ’Diagnostic information’

Объект запроса задачи диагностической информации имеет следующие свойства. Объект запроса статистики для всех типов свойств подробно описан ниже.

Свойство	Тип	Описание
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 предобработки. Доступно на сервере и прокси.

Свойство	Тип	Описание
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.

Свойство	Тип	Описание
stats	query	Свойства объекта статистики, которые должны быть возвращены. 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 статистических полей, доступных для каждого типа запроса диагностической информации

Следующие статистические поля могут быть запрошены для каждого типа свойства запроса диагностической информации.

Тип диагностики	Доступные поля	Описание
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 данных истории.
	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 предобработки.
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.

Тип диагностики	Доступные поля	Тип
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 возвращается в поле `result` объекта задачи.

Свойство	Type	Описание
status	integer	Статус результата задачи. Возможные значения: -1 - во время выполнения 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 во время выполнения задачи произошла ошибка.

task.create

Описание

`object task.create(object/array tasks)`

Этот метод позволяет создавать задачи.

> **Note:**
> Этот метод доступен пользователям любого типа. Права на вызов метода могут быть отозваны в настройках роли пользователя. Подробнее см. в разделе Роли пользователей.

Параметры

`(object/array)` Задача для создания.

Метод принимает задачи со стандартными свойствами задачи.

Обратите внимание, что задачи типа «Выполнить сейчас» могут быть созданы только для следующих типов элементов данных/правил обнаружения:

- Zabbix агент (пассивный)
- Простая проверка
- SNMP агент (v1/v2/v3)
- Внутренняя проверка Zabbix
- Внешняя проверка
- Монитор базы данных
- HTTP агент
- IPMI агент
- SSH агент
- TELNET агент
- JMX агент
- Вычисляемый
- Зависимый элемент данных
- Скрипт
- Браузер

Если элемент данных/правило обнаружения имеет тип «Зависимый элемент данных», то его мастер-элемент данных также должен быть одним из перечисленных 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 'Execute now' для элемента данных и правила обнаружения низкого уровня.

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 'Refresh proxy configuration' 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 'Diagnostic information'.

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 'Выполнить сейчас'
- Объект запроса 'Обновить конфигурацию прокси'
- Объект запроса 'Диагностическая информация'

Источник

CTask::create() в *ui/include/classes/api/services/CTask.php*.

task.get

Описание

`integer/array task.get(object parameters)`

Этот метод позволяет получать задачи в соответствии с заданными параметрами.

> **Note:**
> Этот метод доступен только для типа пользователя *Супер-администратор*. Разрешения на вызов метода могут быть отозваны в настройках роли пользователя. Подробнее см. в разделе Роли пользователей.

Параметры

(object) Параметры, определяющие требуемый вывод.

Метод поддерживает следующие параметры.

Параметр	Тип	Описание
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

Получение всех данных 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 'Выполнить сейчас'
- Объект запроса 'Обновить конфигурацию прокси'
- Объект запроса 'Диагностическая информация'

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 — создание новых изображений
- image.delete — удаление изображений
- image.get — получение изображений
- image.update — обновление изображений

Объект image

Следующие объекты напрямую связаны с API `image`.

Изображение

Объект изображения имеет следующие свойства.

Свойство	Тип	Описание
imageid	ID	ID изображения. Поведение свойства: - *только для чтения* - *обязательно* для операций обновления
name	string	Имя изображения. Поведение свойства: - *обязательно* для операций создания

Свойство	Тип	Описание
imagetype	integer	Тип изображения. Возможные значения: 1 - *(по умолчанию)* значок; 2 - фоновое изображение. Поведение свойства: - *постоянное* - *обязательно* для операций создания
image	string	Изображение, закодированное в Base64. Максимальный размер закодированного изображения составляет 1 МБ. Максимальный размер можно изменить, изменив значение константы ZBX_MAX_IMAGE_SIZE. Поддерживаемые форматы изображений: PNG, JPEG, GIF и WebP. Поведение свойства: - *обязательно* для операций создания

image.create

Описание

`object image.create(  /  images)`

Этот метод позволяет создавать новые изображения.

> **Note:**
> Этот метод доступен только пользователю с типом доступа *Супер-администратор*. Разрешения на вызов метода можно отозвать в настройках ролей пользователя. Дополнительную информацию см. в разделе Роли пользователей.

Параметры

`(object/array)` Изображения для создания.

Метод принимает изображения со стандартными свойствами изображения.

Возвращаемые значения

`(  )` 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 как шаблон.

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(    imageIds)`

Этот метод позволяет удалять изображения.

> **Note:**
> Этот метод доступен только пользователю с типом доступа *Супер-администратор*. Разрешения на вызов метода можно отозвать в настройках ролей пользователя. Дополнительную информацию см. в разделе Роли пользователей.

Параметры

( ) ID удаляемых изображений.

Возвращаемые значения

( ) Возвращает объект, который содержит ID удаленных изображений, указанных в свойстве `imageids`.

Примеры

Удаление нескольких изображений

Удалите два изображения.

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(          )`

Этот метод позволяет получать изображения в соответствии с заданными параметрами.

> **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)` Параметры, определяющие желаемый вывод.

Метод поддерживает следующие параметры.

Параметр	Тип	Описание
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	
output	query	
preservekeys	boolean	
search	object	
searchByAny	boolean	
searchWildcardsEnabled	boolean	
sortorder	string/array	

Параметр	Тип	Описание
startSearch	boolean	

Возвращаемые значения

(/) Возвращает либо:

- массив объектов;
- количество извлеченных объектов, если использовался параметр `countOutput`.

Примеры

Получение изображения

Получение всех данных об изображении 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( / images)`

Этот метод позволяет обновлять существующие изображения.

Note:
Этот метод доступен только пользователю с типом доступа *Супер-администратор*. Разрешения на вызов метода можно отозвать в настройках ролей пользователя. Дополнительную информацию см. в разделе Роли пользователей.

Параметры

`(object/array)` Свойства изображения, которые необходимо обновить.

Свойство `imageid` должно быть определено для каждого изображения, все остальные свойства являются необязательными. 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

Переименование изображения

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 предназначен для работы с интерфейсами узла сети.

Ссылки на объекты:

- Интерфейс узла сети
  - Сведения

Доступные методы:

- hostinterface.create - создать новые интерфейсы узла сети
- hostinterface.delete - удалить интерфейсы узла сети
- hostinterface.get - получить интерфейсы узла сети
- hostinterface.massadd - добавить интерфейсы узла сети к узлам сети
- hostinterface.massremove - удалить интерфейсы узла сети с узлов сети
- hostinterface.update - обновить интерфейсы узла сети

Объект интерфейса узла сети

Следующие объекты напрямую связаны с API `hostinterface`.

Интерфейс узла сети

Объект интерфейса узла сети имеет следующие свойства.

> **Attention:**
> Обратите внимание, что свойства `ip` и `dns` *обязательны* для операций создания. Если вы не хотите использовать DNS, установите для него пустую строку.

Свойство	Type	Описание
interfaceid	ID	ID 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	ID 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	DNS-имя, 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	Следует ли выполнять подключение через IP. Возможные значения: 0 - подключаться, используя DNS-имя узла сети; 1 - подключаться, используя IP-адрес узла сети. Поведение свойства: - *обязательно* для операций создания

Свойство	Type	Описание
main	integer	Используется ли интерфейс как интерфейс по умолчанию на узле сети. Только один интерфейс определенного типа может быть установлен как интерфейс по умолчанию на узле сети. Возможные значения: 0 - не по умолчанию; 1 - по умолчанию. Поведение свойства: - *обязательно* для операций создания
details	object	Дополнительный объект details для интерфейса. Поведение свойства: - *обязательно*, если `type` установлен в ”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.

Свойство	Тип	Описание
version	integer	Версия интерфейса SNMP. Возможные значения: 1 - SNMPv1; 2 - SNMPv2c; 3 - SNMPv3. Поведение свойства: - *обязательно*
bulk	integer	Следует ли использовать групповые SNMP-запросы. Возможные значения: 0 - не использовать групповые запросы; 1 - (по умолчанию) - использовать групповые запросы.
community	string	Community SNMP. Используется только интерфейсами SNMPv1 и SNMPv2. Поведение свойства: - *обязательно*, если `version` установлено в ”SNMPv1” или ”SNMPv2c”
max_repetitions	integer	Максимальное значение повторений для встроенных групповых SNMP-запросов (GetBulkRequest-PDUs). Используется только для элементов данных `discovery[]` и `walk[]` в SNMPv2 и v3. По умолчанию: 10.
securityname	string	Имя безопасности SNMPv3. Используется только интерфейсами SNMPv3.

Свойство	Тип	Описание
securitylevel	integer	Уровень безопасности SNMPv3. Используется только интерфейсами SNMPv3. Возможные значения: 0 - (по умолчанию) - noAuthNoPriv; 1 - authNoPriv; 2 - authPriv.
authpassphrase	string	Парольная фраза аутентификации SNMPv3. Используется только интерфейсами SNMPv3.
privpassphrase	string	Парольная фраза приватности 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 приватности 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.create

Описание

`object hostinterface.create(  /  hostInterfaces)`

Этот метод позволяет создавать новые интерфейсы узлов сети.

Note:
Этот метод доступен только пользователям с типом доступа *Администратор* и *Супер-администратор*. Разрешения на вызов метода можно отозвать в настройках ролей пользователя. Дополнительную информацию см. в разделе Роли пользователей.

Параметры

`(object/array)` Интерфейсы узла сети для создания.

Метод принимает интерфейсы узла сети со стандартными свойствами интерфейса узла сети.

Возвращаемые значения

`(  )` Возвращает объект, который содержит ID созданных интерфейсов узлов сети, указанных в свойстве `interfaceids`. Порядок возвращаемых ID совпадает с порядком переданных интерфейсов узлов сети.

Примеры

Создание нового интерфейса

Создайте дополнительный IP-интерфейс агента на узле сети "30052."

Запрос:

```
{
    "jsonrpc": "2.0",
    "method": "hostinterface.create",
    "params": {
        "hostid": "30052",
        "main": "0",
        "type": "1",
        "useip": "1",
        "ip": "127.0.0.1",
        "dns": "",
        "port": "10050"
    },
    "id": 1
}
```

Ответ:

```
{
    "jsonrpc": "2.0",
    "result": {
        "interfaceids": [
            "30062"
        ]
    },
    "id": 1
}
```

Создание интерфейса с данными SNMP

Запрос:

```
{
    "jsonrpc": "2.0",
    "method": "hostinterface.create",
    "params": {
        "hostid": "10456",
        "main": "0",
        "type": "2",
        "useip": "1",
        "ip": "127.0.0.1",
        "dns": "",
        "port": "1601",
        "details": {
            "version": "2",
            "bulk": "1",
            "community": "{$SNMP_COMMUNITY}"
        }
    },
    "id": 1
}
```

Ответ:

```
{
    "jsonrpc": "2.0",
    "result": {
        "interfaceids": [
            "30063"
        ]
    },
    "id": 1
}
```

Смотрите также

- ·hostinterface.massadd

- ·host.massadd

Источник

CHostInterface::create() в *ui/include/classes/api/services/CHostInterface.php*.

hostinterface.delete

Описание

`object hostinterface.delete( hostInterfaceIds)`

Этот метод позволяет удалять интерфейсы узлов сети.

> **Note:**
> Этот метод доступен только пользователям с типом доступа *Администратор* и *Супер-администратор*. Разрешения на вызов метода можно отозвать в настройках ролей пользователя. Дополнительную информацию см. в разделе Роли пользователей.

Параметры

( ) ID удаляемых интерфейсов узлов сети.

Возвращаемые значения

( ) Возвращает объект, который содержит ID удаленных интерфейсов узлов сети, указанных в свойстве `interfaceids`.

Примеры

Удаление интерфейса узла сети

Удалите интерфейс узла сети с ID 30062.

Запрос:

```
{
    "jsonrpc": "2.0",
    "method": "hostinterface.delete",
    "params": [
        "30062"
    ],
    "id": 1
}
```

Ответ:

```
{
    "jsonrpc": "2.0",
    "result": {
        "interfaceids": [
            "30062"
        ]
    },
    "id": 1
}
```

Смотрите также

- hostinterface.massremove
- host.massremove

Источник

CHostInterface::delete() в *ui/include/classes/api/services/CHostInterface.php*.

hostinterface.get

Описание

`integer/array hostinterface.get( )`

Этот метод позволяет получать интерфейсы узлов сети в соответствии с заданными параметрами.

> **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) Параметры, определяющие желаемый вывод.

Метод поддерживает следующие параметры.

Параметр	Тип	Описание
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.
selectItems	query	Возвращать свойство `items` с элементами данных, использующими интерфейс. Поддерживает `count`.
selectHosts	query	Возвращать свойство `hosts` с массивом узлов сети, использующих интерфейс.
limitSelects	integer	Ограничивает количество записей, возвращаемых подзапросами. Применяется к следующим подзапросам: `selectItems`.
sortfield	string/array	Сортировать результат по указанным свойствам. Возможные значения: `interfaceid`, `dns`, `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

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*.

hostinterface.massadd

Описание

`object hostinterface.massadd(          )`

Этот метод позволяет добавить одновременно несколько интерфейсов узлов сети на узлы сети.

Note:
Этот метод доступен только пользователям с типом доступа *Администратор* и *Супер-администратор*. Разрешения на вызов метода можно отозвать в настройках ролей пользователя. Дополнительную информацию см. в разделе Роли пользователей.

Параметры

`(object)` Параметры, содержащие интерфейсы узлов сети, которые должны быть созданы на указанных узлах сети.

Метод принимает следующие параметры.

Параметр	Тип	Описание
interfaces	object/array	Интерфейсы узлов сети, которые должны быть созданы на указанных узлах сети. Поведение параметра: - *обязательный*
hosts	object/array	Узлы сети, которые должны быть обновлены. Для узлов сети должно быть определено только свойство `hostid`. Поведение параметра: - *обязательный*

Возвращаемые значения

( ) Возвращает объект, который содержит ID созданных интерфейсов узла сети, указанных в свойстве `interfaceids`.

Примеры

Создание интерфейсов

Создайте интерфейс на двух узлах сети.

Запрос:

```
{
    "jsonrpc": "2.0",
    "method": "hostinterface.massadd",
    "params": {
        "hosts": [
            {
                "hostid": "30050"
            },
            {
                "hostid": "30052"
            }
        ],
        "interfaces": {
            "dns": "",
            "ip": "127.0.0.1",
            "main": 0,
            "port": "10050",
            "type": 1,
            "useip": 1
        }
    },
    "id": 1
}
```

Ответ:

```
{
    "jsonrpc": "2.0",
    "result": {
        "interfaceids": [
```

```
            "30069",
            "30070"
        ]
    },
    "id": 1
}
```

Смотрите также

- hostinterface.create
- host.massadd
- Узел сети

Источник

CHostInterface::massAdd() в *ui/include/classes/api/services/CHostInterface.php*.

hostinterface.massremove

Описание

`object hostinterface.massremove(  )`

Этот метод позволяет удалить интерфейсы узлов сети у заданных узлов сети.

Note:
Этот метод доступен только пользователям с типом доступа *Администратор* и *Супер-администратор*. Разрешения на вызов метода можно отозвать в настройках ролей пользователя. Дополнительную информацию см. в разделе Роли пользователей.

Параметры

`(object)` Параметры, содержащие идентификаторы узлов сети, которые необходимо обновить, и интерфейсы, которые необходимо удалить.

Параметр	Тип	Описание
interfaces	object/array	Интерфейсы узла сети, которые необходимо удалить с указанных узлов сети. В объекте интерфейса узла сети должны быть определены только свойства `ip`, `dns` и `port`. Поведение параметра: - *обязательный*
hostids	ID/array	Идентификаторы узлов сети, которые необходимо обновить. Поведение параметра: - *обязательный*

Возвращаемые значения

`(  )` Возвращает объект, который содержит ID удаленных интерфейсов узлов сети, указанных в свойстве `interfaceids`.

Примеры

Удаление интерфейсов

Удалите SNMP-интерфейс "127.0.0.1" с двух узлов сети.

Запрос:

```
{
    "jsonrpc": "2.0",
    "method": "hostinterface.massremove",
    "params": {
        "hostids": [
            "30050",
```

```
            "30052"
        ],
        "interfaces": {
            "dns": "",
            "ip": "127.0.0.1",
            "port": "161"
        }
    },
    "id": 1
}
```

Ответ:

```
{
    "jsonrpc": "2.0",
    "result": {
        "interfaceids": [
            "30069",
            "30070"
        ]
    },
    "id": 1
}
```

Смотрите также

- hostinterface.delete
- host.massremove

Источник

CHostInterface::massRemove() в *ui/include/classes/api/services/CHostInterface.php*.

hostinterface.update

Описание

`object hostinterface.update(  /  hostInterfaces)`

Этот метод позволяет обновлять существующие интерфейсы узлов сети.

Note:
Этот метод доступен только пользователям с типом доступа *Администратор* и *Супер-администратор*. Разрешения на вызов метода можно отозвать в настройках ролей пользователя. Дополнительную информацию см. в разделе Роли пользователей.

Параметры

`(object/array)` Свойства интерфейса узла сети, которые необходимо обновить.

Свойство `interfaceid` должно быть определено для каждого интерфейса узла сети, все остальные свойства являются необязательными. Будут обновлены только указанные свойства, все остальные останутся без изменений.

Возвращаемые значения

`(  )` Возвращает объект, который содержит ID обновленных интерфейсов узлов сети, указанных в свойстве `interfaceids`.

Примеры

Изменение порта интерфейса узла сети

Измените порт интерфейса узла сети.

Запрос:

```
{
    "jsonrpc": "2.0",
    "method": "hostinterface.update",
    "params": {
```

```
        "interfaceid": "30048",
        "port": "10055"
    },
    "id": 1
}
```

Ответ:

```
{
    "jsonrpc": "2.0",
    "result": {
        "interfaceids": [
            "30048"
        ]
    },
    "id": 1
}
```

Изменение портов нескольких интерфейсов узла сети

Измените порт нескольких интерфейсов узла сети.

Запрос:

```
{
    "jsonrpc": "2.0",
    "method": "hostinterface.update",
    "params": [
        {
            "interfaceid": "30049",
            "port": "10055"
        },
        {
            "interfaceid": "30050",
            "port": "10055"
        },
        {
            "interfaceid": "30051",
            "port": "10055"
        }
    ],
    "id": 1
}
```

Ответ:

```
{
    "jsonrpc": "2.0",
    "result": {
        "interfaceids": [
            "30049",
            "30050",
            "30051"
        ]
    },
    "id": 1
}
```

Источник

CHostInterface::update() в *ui/include/classes/api/services/CHostInterface.php*.

История

Этот класс предназначен для работы с данными истории.

Ссылки на объекты:

- История с плавающей точкой
- Целочисленная история
- Строковая история
- Текстовая история
- История журнала

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 значений с плавающей точкой

Объект истории значений с плавающей точкой 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	float	Полученное значение.

История целых чисел

Объект истории целых чисел 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	integer	Полученное значение.

История строк

Объект истории строк имеет следующие свойства.

Свойство	Type	Описание
clock	timestamp	Время, когда это значение было получено.
itemid	ID	ID связанного элемента данных.
ns	integer	Наносекунды, когда значение было получено.
value	string	Полученное значение.

История текста

Объект истории текста имеет следующие свойства.

Свойство	Тип	Описание
id	ID	ID записи истории.
clock	timestamp	Время, когда это значение было получено.
itemid	ID	ID связанного элемента данных.
ns	integer	Наносекунды, когда значение было получено.

Свойство	Тип	Описание
value	text	Полученное значение.

История журнала

Объект истории журнала имеет следующие свойства.

Свойство	Тип	Описание
id	ID	ID записи истории.
clock	timestamp	Время, когда это значение было получено.
itemid	ID	ID связанного элемента данных.
logeventid	integer	ID записи журнала событий Windows.
ns	integer	Наносекунды, когда значение было получено.
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.

Бинарная история

Объект бинарной истории имеет следующие свойства.

Свойство	Type	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.

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(    itemids)`

Этот метод позволяет очистить историю элемента.

Note:
Этот метод доступен только пользователям с типом доступа *Администратор* и *Супер-администратор*. Разрешения на вызов метода можно отозвать в настройках ролей 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(          )`

Метод позволяет получить данные истории по заданным параметрам.

Attention:
Этот метод может возвращать исторические данные удаленного объекта, если эти данные еще не были удалены служебной программой.

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)` Параметры, определяющие желаемый вывод.

Метод поддерживает следующие параметры.

Параметр	Тип	Описание
history	integer	Типы объектов истории для возврата. Возможные значения: 0 - числовое с плавающей точкой; 1 - символьное; 2 - журнал; 3 - *(по умолчанию)* числовое без знака; 4 - текст; 5 - двоичное; 6 - JSON.
hostids	ID/array	Возвращать только историю для указанных узлов сети.
itemids	ID/array	Возвращать только историю для указанных элементов данных.

Параметр	Тип	Описание
maxValueSize	integer	Максимальное количество байт, возвращаемых в свойстве `value`. Диапазон возможных значений: 1-134217728 (от 1 байта до 128MiB). Если установлено значение `null`, ограничение не применяется. По умолчанию: 65536 (64KiB). Поведение параметра: - *поддерживается*, если `history` установлено в "binary" или "JSON"
time_from	timestamp	Возвращать только значения, полученные после указанного времени или в указанный момент.
time_till	timestamp	Возвращать только значения, полученные до указанного времени или в указанный момент.
sortfield	string/array	Сортировать результат по указанным свойствам. Возможные значения: `itemid`, `clock`, `ns`.
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 исторических данных элемента данных

Вернуть 10 последних значений, полученных от числового элемента данных (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.

Параметр	Тип	Описание
itemid	ID	ID связанного элемента данных. Поведение параметра: - *обязателен*, если `host` и `key` не заданы
host	string	Техническое имя узла сети. Поведение параметра: - *обязателен*, если `itemid` не задан
key	string	Ключ элемента данных. Поведение параметра: - *обязателен*, если `itemid` не задан
value	mixed	Значение элемента данных. Поведение параметра: - *обязателен*
clock	timestamp	Время получения значения.
ns	integer	Наносекунды в момент получения значения.

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 агент
- Узел сети
- Элемент данных

Исходный код

CHistory::push() в *ui/include/classes/api/services/CHistory.php*.

Карта

Этот класс предназначен для работы с картами.

Ссылки на объекты:

- Карта
- Элемент карты
  - Элемент карты Узел сети
  - Элемент карты Группа узлов сети
  - Элемент карты Карта
  - Элемент карты Триггер
  - Элемент карты Тег
  - Элемент карты URL
- Связь карты
  - Триггер связи карты
  - Индикаторы связи карты
- URL карты
- Пользователь карты
- Группа пользователей карты
- Фигуры карты
- Линии карты

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.

Свойство	Тип	Описание
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	Следует ли включить пропорциональное масштабирование фонового изображения. Возможные значения: 0 - отключить масштабирование; 1 - *(по умолчанию)* включить масштабирование изображения.
expand_macros	integer	Следует ли раскрывать макросы в метках при настройке карты. Возможные значения: 0 - *(по умолчанию)* не раскрывать макросы; 1 - раскрывать макросы.

Свойство	Тип	Описание
expandproblem	integer	Следует ли отображать проблемный триггер для элементов с одной проблемой. Возможные значения: 0 - всегда отображать количество проблем; 1 - *(по умолчанию)* отображать проблемный триггер, если проблема только одна.
grid_align	integer	Следует ли включить выравнивание по сетке. Возможные значения: 0 - отключить выравнивание по сетке; 1 - *(по умолчанию)* включить выравнивание по сетке.
grid_show	integer	Следует ли показывать 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 иконок. Возможные значения: 0 - подсветка отключена; 1 - *(по умолчанию)* подсветка включена.
iconmapid	ID	ID карты иконок, используемой на карте.
label_format	integer	Следует ли включить расширенные метки. Возможные значения: 0 - *(по умолчанию)* отключить расширенные метки; 1 - включить расширенные метки.
label_location	integer	Расположение метки элемента карты. Возможные значения: 0 - *(по умолчанию)* снизу; 1 - слева; 2 - справа; 3 - сверху.
label_string_host	string	Пользовательская метка для элементов узла сети. Поведение свойства: - *обязательно*, если `label_type_host` установлено в ”custom”
label_string_hostgroup	string	Пользовательская метка для элементов группы узлов сети. Поведение свойства: - *обязательно*, если `label_type_hostgroup` установлено в ”custom”
label_string_image	string	Пользовательская метка для элементов изображения. Поведение свойства: - *обязательно*, если `label_type_image` установлено в ”custom”
label_string_map	string	Пользовательская метка для элементов карты. Поведение свойства: - *обязательно*, если `label_type_map` установлено в ”custom”
label_string_trigger	string	Пользовательская метка для элементов триггера. Поведение свойства: - *обязательно*, если `label_type_trigger` установлено в ”custom”

Свойство	Тип	Описание
label_type	integer	Тип метки элемента карты. Возможные значения: 0 - метка; 1 - IP-адрес; 2 - *(по умолчанию)* имя элемента; 3 - только состояние; 4 - ничего.
label_type_host	integer	Тип метки для элементов узла сети. Возможные значения: 0 - метка; 1 - IP-адрес; 2 - *(по умолчанию)* имя элемента; 3 - только состояние; 4 - ничего; 5 - пользовательская.
label_type_hostgroup	integer	Тип метки для элементов группы узлов сети. Возможные значения: 0 - метка; 2 - *(по умолчанию)* имя элемента; 3 - только состояние; 4 - ничего; 5 - пользовательская.
label_type_image	integer	Тип метки для элементов группы узлов сети. Возможные значения: 0 - метка; 2 - *(по умолчанию)* имя элемента; 4 - ничего; 5 - пользовательская.
label_type_map	integer	Тип метки для элементов карты. Возможные значения: 0 - метка; 2 - *(по умолчанию)* имя элемента; 3 - только состояние; 4 - ничего; 5 - пользовательская.
label_type_trigger	integer	Тип метки для элементов триггера. Возможные значения: 0 - метка; 2 - *(по умолчанию)* имя элемента; 3 - только состояние; 4 - ничего; 5 - пользовательская.
markelements	integer	Следует ли подсвечивать элементы карты, у которых недавно изменилось состояние. Возможные значения: 0 - *(по умолчанию)* не подсвечивать элементы; 1 - подсвечивать элементы.
severity_min	integer	Минимальная важность триггеров, которые будут отображаться на карте. Список поддерживаемых уровней важности триггеров см. в свойстве триггера `severity`.

Свойство	Тип	Описание
show_element_label	integer	Как отображать метки элементов по умолчанию. Возможные значения: 0 - всегда отображать; 1 - *(по умолчанию)* автоматически скрывать.
show_link_label	integer	Как отображать метки связей по умолчанию. 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)* приватная карта.
show_suppressed	integer	Следует ли показывать 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.

Свойство	Тип	Описание
selementid	ID	ID элемента карты. Поведение свойства: - *только для чтения*
elements	array	Объект данных элемента. Поведение свойства: - *обязательно*, если `elementtype` установлено в ”host”, ”map”, ”trigger” или ”host group”
elementtype	integer	Тип элемента карты. Возможные значения: 0 - узел сети; 1 - карта; 2 - триггер; 3 - группа узлов сети; 4 - изображение. Поведение свойства: - *обязательно*
iconid_off	ID	ID изображения, используемого для отображения элемента в состоянии по умолчанию. Поведение свойства: - *обязательно*

Свойство	Тип	Описание
areatype	integer	Как должны отображаться отдельные узлы сети группы узлов сети. Возможные значения: 0 - *(по умолчанию)* элемент группы узлов сети будет занимать всю карту; 1 - элемент группы узлов сети будет иметь 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	Условие фильтрации тегов элемента карты, метод вычисления. Возможные значения: 0 - *(по умолчанию)* И/Или; 2 - Или.
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` установлено в ”host”, ”map”, ”trigger” или ”host group”
iconid_maintenance	ID	ID изображения, используемого для отображения элементов карты в обслуживании. Поведение свойства: - *поддерживается*, если `elementtype` установлено в ”host”, ”map”, ”trigger” или ”host group”
iconid_on	ID	ID изображения, используемого для отображения элементов карты с проблемами. Поведение свойства: - *поддерживается*, если `elementtype` установлено 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.
show_label	integer	Как отображать метку элемента. Возможные значения: -1 - *(по умолчанию)* значение карты по умолчанию; 0 - отображать всегда; 1 - автоматически скрывать.

Свойство	Тип	Описание
sysmapid	ID	ID карты, к которой принадлежит элемент. Поведение свойства: - *только для чтения*
urls	array	URL-адреса элемента карты. Объект URL элемента карты подробно описан ниже.
use_iconmap	integer	Следует ли использовать сопоставление иконок для элементов узлов сети. Возможные значения: 0 - не использовать сопоставление иконок; 1 - *(по умолчанию)* использовать сопоставление иконок.
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.

Элемент карты «Узел сети»

Объект элемента карты Host определяет один элемент узла сети.

Свойство	Type	Описание
hostid	ID	ID узла сети.

Элемент карты «Группа узлов сети»

Объект элемента карты Host group определяет один элемент группы узлов сети.

Свойство	Type	Описание
groupid	ID	ID группы узлов сети.

Элемент карты Map

Объект элемента карты Map определяет один элемент карты.

Свойство	Тип	Описание
sysmapid	ID	ID карты.

Элемент карты Trigger

Объект элемента карты Trigger определяет один или несколько элементов триггера.

Свойство	Type	Описание
triggerid	ID	ID триггера.

Тег элемента карты

Объект тега элемента карты имеет следующие свойства.

Свойство	Тип	Описание
tag	string	Имя тега элемента карты. Поведение свойства: - *обязательно*
operator	integer	Условие оператора тега элемента карты. Возможные значения: 0 - *(по умолчанию)* 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 определенного элемента карты. Он имеет следующие свойства:

Свойство	Тип	Описание
sysmapelementurlid	ID	ID URL элемента карты. Поведение свойства: - *только для чтения*
name	string	Подпись ссылки. Поведение свойства: - *обязательно*
url	string	URL ссылки. Поведение свойства: - *обязательно*
selementid	ID	ID элемента карты, к которому относится URL.

Ссылка карты

Объект ссылки карты определяет связь между двумя элементами карты. Он имеет следующие свойства.

Свойство	Тип	Описание
linkid	ID	ID ссылки карты. Поведение свойства: - *только для чтения*
sysmapid	ID	ID карты, к которой принадлежит ссылка.
selementid1	ID	ID первого элемента карты, связанного с одной стороны. Поведение свойства: - *обязательно*

Свойство	Тип	Описание
selementid2	ID	ID первого элемента карты, связанного с другой стороны. Поведение свойства: - *обязательно*
drawtype	integer	Стиль отрисовки линии ссылки. 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 ссылки.
show_label	integer	Как отображать метку ссылки. Возможные значения: -1 - *(по умолчанию)* значение карты по умолчанию; 0 - отображать всегда; 1 - скрывать автоматически.
indicator_type	integer	Выберите тип индикатора ссылки. Возможные значения: 0 - *(по умолчанию)* статическая ссылка; 1 - триггер; 2 - значение элемента данных.
linktriggers	array	Триггеры ссылки карты, используемые как индикаторы состояния ссылки. Объект триггера ссылки карты подробно описан ниже. Поведение свойства: - *обязательно*, если `indicator_type` установлено в ”trigger” (1).
itemid	ID	ID элемента данных. Поведение свойства: - *обязательно*, если `indicator_type` установлено в ”item value” (2).
highlights	array	Подсветки ссылки карты, используемые как индикаторы состояния ссылки. Объект индикаторов ссылки карты подробно описан ниже. Поведение свойства: - *обязательно*, если `indicator_type` установлено в ”item value” (2).
thresholds	array	Пороговые значения ссылки карты, используемые как индикаторы состояния ссылки. Объект индикаторов ссылки карты подробно описан ниже. Поведение свойства: - *обязательно*, если `indicator_type` установлено в ”item value” (2).
permission	integer	Тип уровня прав доступа. Возможные значения: -1 - нет; 2 - только чтение; 3 - чтение-запись.

Триггер ссылки карты

Объект триггера ссылки карты определяет индикатор состояния ссылки карты на основе состояния триггера. Он имеет следующие свойства:

Свойство	Type	Описание
triggerid	ID	ID триггера, используемого в качестве индикатора ссылки. Поведение свойства: - *обязательно*
color	string	Цвет индикатора в виде шестнадцатеричного кода цвета. По умолчанию: `DD0000`.
drawtype	integer	Стиль отрисовки 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 определённого типа на карте. Он имеет следующие свойства:

Свойство	Type	Описание
sysmapurlid	ID	ID URL карты. Property behavior: - *read-only*
name	string	Подпись ссылки. Property behavior: - *required*

Свойство	Type	Описание
url	string	URL ссылки. Property behavior: - *required*
elementtype	integer	Тип элемента карты, для которого URL будет доступен. Список поддерживаемых типов см. в свойстве `type` элемента карты. По умолчанию: 0.
sysmapid	ID	ID карты, к которой относится URL.

Пользователь карты

Список разрешений карты на основе пользователей. Имеет следующие свойства:

Свойство	Type	Описание
sysmapuserid	ID	ID пользователя карты. Поведение свойства: - *только для чтения*
userid	ID	ID пользователя. Поведение свойства: - *обязательно*
permission	integer	Тип уровня разрешений. Возможные значения: 2 - только чтение; 3 - чтение-запись. Поведение свойства: - *обязательно*

Группа пользователей карты

Список разрешений карты на основе групп пользователей. Имеет следующие свойства:

Свойство	Type	Описание
sysmapusrgrpid	ID	ID группы пользователей карты. Property behavior: - *только для чтения*
usrgrpid	ID	ID группы пользователей. Property behavior: - *обязательно*
permission	integer	Тип уровня разрешений. Возможные значения: 2 - только чтение; 3 - чтение-запись. Property behavior: - *обязательно*

Фигуры карты

Объект фигуры карты определяет геометрическую фигуру (с текстом или без него), отображаемую на карте. Он имеет следующие свойства:

Свойство	Тип	Описание
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 является обязательным при создании новых фигур. Поведение свойства: - *обязательно*
x	integer	X-координаты фигуры в пикселях. По умолчанию: 0.
y	integer	Y-координаты 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.
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.

Свойство	Тип	Описание
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: ( ).
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:

Свойство	Тип	Описание
sysmap_shapeid	ID	ID элемента фигуры карты. Поведение свойства: - *только для чтения*
x1	integer	X-координаты точки 1 линии в пикселях. По умолчанию: 0.
y1	integer	Y-координаты точки 1 линии в пикселях. По умолчанию: 0.
x2	integer	X-координаты точки 2 линии в пикселях. По умолчанию: 200.
y2	integer	Y-координаты точки 2 линии в пикселях. По умолчанию: 200.

Свойство	Тип	Описание
line_type	integer	Тип линий. Возможные значения: 0 - нет; 1 - ————; 2 - ┅; 3 - – – –. По умолчанию: 0.
line_width	integer	Ширина линий 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(  /  maps)`

Этот метод позволяет создавать новые карты.

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)` Карты для создания.

В дополнение к стандартным свойствам карты, метод принимает следующие параметры.

Параметр	Type	Описание
links	array	Связи карты, которые будут созданы на карте.
selements	array	Элементы карты, которые будут созданы на карте.
urls	array	URL карты, которые будут созданы на карте.
users	array	Общие доступы пользователя карты, которые будут созданы на карте.
userGroups	array	Общие доступы группы пользователей карты, которые будут созданы на карте.
shapes	array	Фигуры карты, которые будут созданы на карте.
lines	array	Линии карты, которые будут созданы на карте.

Note:
Чтобы создать связи карты, необходимо задать элементу карты `selementid` произвольное значение, а затем использовать это значение для ссылки на этот элемент в свойствах связи `selementid1` или `selementid2`. Когда элемент будет создан, это значение будет заменено на корректный ID, сгенерированный Zabbix. См. пример.

Возвращаемые значения

`(object)` Возвращает объект, содержащий идентификаторы созданных карт в свойстве `sysmapids`. Порядок возвращаемых идентификаторов соответствует порядку переданных карт.

Примеры

Создание пустой карты

Создайте карту без элементов.

Запрос:

```
{
    "jsonrpc": "2.0",
    "method": "map.create",
    "params": {
        "name": "Map",
        "width": 600,
        "height": 600
    },
    "id": 1
}
```

Ответ:

```
{
    "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 ссылки карты для указания 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, содержащим имя 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.

Запрос:

```
{
    "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
}
```

Ответ:

```
{
    "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(    mapIds)`

Этот метод позволяет удалять карты.

> **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

(array) Идентификаторы карт для удаления.

Возвращаемые значения

(object) Возвращает объект, содержащий идентификаторы удалённых карт в свойстве sysmapids.

Примеры

Удаление нескольких карт

Удалите две карты.

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( )

Метод позволяет получать карты по заданным параметрам.

> **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) Параметры, определяющие желаемый вывод.

Метод поддерживает следующие параметры.

Параметр	Тип	Описание
sysmapids	ID/array	Возвращает только карты с указанными ID.
userids	ID/array	Возвращает только карты, принадлежащие указанным ID пользователей.
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 иконок, используемой на карте.

Параметр	Тип	Описание
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, которым предоставлен общий доступ к карте.
selectUserGroups	query	Возвращает свойство `userGroups` с группами пользователей, которым предоставлен общий доступ к карте.
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

Получение всех данных 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 иконок
- Элемент карты
- Связь 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(  /  maps)`

Этот метод позволяет обновлять существующие карты.

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)` Свойства карты для обновления.

Свойство `mapid` должно быть определено для каждой карты, все остальные свойства являются необязательными. Будут обновлены только переданные свойства, все остальные останутся без изменений.

В дополнение к стандартным свойствам карты, метод принимает следующие параметры.

Параметр	Тип	Описание
links	array	Связи карты для замены существующих связей.
selements	array	Элементы карты для замены существующих элементов.
urls	array	URL карты для замены существующих URL.
users	array	Общие доступы пользователя карты для замены существующих элементов.
userGroups	array	Общие доступы групп 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 связи карты между новыми элементами карты, необходимо установить `selementid` элемента в произвольное значение, а затем использовать это значение для ссылки на данный элемент в свойствах связей `selementid1` или `selementid2`. Когда элемент будет создан, это значение будет заменено на корректный идентификатор, сгенерированный Zabbix. См. пример для map.create.

Возвращаемые значения

`(object)` Возвращает объект, содержащий идентификаторы обновлённых карт в свойстве `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 для администраторов и суперадминистраторов.

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*.

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.

Свойство	Тип	Описание
valuemapid	ID	ID карты значений. Поведение свойства: - *только для чтения* - *обязательно* для операций обновления
hostid	ID	ID узла сети или шаблона, которому принадлежит карта значений. Поведение свойства: - *неизменяемое* - *обязательно* для операций создания
name	string	Имя карты значений. Поведение свойства: - *обязательно* для операций создания
mappings	array	Сопоставления значений для текущей карты значений. Объект сопоставления подробно описан ниже. Поведение свойства: - *обязательно* для операций создания
uuid	string	Универсальный уникальный идентификатор, используемый для связывания импортированных карт значений с уже существующими. Если не указан, генерируется автоматически. Поведение свойства: - *поддерживается*, если карта значений принадлежит шаблону

Сопоставления значений

Объект сопоставлений значений определяет сопоставления значений карты значений. Он имеет следующие свойства.

Свойство	Type	Описание
type	integer	Тип совпадения 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 (диапазоны включают границы; можно определить несколько диапазонов, разделённых 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` установлен в ”0”, ”1”, ”2”, ”3”, ”4”, тогда `value` не может быть пустым. Если `type` установлен в ”5”, тогда `value` должен быть пустым.
value	string	Исходное значение. Property behavior: - *обязательно* если `type` установлен в ”1”, ”2”, ”3”, ”4” - *поддерживается* если `type` установлен в ”5”
newvalue	string	Значение, в которое преобразуется исходное значение. Property behavior: - *обязательно*

[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( / valuemaps)`

Этот метод позволяет создавать новые карты значений.

Note:
Этот метод доступен только пользователю с типом доступа *Супер-администратор*. Разрешения на вызов метода можно отозвать в настройках ролей 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(    valuemapids)`

Этот метод позволяет удалить преобразование значений.

> **Note:**
> Этот метод доступен только пользователю с типом доступа *Супер-администратор*. Разрешения на вызов метода можно отозвать в настройках ролей пользователя. Дополнительную информацию см. в разделе Роли пользователей.

Параметры

( ) ID удаляемых преобразований значений.

Возвращаемые значения

( ) Возвращает объект, который содержит ID удаленных преобразований значений, указанных в свойстве `valuemapids`.

Примеры

Удаление нескольких карт значений

Удалить две карты 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(          )`

Метод позволяет получить преобразование значений в соответствии с заданными параметрами.

> **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)` Параметры, определяющие желаемый вывод.

Метод поддерживает следующие параметры.

Параметр	Тип	Описание
valuemapids	ID/array	Возвращать только карты соответствия значений с указанными ID.
selectMappings	query	Возвращать соответствия значений для текущей карты соответствия 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	

Параметр	Тип	Описание
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 вместе с её 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(  /  valuemaps)`

Этот метод позволяет обновлять существующие преобразования значений.

> **Note:**
> Этот метод доступен только пользователю с типом доступа *Супер-администратор*. Разрешения на вызов метода можно отозвать в настройках ролей пользователя. Дополнительную информацию см. в разделе Роли пользователей.

Параметры

(object/array) Свойства карты соответствий, которые необходимо обновить.

Свойство valuemapid должно быть определено для каждой карты соответствий, все остальные свойства являются необязательными. Будут обновлены только переданные свойства, все остальные останутся без изменений.

Возвращаемые значения

( ) Возвращает объект, который содержит ID обновленных преобразований значений, указанных в свойствеvaluemapids.

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 иконок

Следующие объекты напрямую связаны с API `iconmap`.

Карта иконок

Объект карты иконок имеет следующие свойства.

Свойство	Тип	Описание
iconmapid	ID	ID карты иконок. Поведение свойства: - *только для чтения* - *обязательно* для операций обновления
default_iconid	ID	ID иконки по умолчанию. Поведение свойства: - *обязательно* для операций создания
name	string	Имя карты иконок. Поведение свойства: - *обязательно* для операций создания

Сопоставление значков

Объект сопоставления значков определяет конкретный значок, который будет использоваться для узлов сети с определённым значением поля инвентарных данных. Он имеет следующие свойства.

Свойство	Тип	Описание
iconid	ID	ID значка, используемого сопоставлением значков. Поведение свойства: - *обязательно*
expression	string	Выражение для сопоставления со значением поля инвентарных данных. Поведение свойства: - *обязательно*

Свойство	Тип	Описание
inventory_link	integer	ID поля инвентарных данных узла сети. Список поддерживаемых полей инвентарных данных см. в объекте инвентарных данных узла сети. Поведение свойства: - *обязательно*
sortorder	integer	Позиция сопоставления значков на карте значков. Поведение свойства: - *только для чтения*

iconmap.create

Описание

`object iconmap.create(  /  iconMaps)`

Этот метод позволяет создавать новые соответствия иконок.

Note:
Этот метод доступен только пользователю с типом доступа *Супер-администратор*. Разрешения на вызов метода можно отозвать в настройках ролей пользователя. Дополнительную информацию см. в разделе Роли пользователей.

Параметры

`(object/array)` Карты значков для создания.

В дополнение к стандартным свойствам карты значков, метод принимает следующие параметры.

Параметр	Тип	Описание
mappings	array	Сопоставления значков, которые будут созданы для карты значков. Поведение параметра: - *обязательный*

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 иконок

Создайте карту иконок 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(    iconMapIds)`

Этот метод позволяет удалять соответствия иконок.

Note:
Этот метод доступен только пользователю с типом доступа *Супер-администратор*. Разрешения на вызов метода можно отозвать в настройках ролей пользователя. Дополнительную информацию см. в разделе Роли пользователей.

Параметры

( ) ID удаляемых соответствий иконок.

Возвращаемые значения

( ) Возвращает объект, который содержит ID удаленных соответствий иконок, указанных в свойстве `iconmapids`.

Примеры

Удаление нескольких карт иконок

Удаление двух карт иконок.

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(          )`

Метод позволяет получать карты иконок по заданным параметры.

> **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)` Параметры, определяющие желаемый вывод.

Метод поддерживает следующие параметры.

Параметр	Тип	Описание
iconmapids	ID/array	Возвращать только карты иконок с указанными ID.
sysmapids	ID/array	Возвращать только карты иконок, используемые в указанных картах сети.
selectMappings	query	Возвращать свойство `mappings` с используемыми сопоставлениями иконок.
sortfield	string/array	Сортировать результат по указанным свойствам. Возможные значения: `iconmapid`, `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 карты иконок

Получение всех данных о карте иконок "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(    /     iconMaps)`

Этот метод позволяет обновлять существующие карты значков.

Note:
Этот метод доступен только пользователю с типом доступа *Супер-администратор*. Разрешения на вызов метода можно отозвать в настройках ролей пользователя. Дополнительную информацию см. в разделе Роли пользователей.

Параметры

`(object/array)` Свойства карты иконок, которые необходимо обновить.

Свойство `iconmapid` должно быть определено для каждой карты иконок, все остальные свойства являются необязательными. Будут обновлены только переданные свойства, все остальные останутся без изменений.

В дополнение к стандартным свойствам карты иконок, метод принимает следующие параметры.

Параметр	Type	Описание
mappings	array	Сопоставления иконок для замены существующих сопоставлений иконок.

Возвращаемые значения

`(  )` Возвращает объект, который содержит ID обновленных соответствий иконок, указанных в свойстве `iconmapids`.

Примеры

Переименование карты иконок

Переименуйте карту иконок 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 оповещений
  - Сопоставления групп подготовки

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	ID каталога пользователей. Если каталог пользователей удалён, значение свойства `userdirectoryid` объекта User object устанавливается в ”0” для всех пользователей, связанных с удалённым каталогом пользователей. Поведение свойства: - *только для чтения* - *обязательно* для операций обновления
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` установлено в ”Enabled” и `saml_jit_status` объекта Authentication object установлено в ”Enabled for configured SAML IdPs”
user_username	string	Атрибут каталога пользователей LDAP/SAML (а также атрибут SCIM, если `scim_status` установлено в ”SCIM provisioning is enabled”), содержащий имя пользователя, которое используется как значение свойства `name` объекта User object при подготовке пользователя. Примеры: *cn*, *commonName*, *displayName*, *name*
user_lastname	string	Атрибут каталога пользователей LDAP/SAML (а также атрибут SCIM, если `scim_status` установлено в ”SCIM provisioning is enabled”), содержащий фамилию пользователя, которая используется как значение свойства `surname` объекта User object при подготовке пользователя. Примеры: *sn*, *surname*, *lastName*

Свойство	Тип	Описание
provision_status	integer	Статус подготовки каталога пользователей. Возможные значения: 0 - *(по умолчанию)* Отключено (подготовка пользователей, создаваемых этим каталогом пользователей, отключена); 1 - Включено (подготовка пользователей, создаваемых этим каталогом пользователей, включена; дополнительно должен быть включён статус подготовки LDAP или SAML (`ldap_jit_status` или `saml_jit_status` объекта Authentication object)).
provision_groups	array	Массив объектов сопоставлений групп подготовки для сопоставления шаблона группы пользователей LDAP/SAML с группой пользователей Zabbix и ролью пользователя. Поведение свойства: - *обязательно*, если `provision_status` установлено в "Enabled"
provision_media	array	Массив объектов сопоставлений типов медиа для сопоставления LDAP/SAML-атрибутов медиа пользователя (например, email) с пользовательскими медиа Zabbix для отправки уведомлений.
LDAP-специфичные свойства:		
name	string	Уникальное имя каталога пользователей. Поведение свойства: - *обязательно*, если `idp_type` установлено в "User directory of type LDAP"
host	string	Имя узла, IP или URI LDAP-сервера. URI должен содержать схему (`ldap://` или `ldaps://`), имя узла и порт (необязательно). Примеры: *host.example.com* *127.0.0.1* *ldap://ldap.example.com:389* Поведение свойства: - *обязательно*, если `idp_type` установлено в "User directory of type LDAP"
port	integer	Порт LDAP-сервера. Поведение свойства: - *обязательно*, если `idp_type` установлено в "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}*" обязателен) Поведение свойства: - *обязательно*, если `idp_type` установлено в "User directory of type LDAP"

Свойство	Тип	Описание
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` установлено в ”User directory of type LDAP”
bind_dn	string	Учётная запись LDAP-сервера для привязки и поиска на LDAP-сервере. Для прямой привязки пользователя и анонимной привязки `bind_dn` должно 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` установлено в ”User directory of type LDAP”
bind_password	string	LDAP-пароль учётной записи для привязки и поиска на LDAP-сервере. Для прямой привязки пользователя и анонимной привязки `bind_password` должно быть пустым. Поведение свойства: - *поддерживается*, если `idp_type` установлено в ”User directory of type LDAP”
description	string	Описание каталога пользователей. Поведение свойства: - *поддерживается*, если `idp_type` установлено в ”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` установлено в ”User directory of type LDAP”

Свойство	Тип	Описание
group_filter	string	Строка фильтра для получения групп каталога пользователей LDAP, в которые входит 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, IP или URI LDAP-сервера (заменяется значением свойства `host`); *%{user}* - имя пользователя Zabbix. По умолчанию: *(%{groupattr}=%{user})* Примеры: - *(member=uid=%{ref},ou=Users,dc=example,dc=com)* будет соответствовать "User1", если объект группы LDAP содержит атрибут "*member*" со значением "*uid=User1,ou=Users,dc=example,dc=com*", и вернёт группу, в которую входит "User1"; - *(%{groupattr}=cn=%{ref},ou=Users,ou=Zabbix,DC=example,DC=com)* будет соответствовать "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, в которую входит "User1". Поведение свойства: - *поддерживается*, если `idp_type` установлено в "User directory of type LDAP"
group_member	string	Атрибут каталога пользователей LDAP, содержащий информацию о членах 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` установлено в "User directory of type LDAP"
group_membership	string	Атрибут каталога пользователей LDAP, содержащий информацию о группах, в которые входит пользователь. Пример: *memberOf* Поведение свойства: - *поддерживается*, если `idp_type` установлено в "User directory of type LDAP"

Свойство	Тип	Описание
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` установлено в "User directory of type LDAP"
start_tls	integer	Параметр конфигурации LDAP-сервера, позволяющий защитить обмен данными с LDAP-сервером с помощью Transport Layer Security (TLS). Обратите внимание, что `start_tls` должно быть установлено в "Disabled" для узлов, использующих протокол `ldaps://`. Возможные значения: 0 - *(по умолчанию)* Отключено; 1 - Включено. Поведение свойства: - *поддерживается*, если `idp_type` установлено в "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` установлено в "User directory of type LDAP"
SAML-специфичные свойства:		
idp_entityid	string	URI, идентифицирующий поставщика удостоверений и используемый для взаимодействия с поставщиком удостоверений в сообщениях SAML. Пример: *https://idp.example.com/idp* Поведение свойства: - *обязательно*, если `idp_type` установлено в "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` установлено в "User directory of type SAML"

Свойство	Тип	Описание
username_attribute	string	Атрибут каталога пользователей SAML (а также атрибут SCIM, если `scim_status` установлено в "SCIM provisioning is enabled"), содержащий имя пользователя, которое сравнивается со значением свойства `username` объекта User object при аутентификации. 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` установлено в "User directory of type SAML"
sso_url	string	URL службы единого входа SAML поставщика удостоверений, на который Zabbix будет отправлять запросы аутентификации SAML. Пример: *http://idp.example.com/idp/sso/saml* Поведение свойства: - *обязательно*, если `idp_type` установлено в "User directory of type SAML"
slo_url	string	URL службы единого выхода SAML поставщика удостоверений, на который Zabbix будет отправлять запросы выхода SAML. Пример: *https://idp.example.com/idp/slo/saml* Поведение свойства: - *поддерживается*, если `idp_type` установлено в "User directory of type SAML"
encrypt_nameid	integer	Следует ли шифровать SAML name ID. Возможные значения: 0 - *(по умолчанию)* Не шифровать name ID; 1 - Шифровать name ID. Поведение свойства: - *поддерживается*, если `idp_type` установлено в "User directory of type SAML"
encrypt_assertions	integer	Следует ли шифровать утверждения SAML. Возможные значения: 0 - *(по умолчанию)* Не шифровать утверждения; 1 - Шифровать утверждения. Поведение свойства: - *поддерживается*, если `idp_type` установлено в "User directory of type SAML"
nameid_format	string	Формат Name ID 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` установлено в "User directory of type SAML"

Свойство	Тип	Описание
scim_status	integer	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` установлено в ”User directory of type SAML”
sign_assertions	integer	Следует ли подписывать утверждения SAML подписью SAML. Возможные значения: 0 - *(по умолчанию)* Не подписывать утверждения; 1 - Подписывать утверждения. Поведение свойства: - *поддерживается*, если `idp_type` установлено в ”User directory of type SAML”
sign_authn_requests	integer	Следует ли подписывать запросы SAML AuthN подписью SAML. Возможные значения: 0 - *(по умолчанию)* Не подписывать запросы AuthN; 1 - Подписывать запросы AuthN. Поведение свойства: - *поддерживается*, если `idp_type` установлено в ”User directory of type SAML”
sign_messages	integer	Следует ли подписывать сообщения SAML подписью SAML. Возможные значения: 0 - *(по умолчанию)* Не подписывать сообщения; 1 - Подписывать сообщения. Поведение свойства: - *поддерживается*, если `idp_type` установлено в ”User directory of type SAML”
sign_logout_requests	integer	Следует ли подписывать запросы выхода SAML подписью SAML. Возможные значения: 0 - *(по умолчанию)* Не подписывать запросы выхода; 1 - Подписывать запросы выхода. Поведение свойства: - *поддерживается*, если `idp_type` установлено в ”User directory of type SAML”
sign_logout_responses	integer	Следует ли подписывать ответы выхода SAML подписью SAML. Возможные значения: 0 - *(по умолчанию)* Не подписывать ответы выхода; 1 - Подписывать ответы выхода. Поведение свойства: - *поддерживается*, если `idp_type` установлено в ”User directory of type SAML”

Свойство	Тип	Описание
idp_certificate	string	Содержимое сертификата Service Provider (SP) для настройки службы единого входа SAML (SSO). Примеры: *-----BEGIN CERTIFICATE-----* * MIIBIjANBgkqhkiG9w0BAQEFAAOCAQ8AMIIBCgKCAQEA7... ...more encoded data... -----END CERTIFICATE-----* Поведение свойства: - *поддерживается*, если `idp_type` установлено в "User directory of type SAML" и `$SSO['CERT_STORAGE']` установлено в `database` в zabbix.conf.php - *только для записи*
sp_private_key	string	Содержимое закрытого ключа Service Provider (SP) для настройки службы единого входа SAML (SSO). Обеспечивает безопасную аутентификацию и обмен данными с Identity Provider (IdP). Примеры: *-----BEGIN CERTIFICATE----- MIIEvQIBADANBgkqhkiG9w0BA... ...more encoded data... -----END CERTIFICATE-----* Поведение свойства: - *поддерживается*, если `idp_type` установлено в "User directory of type SAML" и `$SSO['CERT_STORAGE']` установлено в `database` в zabbix.conf.php - *только для записи*
sp_certificate	string	Содержимое сертификата Service Provider (SP) для настройки службы единого входа SAML (SSO). Примеры: *-----BEGIN CERTIFICATE-----* * MIIBIjANBgkqhkiG9w0BAQEFAAOCAQ8AMIIBCgKCAQEA7... ...more encoded data... -----END CERTIFICATE-----* Поведение свойства: - *поддерживается*, если `idp_type` установлено в "User directory of type SAML" и `$SSO['CERT_STORAGE']` установлено в `database` в zabbix.conf.php - *только для записи*
idp_certificate_hash	string	Хэш md5 значения idp_certificate. Возвращает пустую строку для пустого idp_certificate. Поведение свойства: - *поддерживается*, если `idp_type` установлено в "User directory of type SAML" и `$SSO['CERT_STORAGE']` установлено в `database` в zabbix.conf.php - *только для чтения*
sp_private_key_hash	string	Хэш md5 значения sp_private_key. Возвращает пустую строку для пустого sp_private_key. Поведение свойства: - *поддерживается*, если `idp_type` установлено в "User directory of type SAML" и `$SSO['CERT_STORAGE']` установлено в `database` в zabbix.conf.php - *только для чтения*

Свойство	Тип	Описание
sp_certificate_hash	string	Хэш md5 значения sp_certificate. Возвращает пустую строку для пустого sp_certificate. Поведение свойства: - *поддерживается*, если `idp_type` установлено в "User directory of type SAML" и `$SSO['CERT_STORAGE']` установлено в `database` в zabbix.conf.php - *только для чтения*

Сопоставления типов медиа

Объект сопоставлений типов медиа имеет следующие свойства.

Свойство	Тип	Описание
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 должен быть создан; используется как значение свойства `mediatypeid` объекта Media. Поведение свойства: - *обязательно*
attribute	string	Атрибут каталога пользователей LDAP/SAML (а также атрибут SCIM, если `scim_status` установлен в значение "SCIM provisioning is enabled"), который содержит медиа пользователя (например, *user@example.com*), используемое как значение свойства `sendto` объекта Media. Если он присутствует в данных, полученных от поставщика удостоверений LDAP/SAML, и значение не пустое, это приведет к созданию 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.
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 подготовки

Сопоставления групп подготовки имеют следующие свойства.

Свойство	Тип	Описание
name	string	Полное имя группы (например, *Zabbix administrators*) в каталоге пользователей LDAP/SAML (а также SCIM, если `scim_status` установлено в значение "SCIM provisioning is enabled"). Поддерживает подстановочный символ "*". Уникально среди всех сопоставлений групп подготовки. Поведение свойства: - *обязательно*
roleid	ID	ID роли пользователя, назначаемой пользователю. Если совпадают несколько сопоставлений групп подготовки, пользователю назначается роль с наивысшим типом пользователя (*User*, *Admin* или *Super admin*). Если имеется несколько ролей с одинаковым типом пользователя, пользователю назначается первая роль (отсортированная в алфавитном порядке). Поведение свойства: - *обязательно*
user_groups	array	Массив объектов ID групп пользователей Zabbix. Каждый объект имеет следующие свойства: `usrgrpid` - `(ID)` ID группы пользователей Zabbix, назначаемой пользователю. Если совпадают несколько сопоставлений групп подготовки, пользователю назначаются группы пользователей Zabbix из всех совпавших сопоставлений. Поведение свойства: - *обязательно*

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)` Пользовательские каталоги для создания.

Метод принимает пользовательские каталоги со стандартными свойствами пользовательского каталога.

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 (с включённым JIT-подготовкой пользователей)

Создайте каталог пользователей для аутентификации пользователей через LDAP (с включённой JIT-подготовкой пользователей). 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 как минимум одной группой пользователей.
Каталог пользователей LDAP по умолчанию нельзя удалить, если `authentication.ldap_configured` установлен в `1` или если остаются другие каталоги пользователей.

> **Note:**
> Этот метод доступен только пользователям типа *Супер-администратор*.

Параметры

`(array)` Идентификаторы каталогов пользователей, которые необходимо удалить.

Возвращаемые значения

`(object)` Возвращает объект, содержащий идентификаторы удалённых каталогов пользователей в свойстве `userdirectoryids`.

Примеры

Удаление нескольких каталогов пользователей

Удалите два каталога 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)`

Этот метод позволяет получать каталоги пользователей в соответствии с заданными параметрами.

Note:
Этот метод доступен только для типов пользователей *Супер-администратор*.

Параметры

`(object)` Параметры, определяющие желаемый вывод.

Метод поддерживает следующие параметры.

Параметр	Тип	Описание
userdirectoryids	ID/array	Возвращать только каталоги пользователей с указанными ID.
selectUsrgrps	query	Возвращать свойство `usrgrps` с группами пользователей, связанными с каталогом пользователей. Поддерживает `count`.
selectProvisionMedia	query	Возвращать свойство `provision_media` с сопоставлениями типов медиа, связанными с каталогом пользователей.
selectProvisionGroups	query	Возвращать свойство `provision_groups` с сопоставлениями групп подготовки, связанными с каталогом пользователей.
sortfield	string/array	Сортировать результат по указанным свойствам. Возможные значения: `name`.
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. Поддерживаемые свойства: `userdirectoryid`, `idp_type`, `provision_status`.

Параметр	Тип	Описание
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 "%…%"`. Поддерживаемые свойства: `name`, `description`. Каталог пользователей типа SAML будет иметь пустое значение как для свойства `name`, так и для свойства `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 одно из следующего:

- массив объектов
- количество полученных объектов, если был использован параметр `countOutput`

Примеры

Получение каталогов пользователей

Получить все каталоги пользователей с дополнительными свойствами, отображающими сопоставления типов оповещений и сопоставления групп подготовки, связанных 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

Исходный код

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, группы пользователей, способы оповещения пользователя будут назначены пользователю). Для такого типа проверки API-запрос должен быть выполнен для каталога пользователей, у которого `provision_status` установлен в enabled.

Note:
Этот метод доступен только для пользователей типа *Супер-администратор*.

Параметры

`(object)` Свойства каталога пользователей.

Поскольку API `userdirectory.get` не возвращает поле `bind_password`, необходимо указать `userdirectoryid` и/или `bind_password`.
В дополнение к стандартным свойствам каталога пользователей, метод принимает следующие параметры.

Параметр	Тип	Описание
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 успешного выполнения.

Примеры

Проверка каталога пользователей для существующего пользователя

Проверьте каталог 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 пользователя "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

Проверка каталога пользователей "3" на соответствие настроенных данных параметрам подготовки пользователя "user3" (например, какая роль пользователя, какие группы пользователей, какие способы оповещения пользователя 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
}
```

Исходный код

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 типа пользователя *Супер-администратор*.

Параметры

`(object/array)` Свойства каталога пользователей, которые необходимо обновить.

Свойство `userdirectoryid` должно быть определено для каждого каталога пользователей, все остальные свойства являются необязательными. Будут обновлены только переданные свойства, все остальные останутся без изменений.

Возвращаемые значения

`(object)` Возвращает объект, содержащий идентификаторы обновлённых каталогов пользователей в свойстве `userdirectoryids`.

Примеры

Обновление bind password для каталога пользователей

Установите новый bind password 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 подготовки 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
}
```

Исходный код

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( )`

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 ролей пользователя. Дополнительную информацию см. в разделе Роли пользователей.

Параметры

`(object)` Параметры, определяющие объекты для экспорта и используемый формат.

Параметр	Тип	Описание
format	string	Формат, в котором должны быть экспортированы данные. Возможные значения: `yaml` — YAML; `xml` — XML; `json` — JSON; `raw` — необработанный PHP-массив. Поведение параметра: - *обязательный*
prettyprint	boolean	Сделать вывод более удобочитаемым для человека, добавив отступы. Возможные значения: `true` — добавить отступы; `false` — *(по умолчанию)* не добавлять отступы.
options	object	Объекты для экспорта. Объект `options` имеет следующие параметры: `host_groups` — `(array)` ID групп узлов сети для экспорта; `hosts` — `(array)` ID узлов сети для экспорта; `images` — `(array)` ID изображений для экспорта; `maps` — `(array)` ID карт для экспорта; `mediaTypes` — `(array)` ID типов оповещений для экспорта; `template_groups` — `(array)` ID групп шаблонов для экспорта; `templates` — `(array)` ID шаблонов для экспорта. Пользователи типов *Admin* и *User* могут экспортировать только те объекты, для которых у них есть разрешение *только для чтения* или *чтение-запись*, а также изображения, но не типы оповещений. Поведение параметра: - *обязательный*

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(          )`

Этот метод позволяет импортировать данные конфигурации из сериализованной строки.

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)` Параметры, содержащие данные для импорта и правила обработки этих данных.

Параметр	Тип	Описание
format	string	Формат сериализованной строки. Возможные значения: `yaml` - YAML; `xml` - XML; `json` - JSON. Поведение параметра: - *обязательный*
source	string	Сериализованная строка, содержащая данные конфигурации. Поведение параметра: - *обязательный*

Параметр	Тип	Описание
rules	object	Правила импорта новых и существующих объектов. Пользователи типа *Admin* могут импортировать только те объекты, на которые у них есть разрешение *чтение-запись*, 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 пользователя имеет разрешение на группу узлов сети, к которой будет принадлежать импортируемый узел сети. Изображения и типы медиа не могут быть импортированы пользователями типа *Admin*. Параметр `rules` подробно описан в таблице ниже. Поведение параметра: *- обязательный*

Attention:
Если правила не заданы, конфигурация не будет обновлена.

Объект `rules` поддерживает следующие параметры.

Параметр	Тип	Описание
discoveryRules	object	Правила импорта правил LLD. Поддерживаемые параметры: `createMissing` - `(boolean)` если установлено значение `true`, будут созданы новые правила LLD; по умолчанию: `false`; `updateExisting` - `(boolean)` если установлено значение `true`, существующие правила LLD будут обновлены; по умолчанию: `false`; `deleteMissing` - `(boolean)` если установлено значение `true`, правила LLD, отсутствующие в импортируемых данных, будут удалены из базы данных; по умолчанию: `false`.
graphs	object	Правила импорта графиков. Поддерживаемые параметры: `createMissing` - `(boolean)` если установлено значение `true`, будут созданы новые графики; по умолчанию: `false`; `updateExisting` - `(boolean)` если установлено значение `true`, существующие графики будут обновлены; по умолчанию: `false`; `deleteMissing` - `(boolean)` если установлено значение `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 значение `true`, будут созданы новые группы узлов сети; по умолчанию: `false`; `updateExisting` - `(boolean)` если установлено значение `true`, существующие группы узлов сети будут обновлены; по умолчанию: `false`.
template_groups	object	Правила импорта групп шаблонов. Поддерживаемые параметры: `createMissing` - `(boolean)` если установлено значение `true`, будут созданы новые группы шаблонов; по умолчанию: `false`; `updateExisting` - `(boolean)` если установлено значение `true`, существующие группы шаблонов будут обновлены; по умолчанию: `false`.

Параметр	Тип	Описание
hosts	object	Правила импорта узлов сети. Поддерживаемые параметры: `createMissing` - `(boolean)` если установлено значение `true`, будут созданы новые узлы сети; по умолчанию: `false`; `updateExisting` - `(boolean)` если установлено значение `true`, существующие узлы сети будут обновлены; по умолчанию: `false`.
httptests	object	Правила импорта веб-сценариев. Поддерживаемые параметры: `createMissing` - `(boolean)` если установлено значение `true`, будут созданы новые веб-сценарии; по умолчанию: `false`; `updateExisting` - `(boolean)` если установлено значение `true`, существующие веб-сценарии будут обновлены; по умолчанию: `false`; `deleteMissing` - `(boolean)` если установлено значение `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 значение `true`, будут созданы новые изображения; по умолчанию: `false`; `updateExisting` - `(boolean)` если установлено значение `true`, существующие изображения будут обновлены; по умолчанию: `false`.
items	object	Правила импорта элементов данных. Поддерживаемые параметры: `createMissing` - `(boolean)` если установлено значение `true`, будут созданы новые элементы данных; по умолчанию: `false`; `updateExisting` - `(boolean)` если установлено значение `true`, существующие элементы данных будут обновлены; по умолчанию: `false`; `deleteMissing` - `(boolean)` если установлено значение `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 значение `true`, будут созданы новые карты; по умолчанию: `false`; `updateExisting` - `(boolean)` если установлено значение `true`, существующие карты будут обновлены; по умолчанию: `false`.
mediaTypes	object	Правила импорта типов медиа. Поддерживаемые параметры: `createMissing` - `(boolean)` если установлено значение `true`, будут созданы новые типы медиа; по умолчанию: `false`; `updateExisting` - `(boolean)` если установлено значение `true`, существующие типы медиа будут обновлены; по умолчанию: `false`.

Параметр	Тип	Описание
templateLinkage	object	Правила импорта связей шаблонов. Поддерживаемые параметры: `createMissing` - (`boolean`) если установлено значение `true`, шаблоны, не связанные с импортируемым узлом сети или шаблоном, но присутствующие в импортируемых данных, будут связаны; по умолчанию: `false`; `deleteMissing` - (`boolean`) если установлено значение `true`, шаблоны, связанные с импортируемым узлом сети или шаблоном, но отсутствующие 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 значение `true`, будут созданы новые шаблоны; по умолчанию: `false`; `updateExisting` - (`boolean`) если установлено значение `true`, существующие шаблоны будут обновлены; по умолчанию: `false`.
templateDashboards	object	Правила импорта панелей шаблонов. Поддерживаемые параметры: `createMissing` - (`boolean`) если установлено значение `true`, будут созданы новые панели шаблонов; по умолчанию: `false`; `updateExisting` - (`boolean`) если установлено значение `true`, существующие панели шаблонов будут обновлены; по умолчанию: `false`; `deleteMissing` - (`boolean`) если установлено значение `true`, панели шаблонов, отсутствующие в импортируемых данных, будут удалены из базы данных; по умолчанию: `false`.
triggers	object	Правила импорта триггеров. Поддерживаемые параметры: `createMissing` - (`boolean`) если установлено значение `true`, будут созданы новые триггеры; по умолчанию: `false`; `updateExisting` - (`boolean`) если установлено значение `true`, существующие триггеры будут обновлены; по умолчанию: `false`; `deleteMissing` - (`boolean`) если установлено значение `true`, триггеры, отсутствующие в импортируемых данных, будут удалены из базы данных; по умолчанию: `false`.
valueMaps	object	Правила импорта карт соответствия значений узла сети или шаблона. Поддерживаемые параметры: `createMissing` - (`boolean`) если установлено значение `true`, будут созданы новые карты соответствия значений; по умолчанию: `false`; `updateExisting` - (`boolean`) если установлено значение `true`, существующие карты соответствия значений будут обновлены; по умолчанию: `false`; `deleteMissing` - (`boolean`) если установлено значение `true`, карты соответствия 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 всё 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(  )`

Этот метод позволяет сравнить файл импорта с текущими элементами системы и показывает, что изменится, если этот файл импорта будет импортирован.

> **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)` Параметры, содержащие возможные данные для импорта и правила обработки этих данных.

Параметр	Тип	Описание
format	string	Формат сериализованной строки. Возможные значения: `yaml` - YAML; `xml` - XML; `json` - JSON. Поведение параметра: - *обязательный*
source	string	Сериализованная строка, содержащая данные конфигурации. Поведение параметра: - *обязательный*
rules	object	Правила того, как следует сравнивать новые и существующие объекты. Пользователи типа *Admin* и *User* могут сравнивать новые объекты только с существующими объектами, для которых у них есть права *только чтение* или *чтение-запись*. Параметр `rules` подробно описан в таблице ниже. Поведение параметра: - *обязательный*

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.

Параметр	Тип	Описание
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`.
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`.

Параметр	Тип	Описание
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`.
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`.
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`. Этот параметр не влияет на вывод. Он разрешен только для согласованности с `configuration.import`.
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`. Этот параметр не влияет на вывод. Он разрешен только для согласованности с `configuration.import`.
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`.

Параметр	Тип	Описание
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`. Этот параметр не влияет на вывод. Он разрешен только для согласованности с `configuration.import`.
mediaTypes	object	Правила импорта типов медиа. Поддерживаемые параметры: `createMissing` - (`boolean`) если установлено в `true`, будут созданы новые типы медиа; по умолчанию: `false`; `updateExisting` - (`boolean`) если установлено в `true`, существующие типы медиа будут обновлены; по умолчанию: `false`. Этот параметр не влияет на вывод. Он разрешен только для согласованности с `configuration.import`.
templateLinkage	object	Правила импорта связей шаблонов. Поддерживаемые параметры: `createMissing` - (`boolean`) если установлено в `true`, шаблоны, не связанные с импортируемым узлом сети или шаблоном, но присутствующие в импортируемых данных, будут связаны; по умолчанию: `false`; `deleteMissing` - (`boolean`) если установлено в `true`, шаблоны, связанные с импортируемым узлом сети или шаблоном, но отсутствующие 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`.
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`.
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`.

Параметр	Тип	Описание
valueMaps	object	Правила импорта карт соответствия значений узла сети или шаблона. Поддерживаемые параметры: `createMissing` - (`boolean`) если установлено в `true`, будут созданы новые карты соответствия значений; по умолчанию: `false`; `updateExisting` - (`boolean`) если установлено в `true`, существующие карты соответствия значений будут обновлены; по умолчанию: `false`; `deleteMissing` - (`boolean`) если установлено в `true`, карты соответствия 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:

- Корреляция
- Операция корреляции
- Фильтр корреляции
  - Условие фильтра корреляции

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.

Свойство	Тип	Описание
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 или отключена корреляция. Возможные значения: 0 - *(по умолчанию)* включена; 1 - отключена.

Операция корреляции

Объект операции корреляции определяет операцию, которая будет выполнена при выполнении корреляции. Он имеет следующие свойства.

Свойство	Type	Описание
type	integer	Тип операции. Возможные значения: 0 — закрыть старые события; 1 — закрыть новое событие. Поведение свойства: - *обязательно*

Фильтр корреляции

Объект фильтра корреляции определяет набор условий, которые должны быть выполнены для выполнения настроенных операций корреляции. Он имеет следующие свойства.

Свойство	Type	Описание
conditions	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 - And/Or; 1 - And; 2 - Or; 3 - Пользовательское выражение. Поведение свойства: - *обязательно*
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 корреляции

Объект условия фильтра корреляции определяет конкретное условие, которое должно быть проверено перед выполнением операций корреляции.

Свойство	Type	Описание
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: - *обязательно*

Свойство	Type	Описание
tag	string	Тег события (старый или новый). Поведение свойства: - *обязательно*, если type установлен в ”старый тег события”, ”новый тег события”, ”значение старого тега события” или ”значение нового тега события”
groupid	ID	ID группы узлов сети. Поведение свойства: - *обязательно*, если type установлен в ”новая группа узлов сети события”
oldtag	string	Старый тег события. Поведение свойства: - *обязательно*, если type установлен в ”пара тегов события”
newtag	string	Старый тег события. Поведение свойства: - *обязательно*, если type установлен в ”пара тегов события”
value	string	Значение тега события (старого или нового). Поведение свойства: - *обязательно*, если type установлен в ”значение старого тега события” или ”значение нового тега события”
formulaid	string	Произвольный уникальный ID, который используется для ссылки на условие из пользовательского выражения. Может содержать только заглавные буквы. ID должен быть определен пользователем при изменении условий фильтра, но будет сгенерирован заново при последующем их запросе.
operator	integer	Оператор условия. Поведение свойства: - *обязательно*, если type установлен в ”новая группа узлов сети события”, ”значение старого тега события” или ”значение нового тега события”

Note:
Чтобы лучше понять, как использовать фильтры с различными типами выражений, смотрите примеры на страницах методов correlation.get и correlation.create.

Для каждого типа условия поддерживаются следующие операторы и значения.

Условие	Название условия	Поддерживаемые операторы	Ожидаемое значение
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( / correlations)`

Этот метод позволяет создавать новые корреляции.

Note:
Этот метод доступен только пользователю с типом доступа *Супер-администратор*. Разрешения на вызов метода можно отозвать в настройках ролей пользователя. Дополнительную информацию см. в разделе Роли пользователей.

Параметры

`(object/array)` Корреляции для создания.

В дополнение к стандартным свойствам корреляции, метод принимает следующие параметры.

Параметр	Type	Описание
operations	array	Операции корреляции для создания корреляции. Поведение параметра: - *обязательный*
filter	object	Объект фильтра корреляции для корреляции. Поведение параметра: - *обязательный*

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 вычисления `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” и ”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( correlationids)`

Этот метод позволяет удалять корреляции.

> **Note:**
> Этот метод доступен только пользователю с типом доступа *Супер-администратор*. Разрешения на вызов метода можно отозвать в настройках ролей 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

Удалите две корреляции.

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( )`

Этот метод позволяет получать корреляции в соответствии с заданными параметрами.

> **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)` Параметры, определяющие желаемый вывод.

Метод поддерживает следующие параметры.

Параметр	Тип	Описание
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 вычисления ”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( / correlations)`

Этот метод позволяет обновить существующие корреляции.

Note:
Этот метод доступен только пользователю с типом доступа *Супер-администратор*. Разрешения на вызов метода можно отозвать в настройках ролей пользователя. Дополнительную информацию см. в разделе Роли пользователей.

Параметры

`(object/array)` Свойства корреляции, которые необходимо обновить.

Свойство `correlationid` должно быть определено для каждой корреляции, все остальные свойства являются необязательными. Будут обновлены только переданные свойства, все остальные останутся без изменений.

В дополнение к стандартным свойствам корреляции, метод принимает следующие параметры.

Параметр	Type	Описание
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 вычисления

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*.

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 модуля имеет следующие свойства.

Свойство	Тип	Описание
moduleid	ID	ID модуля, как он хранится в базе данных. Поведение свойства: - *только для чтения* - *обязательно* для операций обновления
id	string	Уникальный ID модуля, определённый разработчиком в файле manifest.json модуля. Возможные значения для встроенных модулей: см. описание свойства "type" в Виджет панели. Поведение свойства: - *обязательно* 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: - *обязательно* для операций создания
status	integer	Указывает, включён или отключён модуль. Возможные значения: 0 — *(по умолчанию)* Отключён; 1 — Включён.
config	object	Конфигурация модуля.

module.create

Описание

`object module.create( / modules)`

Этот метод позволяет устанавливать новые интерфейсные модули.

Note:
Этот метод доступен только пользователю с типом доступа *Супер-администратор*. Разрешения на вызов метода можно отозвать в настройках ролей пользователя. Дополнительную информацию см. в разделе Роли пользователей.

Attention:
Файлы модулей необходимо распаковать вручную в правильные подкаталоги, соответствующие 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 "Включен".

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

Исходный код

CModule::create() в *ui/include/classes/api/services/CModule.php*.

module.delete

Описание

`object module.delete( moduleids)`

Этот метод позволяет удалять модули.

Note:
Этот метод доступен только пользователю с типом доступа *Супер-администратор*. Разрешения на вызов метода можно отозвать в настройках ролей пользователя. Дополнительную информацию см. в разделе Роли пользователей.

Attention:
Файлы модуля необходимо удалять вручную.

Параметры

(array) Идентификаторы модулей для удаления.

Возвращаемые значения

(object) Возвращает объект, содержащий идентификаторы деинсталлированных 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
}
```

Исходный код

CModule::delete() в *ui/include/classes/api/services/CModule.php*.

module.get

Описание

`integer/array module.get(object parameters)`

Этот метод позволяет получать модули в соответствии с заданными параметрами.

Note:
Этот метод доступен только для пользователей типа *Супер-администратор*. Разрешения на вызов этого метода могут быть отозваны в настройках роли пользователя. Для получения дополнительной информации смотрите Роли пользователей.

Параметры

`(object)` Параметры, определяющие желаемый вывод.

Метод поддерживает следующие параметры.

Параметр	Тип	Описание
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	

Параметр	Тип	Описание
startSearch	boolean	

Возвращаемые значения

`(integer/array)` Возвращает одно из следующего:

- 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

Исходный код

CModule::get() в *ui/include/classes/api/services/CModule.php*.

module.update

Описание

`object module.update( / modules)`

Этот метод позволяет обновлять существующие модули.

> **Note:**
> Этот метод доступен только пользователю с типом доступа *Супер-администратор*. Разрешения на вызов метода можно отозвать в настройках ролей пользователя. Дополнительную информацию см. в разделе Роли пользователей.

Параметры

`(object/array)` Свойства модуля, которые необходимо обновить.

Свойство `moduleid` должно быть определено для каждого модуля, все остальные свойства являются необязательными. Будут обновлены только указанные свойства.

Метод принимает модули со стандартными свойствами модуля.

Возвращаемые значения

`(object)` Возвращает объект, содержащий идентификаторы обновлённых 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

Исходный код

CModule::update() в *ui/include/classes/api/services/CModule.php*.

Настройки

Этот класс предназначен для работы с общими настройками администрирования.

Ссылки на объекты:

- Settings

Доступные методы:

- settings.get — получить настройки
- settings.update — обновить настройки

Объект настроек

Следующие объекты напрямую связаны с API `settings`.

Настройки

Объект settings имеет следующие свойства.

Свойство	Тип	Описание
default_lang	string	Язык системы по умолчанию. По умолчанию: `en_US`.
default_timezone	string	Часовой пояс системы по умолчанию. По умолчанию: `system` — системный по умолчанию. Полный список поддерживаемых часовых поясов см. в документации PHP.
default_theme	string	Тема по умолчанию. Возможные значения: `blue-theme` - *(по умолчанию)* Blue; `dark-theme` - Dark; `hc-light` - High-contrast light; `hc-dark` - High-contrast dark.
search_limit	integer	Ограничение для результатов поиска и фильтрации. По умолчанию: 1000.
max_overview_table_size	integer	Максимальное количество столбцов и строк в виджетах панели мониторинга Обзор данных и Обзор триггеров. По умолчанию: 50.
max_in_table	integer	Максимальное количество элементов для отображения 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 без прав Super admin и пользователям, не входящим в группы пользователей с включённым режимом отладки. Возможные значения: 0 - *(по умолчанию)* Не показывать технические ошибки; 1 - Показывать технические ошибки.

Свойство	Тип	Описание
history_period	string	Максимальный период отображения исторических данных в виджетах панели мониторинга Последние данные, Веб и Обзор данных. Принимает секунды и единицу времени с суффиксом. По умолчанию: 24h.
period_default	string	Период по умолчанию для фильтра времени. Принимает секунды и единицу времени с суффиксом, с поддержкой месяцев и лет (30s, 1m, 2h, 1d, 1M, 1y). По умолчанию: 1h.
max_period	string	Максимальный период для фильтра времени. Принимает секунды и единицу времени с суффиксом, с поддержкой месяцев и лет (30s, 1m, 2h, 1d, 1M, 1y). По умолчанию: 2y.
severity_color_0	string	Цвет для важности ”Не классифицировано” в виде шестнадцатеричного кода цвета. По умолчанию: 97AAB3.
severity_color_1	string	Цвет для важности ”Информация” в виде шестнадцатеричного кода цвета. По умолчанию: 7499FF.
severity_color_2	string	Цвет для важности ”Предупреждение” в виде шестнадцатеричного кода цвета. По умолчанию: FFC859.
severity_color_3	string	Цвет для важности ”Средняя” в виде шестнадцатеричного кода цвета. По умолчанию: FFA059.
severity_color_4	string	Цвет для важности ”Высокая” в виде шестнадцатеричного кода цвета. По умолчанию: E97659.
severity_color_5	string	Цвет для важности ”Катастрофическая” в виде шестнадцатеричного кода цвета. По умолчанию: E45959.
severity_name_0	string	Название для важности ”Не классифицировано”. По умолчанию: Not classified.
severity_name_1	string	Название для важности ”Информация”. По умолчанию: Information.
severity_name_2	string	Название для важности ”Предупреждение”. По умолчанию: Warning.
severity_name_3	string	Название для важности ”Средняя”. По умолчанию: Average.
severity_name_4	string	Название для важности ”Высокая”. По умолчанию: High.
severity_name_5	string	Название для важности ”Катастрофическая”. По умолчанию: Disaster.

Свойство	Тип	Описание
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 OK триггеров. Принимает секунды и единицу времени с суффиксом. По умолчанию: 5m.
blink_period	string	Период мигания при изменении состояния триггера. Принимает секунды и единицу времени с суффиксом. По умолчанию: 2m.
problem_unack_color	string	Цвет для неподтверждённых событий PROBLEM в виде шестнадцатеричного кода цвета. По умолчанию: CC0000.
problem_ack_color	string	Цвет для подтверждённых событий PROBLEM в виде шестнадцатеричного кода цвета. По умолчанию: CC0000.
ok_unack_color	string	Цвет для неподтверждённых событий RESOLVED в виде шестнадцатеричного кода цвета. По умолчанию: 009900.
ok_ack_color	string	Цвет для подтверждённых событий RESOLVED в виде шестнадцатеричного кода цвета. По умолчанию: 009900.
problem_unack_style	integer	Мигание для неподтверждённых событий PROBLEM. Возможные значения: 0 - Не показывать мигание; 1 - *(по умолчанию)* Показывать мигание.
problem_ack_style	integer	Мигание для подтверждённых событий PROBLEM. Возможные значения: 0 - Не показывать мигание; 1 - *(по умолчанию)* Показывать мигание.
ok_unack_style	integer	Мигание для неподтверждённых событий RESOLVED. Возможные значения: 0 - Не показывать мигание; 1 - *(по умолчанию)* Показывать мигание.
ok_ack_style	integer	Мигание для подтверждённых 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)* Отключено; 0 - Вручную; 1 - Автоматически.

Свойство	Тип	Описание
alert_usrgrpid	ID	ID группы пользователей, в которую будет отправляться сообщение тревоги о недоступности базы данных. Если установлено значение ”0”, сообщение тревоги отправляться не будет.
snmptrap_logging	integer	Журналировать несопоставленные SNMP trap. Возможные значения: 0 - Не журналировать несопоставленные SNMP trap; 1 - *(по умолчанию)* Журналировать несопоставленные SNMP trap.
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 секунды и единицу времени с суффиксом. 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 sandboxing для iframe. Возможные значения: 0 - Не использовать; 1 - *(по умолчанию)* Использовать.
iframe_sandboxing_exceptions	string	Исключения sandboxing для iframe.
connect_timeout	string	Тайм-аут соединения с сервером Zabbix. Диапазон возможных значений: 1-30s. По умолчанию: 3s. Поведение свойства: - *обязательно*
socket_timeout	string	Сетевой тайм-аут по умолчанию. Диапазон возможных значений: 1-300s. По умолчанию: 3s. Поведение свойства: - *обязательно*
media_type_test_timeout	string	Сетевой тайм-аут для теста типа медиа. Диапазон возможных значений: 1-300s. По умолчанию: 65s. Поведение свойства: - *обязательно*

Свойство	Тип	Описание
item_test_timeout	string	Сетевой тайм-аут для тестов элемента данных. Диапазон возможных значений: 1-600s. По умолчанию: 60s. Поведение свойства: - *обязательно*
script_timeout	string	Сетевой тайм-аут для выполнения скрипта. Диапазон возможных значений: 1-300s. По умолчанию: 60s. Поведение свойства: - *обязательно*
report_test_timeout	string	Сетевой тайм-аут для теста отчёта по расписанию. Диапазон возможных значений: 1-300s. По умолчанию: 60s. Поведение свойства: - *обязательно*
auditlog_enabled	integer	Включать ли журнал аудита. Возможные значения: 0 - Отключить; 1 - *(по умолчанию)* Включить.
auditlog_mode	integer	Включать ли журнал аудита для действий обнаружения низкого уровня, сетевого обнаружения и авторегистрации, выполняемых сервером (System user). Возможные значения: 0 - Отключить; 1 - *(по умолчанию)* Включить.
ha_failover_delay	string	Задержка переключения при отказе в секундах. По умолчанию: 1m. Поведение свойства: - *только для чтения*
geomaps_tile_provider	string	Поставщик тайлов геокарты. 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 геокарты. Поведение свойства: - *поддерживается*, если `geomaps_tile_provider` задан как пустая строка

Свойство	Тип	Описание
geomaps_max_zoom	integer	Максимальный уровень масштабирования геокарты. Возможные значения: 0-30. Поведение свойства: - *поддерживается*, если `geomaps_tile_provider` задан как пустая строка
geomaps_attribution	string	Текст атрибуции геокарты. Поведение свойства: - *поддерживается*, если `geomaps_tile_provider` задан как пустая строка
vault_provider	integer	Поставщик Vault. Возможные значения: 0 - *(по умолчанию)* HashiCorp Vault; 1 - CyberArk Vault.
proxy_secrets_provider	integer	Разрешать значения секретных макросов следующим образом: Возможные значения: 0 - *(по умолчанию)* только сервером; 1 - сервером и прокси независимо.
timeout_zabbix_agent	string	Тратить на обработку не более `timeout_*` секунд. Принимает секунды или единицу времени с суффиксом (например, 30s, 1m). Также принимает пользовательские макросы. Диапазон возможных значений: 1-600s. По умолчанию: 3s. По умолчанию для `timeout_browser`: 60s. Поведение свойства: - *обязательно*
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		

settings.get

Описание

`object settings.get(          )`

Метод позволяет получить объект настроек по заданным параметры.

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)` Параметры, определяющие желаемый вывод.

Метод поддерживает только один параметр.

Параметр	Тип	Описание
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": {
        "default_theme": "blue-theme",
        "search_limit": "1000",
        "max_in_table": "50",
        "server_check_interval": "10",
        "work_period": "1-5,09:00-18:00",
        "show_technical_errors": "0",
        "history_period": "24h",
        "period_default": "1h",
        "max_period": "2y",
        "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",
        "custom_color": "0",
        "ok_period": "5m",
        "blink_period": "2m",
        "problem_unack_color": "CC0000",
        "problem_ack_color": "CC0000",
        "ok_unack_color": "009900",
        "ok_ack_color": "009900",
        "problem_unack_style": "1",
        "problem_ack_style": "1",
        "ok_unack_style": "1",
        "ok_ack_style": "1",
        "discovery_groupid": "5",
        "default_inventory_mode": "-1",
        "alert_usrgrpid": "7",
        "snmptrap_logging": "1",
        "default_lang": "en_US",
        "default_timezone": "system",
        "login_attempts": "5",
```

```
        "login_block": "30s",
        "validate_uri_schemes": "1",
        "uri_valid_schemes": "http,https,ftp,file,mailto,tel,ssh",
        "x_frame_options": "SAMEORIGIN",
        "iframe_sandboxing_enabled": "1",
        "iframe_sandboxing_exceptions": "",
        "max_overview_table_size": "50",
        "connect_timeout": "3s",
        "socket_timeout": "3s",
        "media_type_test_timeout": "65s",
        "script_timeout": "60s",
        "item_test_timeout": "60s",
        "url": "",
        "report_test_timeout": "60s",
        "auditlog_enabled": "1",
        "auditlog_mode": "1",
        "ha_failover_delay": "1m",
        "geomaps_tile_provider": "OpenStreetMap.Mapnik",
        "geomaps_tile_url": "",
        "geomaps_max_zoom": "0",
        "geomaps_attribution": "",
        "vault_provider": "0",
        "timeout_zabbix_agent": "3s",
        "timeout_simple_check": "3s",
        "timeout_snmp_agent": "3s",
        "timeout_external_check": "3s",
        "timeout_db_monitor": "3s",
        "timeout_http_agent": "3s",
        "timeout_ssh_agent": "3s",
        "timeout_telnet_agent": "3s",
        "timeout_script": "3s"
    },
    "id": 1
}
```


Источник

CSettings::get() в *ui/include/classes/api/services/CSettings.php*.

settings.update

Описание

`object settings.update(          )`

Этот метод позволяет обновить существующие общие настройки.

> **Note:**
> Этот метод доступен только пользователю с типом доступа *Супер-администратор*. Разрешения на вызов метода можно отозвать в настройках ролей пользователя. Дополнительную информацию см. в разделе Роли пользователей.

Параметры

`(object)` Свойства настроек, которые необходимо обновить.

Возвращаемые значения

`(array)` Возвращает массив с именами обновлённых 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

Объект обнаруженной 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.

Свойство	Тип	Описание
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 стала недоступной.
lastup	timestamp	Время, когда обнаруженная служба в последний раз стала доступной.
port	integer	Номер порта службы.
status	integer	Состояние службы. Возможные значения: 0 - служба доступна; 1 - служба недоступна.

Свойство	Тип	Описание
value	string	Значение, возвращаемое службой при выполнении проверки обнаружения Zabbix агент, SNMPv1, SNMPv2 или SNMPv3.

dservice.get

Описание

`integer/array dservice.get( )`

Этот метод позволяет получать обнаруженные сервисы в соответствии с заданными параметрами.

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)` Параметры, определяющие желаемый вывод.

Метод поддерживает следующие параметры.

Параметр	Тип	Описание
dserviceids	ID/array	Возвращать только обнаруженные сервисы с указанными ID.
dhostids	ID/array	Возвращать только обнаруженные сервисы, принадлежащие указанным обнаруженным узлам сети.
dcheckids	ID/array	Возвращать только обнаруженные сервисы, которые были выявлены указанными проверками обнаружения.
druleids	ID/array	Возвращать только обнаруженные сервисы, которые были выявлены указанными правилами обнаружения.
selectDRules	query	Возвращать свойство `drules` с массивом правил обнаружения, выявивших сервис.
selectDHosts	query	Возвращать свойство `dhosts` с массивом обнаруженных узлов сети, которым принадлежит сервис.
selectHosts	query	Возвращать свойство `hosts` с узлами сети, имеющими тот же IP-адрес и тот же прокси, что и сервис. Поддерживает `count`.
limitSelects	integer	Ограничивает количество записей, возвращаемых подзапросами. Применяется к следующим подзапросам: `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

Получение сервисов, обнаруженных на узле сети

Получить все обнаруженные сервисы, 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.

Свойство	Тип	Описание
dhostid	ID	ID обнаруженного узла сети.
druleid	ID	ID правила обнаружения, которое обнаружило узел сети.
lastdown	timestamp	Время, когда обнаруженный узел сети в последний раз стал недоступен.
lastup	timestamp	Время, когда обнаруженный узел сети в последний раз стал доступен.
status	integer	Указывает, доступен обнаруженный узел сети или недоступен. Узел сети считается доступным, если у него есть хотя бы один активный обнаруженный сервис. Возможные значения: 0 - узел сети доступен; 1 - узел сети недоступен.

dhost.get

Описание

`integer/array dhost.get(` `)`

Метод позволяет получить обнаруженные узлы сети по заданным параметрам.

> **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)` Параметры, определяющие желаемый вывод.

Метод поддерживает следующие параметры.

Параметр	Тип	Описание
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 сервисы.
selectDRules	query	Возвращать свойство `drules` с массивом правил обнаружения, которые обнаружили узел сети.
selectDServices	query	Возвращать свойство `dservices` с обнаруженными сервисами, работающими на узле сети. Поддерживает `count`.

Параметр	Тип	Описание
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	
startSearch	boolean	

Возвращаемые значения

`(integer/array)` Возвращает:

- массив объектов;
- количество извлеченных объектов, если использовался параметр `countOutput`.

Примеры

Получение обнаруженных узлов сети по правилу обнаружения

Получить все узлы сети и обнаруженные службы, которые на них запущены, обнаруженные 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

- Обнаруженный сервис
- Правило обнаружения

Источник

CDHost::get() in *ui/include/classes/api/services/CDHost.php*.

Обслуживание

Этот класс предназначен для работы с обслуживаниями.

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.

Свойство	Тип	Описание
maintenanceid	ID	ID обслуживания. Поведение свойства: - *только для чтения* - *обязательно* для операций обновления
name	string	Имя обслуживания. Поведение свойства: - *обязательно* для операций создания

Свойство	Тип	Описание
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 (не включительно). 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)* со сбором данных; 1 - без сбора данных.
tags_evaltype	integer	Метод вычисления тегов проблем. Возможные значения: 0 - *(по умолчанию)* И/Или; 2 - Или.

Период времени

Объект периода времени используется для определения периодов, когда обслуживание должно вступать в силу. Он имеет следующие свойства.

Свойство	Type	Описание
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. Поведение свойства: - *поддерживается*, если `timeperiod_type` установлен в ”только один раз”

Свойство	Type	Описание
start_time	integer	Время суток, когда начинается обслуживание, в секундах. Указанное значение будет округлено вниз до минут. По умолчанию: 0. Поведение свойства: - *поддерживается*, если `timeperiod_type` установлен в ”ежедневно”, ”еженедельно” или ”ежемесячно”
every	integer	Для ежедневных и еженедельных периодов `every` определяет интервалы в днях или неделях, с которыми обслуживание должно вступать в силу. Значение по умолчанию, если `timeperiod_type` установлен в ”ежедневно” или ”еженедельно”: 1. Для ежемесячных периодов, когда задан `day`, свойство `every` определяет день месяца, когда обслуживание должно вступать в силу. Значение по умолчанию, если `timeperiod_type` установлен в ”ежемесячно” и задан `day`: 1. Для ежемесячных периодов, когда задан `dayofweek`, свойство `every` определяет неделю месяца, когда обслуживание должно вступать в силу. Возможные значения, если `timeperiod_type` установлен в ”ежемесячно” и задан `dayofweek`: 1 - *(по умолчанию)* первая неделя; 2 - вторая неделя; 3 - третья неделя; 4 - четвертая неделя; 5 - последняя неделя. Поведение свойства: - *поддерживается*, если `timeperiod_type` установлен в ”ежедневно”, ”еженедельно” или ”ежемесячно”
dayofweek	integer	Дни недели, когда обслуживание должно вступать в силу. Возможные значения битовой маски: 1 - понедельник; 2 - вторник; 4 - среда; 8 - четверг; 16 - пятница; 32 - суббота; 64 - воскресенье. Это поле битовой маски; допустима любая сумма возможных значений битовой маски (например, 21 для понедельника, среды и пятницы). Поведение свойства: - *обязательно*, если `timeperiod_type` установлен в ”еженедельно” или если `timeperiod_type` установлен в ”ежемесячно” и `day` не задан
day	integer	День месяца, когда обслуживание должно вступать в силу. Поведение свойства: - *обязательно*, если `timeperiod_type` установлен в ”ежемесячно” и `dayofweek` не задан

Свойство	Type	Описание
month	integer	Месяцы, когда обслуживание должно вступать в силу. Возможные значения битовой маски: 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 значений битовой маски (например, 585 для января, апреля, июля и октября). Поведение свойства: - *обязательно*, если `timeperiod_type` установлен в ”ежемесячно”

Тег проблемы

Объект тега проблемы используется для определения того, какие проблемы должны быть подавлены, когда обслуживание вступает в силу. Теги можно указывать только если `maintenance_type` объекта обслуживания установлен в значение ”with data collection”. Он имеет следующие свойства.

Свойство	Type	Описание
tag	string	Имя тега проблемы. Поведение свойства: - *обязательно*
operator	integer	Оператор условия. Возможные значения: 0 - Равно; 2 - *(по умолчанию)* Содержит.
value	string	Значение тега проблемы.

maintenance.create

Описание

`object maintenance.create( / maintenances)`

Этот метод позволяет создавать новые обслуживания.

Note:
Этот метод доступен только пользователям с типом доступа *Администратор* и *Супер-администратор*. Разрешения на вызов метода можно отозвать в настройках ролей пользователя. Дополнительную информацию см. в разделе Роли пользователей.

Параметры

`(object/array)` Обслуживания для создания.

В дополнение к стандартным свойствам обслуживания, метод принимает следующие параметры.

Параметр	Type	Описание
groups	object/array	Группы узлов сети, для которых будет выполняться обслуживание. Для групп узлов сети должно быть определено только свойство `groupid`. Поведение параметра: - *обязателен*, если `hosts` не задан
hosts	object/array	Узлы сети, для которых будет выполняться обслуживание. Для узлов сети должно быть определено только свойство `hostid`. Поведение параметра: - *обязателен*, если `groups` не задан
timeperiods	object/array	Периоды времени обслуживания. Поведение параметра: - *обязателен*
tags	object/array	Теги проблем. Определяют, какие проблемы должны быть подавлены. Если теги не указаны, будут подавлены все проблемы узлов сети с активным обслуживанием. Поведение параметра: - *поддерживается*, если `maintenance_type` объекта Объект обслуживания установлен в значение ”with data collection”

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 ID ”2” и с тегами проблем **service:mysqld** и **error**. Оно должно быть активно с 17.03.2026 по 17.03.2027, вступать в силу 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": "2"}
        ],
        "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
}
```

См. также

- Период времени

Источник

CMaintenance::create() в *ui/include/classes/api/services/CMaintenance.php*.

maintenance.delete

Описание

`object maintenance.delete(    maintenanceIds)`

Этот метод позволяет удалять периоды обслуживания.

> **Note:**
> Этот метод доступен только пользователям с типом доступа *Администратор* и *Супер-администратор*. Разрешения на вызов метода можно отозвать в настройках ролей пользователя. Дополнительную информацию см. в разделе Роли пользователей.

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)`

Этот метод позволяет получать обслуживания 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 желаемый вывод.

Метод поддерживает следующие параметры.

Параметр	Тип	Описание
groupids	ID/array	Возвращать только обслуживания, назначенные указанным группам узлов сети.
hostids	ID/array	Возвращать только обслуживания, назначенные указанным узлам сети.
maintenanceids	ID/array	Возвращать только обслуживания с указанными ID.
selectHostGroups	query	Возвращать свойство `hostgroups` с группами узлов сети, назначенными обслуживанию.
selectHosts	query	Возвращать свойство `hosts` с узлами сети, назначенными обслуживанию.
selectTags	query	Возвращать свойство `tags` с тегами проблем обслуживания.
selectTimeperiods	query	Возвращать свойство `timeperiods` с периодами времени 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	

Параметр	Тип	Описание
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 обслуживаний

Получение всех настроенных обслуживаний и данных о назначенных группах узлов сети, заданных периодах времени и тегах проблем.

Запрос:

```
{
    "jsonrpc": "2.0",
    "method": "maintenance.get",
    "params": {
        "output": "extend",
        "selectHostGroups": "extend",
        "selectTimeperiods": "extend",
        "selectTags": "extend"
    },
    "id": 1
}
```

Ответ:

```
{
    "jsonrpc": "2.0",
    "result": [
        {
            "maintenanceid": "3",
            "name": "Sunday maintenance",
            "maintenance_type": "0",
            "description": "",
            "active_since": "1358844540",
            "active_till": "1390466940",
            "tags_evaltype": "0",
            "hostgroups": [
                {
                    "groupid": "4",
                    "name": "Zabbix servers",
                    "flags": "0",
                    "uuid": "6f6799aa69e844b4b3918f779f2abf08"
                }
            ],
            "timeperiods": [
                {
                    "timeperiod_type": "3",
                    "every": "1",
                    "month": "0",
                    "dayofweek": "1",
                    "day": "0",
                    "start_time": "64800",
                    "period": "3600",
                    "start_date": "2147483647"
                }
            ],
```

```
            "tags": [
                {
                    "tag": "service",
                    "operator": "0",
                    "value": "mysqld",
                },
                {
                    "tag": "error",
                    "operator": "2",
                    "value": ""
                }
            ]
        }
    ],
    "id": 1
}
```

См. также

- Узел сети
- Группа узлов сети
- Период времени

Исходный код

CMaintenance::get() в *ui/include/classes/api/services/CMaintenance.php*.

maintenance.update

Описание

`object maintenance.update( / maintenances)`

Этот метод позволяет обновлять существующие обслуживания.

Note:
Этот метод доступен только пользователям с типом доступа *Администратор* и *Супер-администратор*. Разрешения на вызов метода можно отозвать в настройках ролей пользователя. Дополнительную информацию см. в разделе Роли пользователей.

Параметры

`(object/array)` Свойства обслуживания, которые необходимо обновить.

Свойство `maintenanceid` должно быть определено для каждого обслуживания, все остальные свойства являются необязательными. Будут обновлены только переданные свойства, все остальные останутся без изменений.

В дополнение к стандартным свойствам обслуживания, метод принимает следующие параметры.

Параметр	Тип	Описание
groups	object/array	Группы узлов сети для замены текущих групп. Для групп узлов сети должно быть определено только свойство `groupid`. Поведение параметра: - *обязательно*, если `hosts` не задан
hosts	object/array	Узлы сети для замены текущих узлов сети. Для узлов сети должно быть определено только свойство `hostid`. Поведение параметра: - *обязательно*, если `groups` не задан
timeperiods	object/array	Периоды времени обслуживания для замены текущих периодов.

Параметр	Тип	Описание
tags	object/array	Теги проблем для замены текущих тегов. Поведение параметра: - *поддерживается*, если `maintenance_type` объекта Объект обслуживания установлен в значение "with data collection"

Возвращаемые значения

( ) Возвращает объект, который содержит ID обновленных обслуживаний, указанных в свойстве `maintenanceids`.

Примеры

Назначение других узлов сети

Замените узлы сети, в данный момент назначенные обслуживанию, 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*.

Отчёт

Этот класс предназначен для работы с запланированными отчётами.

Ссылки на объекты:

- Report
- Users
- User groups

Доступные методы:

- report.create - создать новые запланированные отчёты
- report.delete - удалить запланированные отчёты
- report.get - получить запланированные отчёты
- report.update - обновить запланированные отчёты

Объект отчёта

Следующие объекты напрямую связаны с API `report`.

Отчет

Объект отчета имеет следующие свойства:

Свойство	Тип	Описание
reportid	ID	ID отчета. Поведение свойства: - *только для чтения* - *обязательно* для операций обновления
userid	ID	ID пользователя, создавшего отчет. Поведение свойства: - *обязательно* для операций создания
name	string	Уникальное имя отчета. Поведение свойства: - *обязательно* для операций создания
dashboardid	ID	ID 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	Расписание повторения периода. 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.
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 битовой маски; допустима любая сумма возможных значений битовой маски (например, 21 для понедельника, среды и пятницы). По умолчанию: 0. Поведение свойства: - *обязательно*, если `cycle` установлено в ”weekly”.

Свойство	Тип	Описание
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	Включен или отключен отчет. Возможные значения: 0 - Отключен; 1 - *(по умолчанию)* Включен.
description	text	Описание отчета.
state	integer	Состояние отчета. Возможные значения: 0 - *(по умолчанию)* отчет еще не был обработан; 1 - отчет был сгенерирован и успешно отправлен всем получателям; 2 - не удалось сгенерировать отчет; ”info” содержит информацию об ошибке; 3 - отчет был сгенерирован, но отправка некоторым (или всем) получателям не удалась; ”info” содержит информацию об ошибке. Поведение свойства: - *только для чтения*
lastsent	timestamp	Unix timestamp 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:

Свойство	Тип	Описание
userid	ID	ID пользователя, которому будет отправлен отчет. Поведение свойства: - *обязательно*
access_userid	ID	ID пользователя, от имени которого будет сформирован отчет. 0 - *(по умолчанию)* Сформировать отчет получателем.
exclude	integer	Следует ли исключить пользователя из списка рассылки. Возможные значения: 0 - *(по умолчанию)* Включить; 1 - Исключить.

Группы пользователей

Объект групп пользователей имеет следующие свойства:

Свойство	Тип	Описание
usrgrpid	ID	ID группы пользователей, которой будет отправлен отчет. Поведение свойства: - *обязательно*
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( / reports)`

Этот метод позволяет создавать новые запланированные отчеты.

Note:
Этот метод доступен только пользователям с типом доступа *Администратор* и *Супер-администратор*. Разрешения на вызов метода можно отозвать в настройках ролей пользователя. Дополнительную информацию см. в разделе Роли пользователей.

Параметры

`(object/array)` Запланированные отчёты для создания.

В дополнение к стандартным свойствам запланированного отчёта, метод принимает следующие параметры.

Параметр	Тип	Описание
users	object/array	Пользователи, которым будет отправлен отчёт. Поведение параметра: - *обязателен*, если `user_groups` не задан
user_groups	object/array	Группы пользователей, которым будет отправлен отчёт. Поведение параметра: - *обязателен*, если `users` не задан

Возвращаемые значения

`(object)` Возвращает объект, содержащий идентификаторы созданных запланированных отчётов в свойстве `reportids`. Порядок возвращаемых идентификаторов соответствует порядку переданных запланированных отчётов.

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(    reportids)`

Этот метод позволяет удалять запланированные отчеты.

Note:
Этот метод доступен только пользователям с типом доступа *Администратор* и *Супер-администратор*. Разрешения на вызов метода можно отозвать в настройках ролей пользователя. Дополнительную информацию см. в разделе Роли пользователей.

Параметры

(array) Идентификаторы запланированных отчётов, которые необходимо удалить.

Возвращаемые значения

(object) Возвращает объект, содержащий идентификаторы удалённых запланированных отчётов 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( )

Метод позволяет получать запланированные отчеты по заданным параметрам.

> **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) Параметры, определяющие желаемый вывод.

Метод поддерживает следующие параметры.

Параметр	Тип	Описание
reportids	ID/array	Возвращать только запланированные отчёты с указанными ID отчётов.
expired	boolean	Если установлено значение true, возвращать только просроченные запланированные отчёты.
selectUsers	query	Возвращать свойство users с пользователями, которым настроена отправка отчёта.
selectUserGroups	query	Возвращать свойство user_groups с группами пользователей, которым настроена отправка отчёта.

Параметр	Тип	Описание
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 отчёта

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(    /    reports)`

Этот метод позволяет обновлять существующие запланированные отчеты.

Note:
Этот метод доступен только пользователям с типом доступа *Администратор* и *Супер-администратор*. Разрешения на вызов метода можно отозвать в настройках ролей 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.

Параметр	Тип	Описание
users	object/array	Пользователи для замены текущих пользователей, назначенных запланированному отчету. Поведение параметра: - *обязательно*, если `user_groups` не задан
user_groups	object/array	Группы пользователей для замены текущих групп пользователей, назначенных запланированному отчету. Поведение параметра: - *обязательно*, если `users` не задан

Возвращаемые значения

`(object)` Возвращает объект, содержащий идентификаторы обновлённых запланированных отчётов 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*.

Очистка истории

Этот класс предназначен для работы с очисткой истории.

Ссылки на объекты:

- Housekeeping

Доступные методы:

- housekeeping.get - получить настройки очистки истории
- housekeeping.update - обновить настройки очистки истории

Объект housekeeping

Следующие объекты напрямую связаны с API `housekeeping`.

Очистка устаревших данных

Объект настроек имеет следующие свойства.

Свойство	Type	Описание
hk_events_mode	integer	Включить внутреннюю очистку устаревших данных для событий и оповещений. Возможные значения: 0 - Отключить; 1 - *(по умолчанию)* Включить.
hk_events_trigger	string	Период хранения данных триггеров. Принимает секунды и единицу времени с суффиксом. По умолчанию: 365d.
hk_events_service	string	Период хранения данных служб. Принимает секунды и единицу времени с суффиксом. По умолчанию: 1d.
hk_events_internal	string	Период хранения внутренних данных. Принимает секунды и единицу времени с суффиксом. По умолчанию: 1d.
hk_events_discovery	string	Период хранения данных сетевого обнаружения. Принимает секунды и единицу времени с суффиксом. По умолчанию: 1d.
hk_events_autoreg	string	Период хранения данных авторегистрации. Принимает секунды и единицу времени с суффиксом. По умолчанию: 1d.
hk_services_mode	integer	Включить внутреннюю очистку устаревших данных для служб. Возможные значения: 0 - Отключить; 1 - *(по умолчанию)* Включить.
hk_services	string	Период хранения данных служб. Принимает секунды и единицу времени с суффиксом. По умолчанию: 365d.
hk_audit_mode	integer	Включить внутреннюю очистку устаревших данных для аудита. Возможные значения: 0 - Отключить; 1 - *(по умолчанию)* Включить.

Свойство	Type	Описание
hk_audit	string	Период хранения данных аудита. Принимает секунды и единицу времени с суффиксом. По умолчанию: 31d.
hk_sessions_mode	integer	Включить внутреннюю очистку устаревших данных для сессий. Возможные значения: 0 - Отключить; 1 - *(по умолчанию)* Включить.
hk_sessions	string	Период хранения данных сессий. Принимает секунды и единицу времени с суффиксом. По умолчанию: 31d.
hk_history_mode	integer	Включить внутреннюю очистку устаревших данных для истории. Возможные значения: 0 - Отключить; 1 - *(по умолчанию)* Включить.
hk_history_global	integer	Переопределить период хранения истории элемента данных. Возможные значения: 0 - Не переопределять; 1 - *(по умолчанию)* Переопределять.
hk_history	string	Период хранения исторических данных. Принимает секунды и единицу времени с суффиксом. По умолчанию: 31d.
hk_trends_mode	integer	Включить внутреннюю очистку устаревших данных для трендов. Возможные значения: 0 - Отключить; 1 - *(по умолчанию)* Включить.
hk_trends_global	integer	Переопределить период хранения трендов элемента данных. Возможные значения: 0 - Не переопределять; 1 - *(по умолчанию)* Переопределять.
hk_trends	string	Период хранения данных трендов. Принимает секунды и единицу времени с суффиксом. По умолчанию: 365d.
db_extension	string	Флаг конфигурации расширения БД. Если этот флаг установлен в ”timescaledb”, то сервер изменяет свое поведение при очистке устаревших данных и удалении элементов данных. Property behavior: - *read-only*
compression_availability	integer	Поддерживается ли сжатие данных базой данных (или ее расширением). Возможные значения: 0 - Недоступно; 1 - Доступно. Property behavior: - *read-only*
compression_status	integer	Включить сжатие TimescaleDB для истории и трендов. Возможные значения: 0 - *(по умолчанию)* Выключено; 1 - Включено.

Свойство	Type	Описание
compress_older	string	Сжимать записи истории и трендов старше указанного периода. Принимает секунды и единицу времени с суффиксом. По умолчанию: 7d.

housekeeping.get

Описание

`object housekeeping.get(  )`

Метод позволяет получить объект обслуживания в соответствии с заданными параметрами.

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)` Параметры, определяющие желаемый вывод.

Метод поддерживает только один параметр.

Параметр	Тип	Описание
output	query	Этот параметр описан в справочном комментарии.

Возвращаемые значения

`(object)` Возвращает объект housekeeper.

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(  )`

Этот метод позволяет обновить существующие настройки управления.

> **Note:**
> Этот метод доступен только пользователю с типом доступа *Супер-администратор*. Разрешения на вызов метода можно отозвать в настройках ролей пользователя. Дополнительную информацию см. в разделе Роли пользователей.

Параметры

`(object)` Свойства очистки истории, которые необходимо обновить.

Возвращаемые значения

`(array)` Возвращает массив с именами обновлённых 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*.

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 мониторинга

Объект панели мониторинга имеет следующие свойства.

Свойство	Тип	Описание
dashboardid	ID	ID панели мониторинга. Поведение свойства: - *только для чтения* - *обязательно* для операций обновления
name	string	Имя панели мониторинга. Поведение свойства: - *обязательно* для операций создания
userid	ID	ID пользователя, который является владельцем панели мониторинга.
private	integer	Тип общего доступа к панели мониторинга. Возможные значения: 0 - общедоступная панель мониторинга; 1 - *(по умолчанию)* частная панель мониторинга.
display_period	integer	Период отображения страницы по умолчанию (в секундах). Возможные значения: 10, 30, 60, 120, 600, 1800, 3600. По умолчанию: 30.
auto_start	integer	Автоматически запускать слайд-шоу. Возможные значения: 0 - не запускать слайд-шоу автоматически; 1 - *(по умолчанию)* запускать слайд-шоу автоматически.

Страница панели мониторинга

Объект страницы панели мониторинга имеет следующие свойства.

Свойство	Тип	Описание
dashboard_pageid	ID	ID страницы панели мониторинга. Поведение свойства: - *только для чтения*
name	string	Имя страницы панели мониторинга. По умолчанию: пустая строка.
display_period	integer	Период отображения страницы панели мониторинга (в секундах). Возможные значения: 0, 10, 30, 60, 120, 600, 1800, 3600. По умолчанию: 0 (будет использоваться период отображения страницы по умолчанию).
widgets	array	Массив объектов виджета панели мониторинга.

Виджет панели мониторинга

Объект виджета панели мониторинга имеет следующие свойства.

Свойство	Тип	Описание
widgetid	ID	ID виджета панели мониторинга. Поведение свойства: - *только для чтения*

Свойство	Тип	Описание
type	string	Тип виджета панели мониторинга. Возможные значения: actionlog - Журнал действий; clock - Часы; discovery - Состояние обнаружения; favgraphs - Избранные графики; favmaps - Избранные карты; gauge - Шкала; 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 картам; piechart - Круговая диаграмма; problemhosts - Узлы сети с проблемами; problems - Проблемы; problemsbysv - Проблемы по важности; 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 - Веб-мониторинг. Поведение свойства: - *обязательное*
name	string	Пользовательское имя виджета.
x	integer	Горизонтальная позиция от левой стороны панели мониторинга. Возможные значения находятся в диапазоне от 0 до 71.
y	integer	Вертикальная позиция от верхней стороны панели мониторинга. Возможные значения находятся в диапазоне от 0 до 63.
width	integer	Ширина виджета. Возможные значения находятся в диапазоне от 1 до 72.
height	integer	Высота виджета. Возможные значения находятся в диапазоне от 1 до 64.
view_mode	integer	Режим отображения виджета. Возможные значения: 0 - *(по умолчанию)* стандартный вид виджета; 1 - со скрытым заголовком;

Свойство	Тип	Описание
fields	array	Массив объектов поля виджета панели мониторинга. Поведение свойства: - см. отдельные виджеты в разделе Поля виджетов панели мониторинга

Поле виджета панели мониторинга

Объект поля виджета панели мониторинга имеет следующие свойства.

Свойство	Тип	Описание
type	integer	Тип поля виджета. Возможные значения: 0 - Целое число; 1 - Строка; 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. Поведение свойства: - *обязательно*
name	string	Имя поля виджета. Возможные значения: см. Поля виджета панели мониторинга. Поведение свойства: - *обязательно*
value	mixed	Значение поля виджета в зависимости от типа. Возможные значения: см. Поля виджета панели мониторинга. Поведение свойства: - *обязательно*

Группа пользователей панели мониторинга

Список разрешений панели мониторинга на основе групп пользователей. Имеет следующие свойства.

Свойство	Тип	Описание
usrgrpid	ID	ID группы пользователей. Поведение свойства: - *обязательно*

Свойство	Тип	Описание
permission	integer	Тип уровня разрешений. Возможные значения: 2 - только чтение; 3 - чтение-запись. Поведение свойства: - *обязательно*

Пользователь панели мониторинга

Список разрешений панели мониторинга по пользователям. Имеет следующие свойства.

Свойство	Type	Описание
userid	ID	ID пользователя. Поведение свойства: - *обязательно*
permission	integer	Тип уровня разрешений. Возможные значения: 2 - только чтение; 3 - чтение-запись. Поведение свойства: - *обязательно*

dashboard.create

Описание

`object dashboard.create( / dashboards)`

Этот метод позволяет создавать новые панели.

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)` Панели мониторинга для создания.

В дополнение к стандартным свойствам панели мониторинга, метод принимает следующие параметры.

Параметр	Тип	Описание
pages	array	Страницы панели мониторинга, которые должны быть созданы для панели мониторинга. Страницы панели мониторинга будут упорядочены в том же порядке, в котором они указаны. Поведение параметра: - *обязательный*
users	array	Общие доступы пользователя панели мониторинга, которые должны быть созданы для панели мониторинга.
userGroups	array	Общие доступы группы пользователей панели мониторинга, которые должны быть созданы для 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

Создайте панель 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.

Запрос:

```
{
    "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
}
```

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::create() в *ui/include/classes/api/services/CDashboard.php*.

dashboard.delete

Описание

`object dashboard.delete(    dashboardids)`

Этот метод позволяет удалять панели.

> **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

(   ) ID удаляемых панелей.

Возвращаемые значения

(   ) Возвращает объект, который содержит ID удаленных панелей, указанных в свойстве `dashboardids`.

Примеры

Удаление нескольких панелей

Удалите две панели.

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(parameters)`

Этот метод позволяет получать панели в соответствии с заданными параметрами.

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)` Параметры, определяющие желаемый вывод.

Метод поддерживает следующие параметры.

Параметр	Тип	Описание
dashboardids	ID/array	Возвращать только панели с указанными ID.
selectPages	query	Возвращать свойство `pages` со страницами панели, упорядоченными должным образом.
selectUsers	query	Возвращать свойство `users` с пользователями, которым предоставлен общий доступ к панели.
selectUserGroups	query	Возвращать свойство `userGroups` с группами пользователей, которым предоставлен общий доступ к панели.
sortfield	string/array	Сортировать результат по указанным свойствам. Возможные значения: `dashboardid`.
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 панели по ID

Получить все данные о панелях мониторинга ”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(  /  dashboards)`

Этот метод позволяет обновлять существующие панели.

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)` Свойства панели мониторинга, которые требуется обновить.

Свойство `dashboardid` должно быть определено для каждой панели мониторинга, все остальные свойства являются необязательными. Будут обновлены только переданные свойства, все остальные останутся без изменений.

В дополнение к стандартным свойствам панели мониторинга, метод принимает следующие параметры.

Параметр	Type	Описание
pages	array	Страницы панели мониторинга для замены существующих страниц панели мониторинга. Страницы панели мониторинга обновляются по свойству `dashboard_pageid`. Для объектов без свойства `dashboard_pageid` будут созданы новые страницы панели мониторинга, а существующие страницы панели мониторинга будут удалены, если 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	Общие доступы пользователей панели мониторинга для замены существующих элементов.
userGroups	array	Общие доступы групп пользователей панели мониторинга для замены существующих элементов.

Возвращаемые значения

( ) Возвращает объект, который содержит ID обновленных панелей, указанных в свойстве `dashboardids`.

Примеры

Переименование панели

Переименуйте панель 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

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": "dashboard.update",
    "params": {
        "dashboardid": "2",
        "pages": [
            {
                "dashboard_pageid": 1,
                "name": "              "
            },
            {
                "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 для администраторов и суперадминистраторов.

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

Эта страница содержит навигационные ссылки на параметры виджетов панели мониторинга и возможные значения свойств для соответствующих объектов поля виджета панели мониторинга.

Чтобы увидеть параметры и значения свойств для каждого виджета, используйте боковую панель для перехода на страницу соответствующего виджета.

1. Журнал действий

Описание

Эти параметры и возможные значения свойств для соответствующих объектов полей виджета 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 некорректного создания или обновления виджетов. Чтобы обеспечить успешное создание или обновление виджета *Журнал действий*, ознакомьтесь с поведением параметров, описанным в таблицах ниже.

Параметры

Для виджета *Журнал действий* 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.
Получатели	11	userids.0	ID пользователя. Примечание: Чтобы настроить нескольких пользователей, создайте объект поля виджета панели с увеличивающимся номером в имени свойства для каждого пользователя.
Действия	12	actionids.0	ID действия. Примечание: Чтобы настроить несколько действий, создайте объект поля виджета панели с увеличивающимся номером в имени свойства для каждого действия.
Типы медиа	13	mediatypeids.0	ID типа медиа. Примечание: Чтобы настроить несколько типов медиа, создайте объект поля виджета панели с увеличивающимся номером в имени свойства для каждого типа медиа.
Состояние	0	statuses.0	0 - В процессе; 1 - Отправлено/Выполнено; 2 - Ошибка. Примечание: Чтобы настроить несколько значений, создайте объект поля виджета панели с увеличивающимся номером в имени свойства для каждого значения.
Строка поиска	1	message	Любое строковое значение.
Период времени	1	time_period._referenceDASHBOARD._timeperiod	DASHBOARD._timeperiod - использовать селектор периода времени панели как источник данных; ABCDE._timeperiod - использовать совместимый виджет (с параметром *Reference*, установленным в "ABCDE") как источник 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 (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

Параметр	type	name	value
Сортировать записи по	0	sort_triggers	3 - Время (по возрастанию); 4 - *(по умолчанию)* Время (по убыванию); 5 - Тип (по возрастанию); 6 - Тип (по убыванию); 7 - Состояние (по возрастанию); 8 - Состояние (по убыванию); 11 - Получатель (по возрастанию); 12 - Получатель (по убыванию).
Показывать строк	0	show_lines	Возможные значения: от 1 до 100. По умолчанию: 25.

Примеры

Следующие примеры предназначены только для описания настройки объектов полей виджета панели мониторинга для виджета *Журнал действий*. Для получения дополнительной информации о настройке панели мониторинга см. dashboard.create.

Настройка виджета *Журнал действий*

Настройте виджет *Журнал действий*, который отображает 10 записей с подробностями операций действий, отсортированных по времени (в порядке возрастания). Кроме того, отображайте подробности только для тех операций действий, в которых была предпринята попытка отправить электронное письмо пользователю "1", но она оказалась неуспешной.

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 позволяют настроить виджет *Clock* в методах `dashboard.create` и `dashboard.update`.

Attention:
Свойства `fields` виджета не проверяются при создании или обновлении панели. Это позволяет пользователям изменять встроенные виджеты и создавать пользовательские виджеты, но также создает риск некорректного создания или обновления виджетов. Чтобы обеспечить успешное создание или обновление виджета *Clock*, ознакомьтесь с поведением параметров, описанным в таблицах ниже.

Параметры

Для виджета *Часы* 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.
Тип времени	0	time_type	0 - *(по умолчанию)* Локальное время; 1 - Время сервера; 2 - Время узла сети.
Тип часов	0	clock_type	0 - *(по умолчанию)* Аналоговые; 1 - Цифровые.

Если *Тип времени* установлен в значение ”Время узла сети”, поддерживаются следующие параметры.

Параметр	type	name	value
Элемент данных	4	itemid.0	ID элемента данных. Поведение параметра: - *обязательно*

Если *Тип часов* установлен в значение ”Цифровые”, поддерживаются следующие параметры.

Параметр	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 с увеличивающимся номером в имени свойства для каждого значения.

Расширенная конфигурация

Следующие параметры расширенной конфигурации поддерживаются, если для *Clock type* установлено значение «Digital».

Параметр	type	name	value
Цвет фона	1	bg_color	Шестнадцатеричный код цвета (например, FF0000). По умолчанию: "" (пусто).

Дата

Следующие дополнительные параметры конфигурации поддерживаются, если для *Clock type* установлено значение ”Digital”, а для *Show* — значение ”Date”.

Параметр	type	name	value
Жирный	0	date_bold	0 - *(по умолчанию)* Отключено; 1 - Включено.
Цвет	1	date_color	Шестнадцатеричный код цвета (например, FF0000). По умолчанию: "" (пусто).

Время

Следующие дополнительные параметры конфигурации поддерживаются, если *Тип часов* установлен в значение ”Digital”, а *Показывать* — в значение ”Time”.

Параметр	type	name	value
Жирный	0	time_bold	0 - *(по умолчанию)* Отключено; 1 - Включено.
Цвет	1	time_color	Шестнадцатеричный код цвета (например, FF0000). По умолчанию: "" (пусто).
Секунды	0	time_sec	0 - Отключено; 1 - *(по умолчанию)* Включено.
Формат	0	time_format	0 - *(по умолчанию)* 24-часовой; 1 - 12-часовой.

Часовой пояс

Следующие расширенные параметры конфигурации поддерживаются, если для *Clock type* установлено значение ”Digital”, а для *Show* — значение ”Time zone”.

Параметр	type	name	value
Жирный	0	tzone_bold	0 — *(по умолчанию)* Отключено; 1 — Включено.
Цвет	1	tzone_color	Шестнадцатеричный код цвета (например, FF0000). По умолчанию: "" (пусто).
Часовой пояс	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: - *поддерживается*, если для *Time type* установлено значение ”Local time” или ”Server time”
Формат	0	tzone_format	0 — *(по умолчанию)* Краткий; 1 — Полный. Поведение параметра: - *поддерживается*, 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*. Для получения дополнительной информации о настройке панели мониторинга см. dashboard.create.

Настройка виджета *Clock*

Настройте виджет *Clock*, отображающий 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. Состояние обнаружения

Описание

Эти параметры и возможные значения свойств для соответствующих объектов полей виджета панели мониторинга позволяют настраивать виджет *Статус обнаружения* в методах `dashboard.create` и `dashboard.update`.

Параметры

Для виджета *Состояние обнаружения* 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.

Примеры

Следующие примеры предназначены только для описания настройки объектов полей виджета панели мониторинга для виджета *Состояние обнаружения*. Для получения дополнительной информации о настройке панели мониторинга смотрите `dashboard.create`.

Настройка виджета *Статус обнаружения*

Настройте виджет *Статус обнаружения* 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 для соответствующих объектов полей виджета панели 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 настройки объектов полей виджета панели мониторинга для виджета *Избранные графики*. Для получения дополнительной информации о настройке панели мониторинга смотрите `dashboard.create`.

Настройка виджета *Избранные графики*

Настройте виджет *Избранные графики*, установив интервал обновления 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 сети

Описание

Эти параметры и возможные значения свойств для соответствующих объектов полей виджета панели мониторинга позволяют настраивать виджет *Избранные карты* в методах dashboard.create и dashboard.update.

Параметры

Для виджета *Избранные карты* 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 настройки объектов полей виджета панели мониторинга для виджета *Избранные карты*. Для получения дополнительной информации о настройке панели мониторинга смотрите dashboard.create.

Настройка виджета *Избранные карты*

Настройте виджет *Избранные карты* с интервалом обновления 10 минут.

Запрос:

```
{
    "jsonrpc": "2.0",
    "method": "dashboard.create",
    "params": {
        "name": "      ",
        "display_period": 30,
        "auto_start": 1,
        "pages": [
            {
                "widgets": [
```

```
                    {
                        "type": "favmaps",
                        "name": "          ",
                        "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. Индикатор

Описание

Эти параметры и возможные значения свойств для соответствующих объектов полей виджета позволяют настраивать виджет *Gauge* в методах `dashboard.create` и `dashboard.update`.

Attention:
Свойства `fields` виджета не проверяются при создании или обновлении панели мониторинга. Это позволяет пользователям изменять встроенные виджеты и создавать пользовательские виджеты, но также создает риск некорректного создания или обновления виджетов. Чтобы обеспечить успешное создание или обновление виджета *Gauge*, обратитесь к описанию поведения параметров в таблицах ниже.

Параметры

Для виджета *Gauge* 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: - *обязательно*, если *Элемент данных (виджет)* не задан
Элемент данных (виджет)	1	itemid._reference	Вместо ID элемента данных: `ABCDE._itemid` — задать совместимый виджет (с параметром *Reference*, установленным в "ABCDE") в качестве источника данных для элементов данных. Поведение параметра: - *обязательно*, если *Элемент данных* не задан
Минимум	1	min	Любое числовое значение. Поддерживаются суффиксы (например, "1d", "2w", "4K", "8G"). По умолчанию: "0".
Максимум	1	max	Любое числовое значение. Поддерживаются суффиксы (например, "1d", "2w", "4K", "8G"). По умолчанию: "100".
Дуга значения	1	value_arc_color	Шестнадцатеричный код цвета (например, `FF0000`). По умолчанию: "" (пусто).
Фон дуги	1	empty_color	Шестнадцатеричный код цвета (например, `FF0000`). По умолчанию: "" (пусто).
Фон	1	bg_color	Шестнадцатеричный код цвета (например, `FF0000`). По умолчанию: "" (пусто).

Параметр	type	name	value
Показывать	0	show.0	1 - Описание; 2 - Значение; 3 - Стрелка; 4 - Шкала; 5 - Дуга значения. Примечание: Чтобы настроить несколько значений, создайте объект поля виджета панели с увеличивающимся номером в имени свойства для каждого значения. По умолчанию: 1, 2, 4, 5. Значения ”Стрелка” и ”Шкала” не поддерживаются, если одновременно: - не задан ни один объект поля виджета панели для *Показывать* со значением ”Дуга значения”; - параметр расширенной конфигурации *Показывать дугу* установлен в значение ”Отключено”. Параметры расширенной конфигурации для опций *Показывать* не поддерживаются, если не заданы объекты поля виджета панели с соответствующими значениями.
Переопределить узел сети	1	override_hostid._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.

Расширенная настройка

Для виджета *Gauge* поддерживаются следующие параметры расширенной настройки.

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 в веб-интерфейс Zabbix (например, `"thresholds.0.threshold":"5"` → `"thresholds.0.threshold":"1"`; `"thresholds.1.threshold":"1"` → `"thresholds.1.threshold": "5"`).

Параметр	type	name	value
Угол	0	angle	Возможные 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	Возможные значения: от 1 до 100. По умолчанию: 15.
Вертикальное положение	0	desc_v_pos	0 - Сверху; 1 - *(по умолчанию)* Снизу.
Жирный	0	desc_bold	0 - *(по умолчанию)* Отключено; 1 - Включено.
Цвет	1	desc_color	Шестнадцатеричный код цвета (например, FF0000). По умолчанию: "" (пусто).

Параметр	type	name	value
Значение			
Десятичные знаки	0	decimal_places	Возможные значения: от 1 до 10. По умолчанию: 2.
Размер	0	value_size	Возможные значения: от 1 до 100. По умолчанию: 25.
Жирный	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 *Единицы измерения* (флажок) установлено в значение ”Включено”
Размер	0	units_size	Возможные значения: от 1 до 100. По умолчанию: 25. Поведение параметра: - *поддерживается*, если *Единицы измерения* (флажок) установлено в значение ”Включено”
Жирный	0	units_bold	0 - *(по умолчанию)* Отключено; 1 - Включено. Поведение параметра: - *поддерживается*, если *Единицы измерения* (флажок) установлено в значение ”Включено”
Положение	0	units_pos	0 - Перед значением; 1 - Над значением; 2 - *(по умолчанию)* После значения; 3 - Под значением. Поведение параметра: - *поддерживается*, если *Единицы измерения* (флажок) установлено в значение ”Включено” Этот параметр игнорируется, если задана одна из следующих единиц измерения, связанных со временем: unixtime, uptime, s.
Цвет	1	units_color	Шестнадцатеричный код цвета (например, FF0000). По умолчанию: "" (пусто).
Дуга значения			
Размер дуги	0	value_arc_size	Возможные 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 задан объект поля виджета панели мониторинга для *Show* со значением ”Value arc”, или *Show arc* установлено в значение ”Включено”

Параметр	type	name	value
Шкала			
Показывать единицы измерения	0	scale_show_units	0 - Отключено; 1 - *(по умолчанию)* Включено. Поведение параметра: - *поддерживается*, если *Единицы измерения* (флажок) установлено в значение ”Включено” и либо задан объект поля виджета панели мониторинга для *Show* со значением ”Value arc”, либо *Show arc* установлено в значение ”Включено”
Размер	0	scale_size	Возможные значения: от 1 до 100. По умолчанию: 15. Поведение параметра: - *поддерживается*, если задан объект поля виджета панели мониторинга для *Show* со значением ”Value arc”, или *Show arc* установлено в значение ”Включено”
Десятичные знаки	0	scale_decimal_places	Возможные значения: от 1 до 10. По умолчанию: 0. Поведение параметра: - *поддерживается*, если задан объект поля виджета панели мониторинга для *Show* со значением ”Value arc”, или *Show arc* установлено в значение ”Включено”
Пороги			
Цвет	1	thresholds.0.color	Шестнадцатеричный код цвета (например, FF0000).
Порог	1	thresholds.0.threshold	Любое числовое значение. Поддерживаются суффиксы (например, ”1d”, ”2w”, ”4K”, ”8G”).
Показывать метки	0	th_show_labels	0 - *(по умолчанию)* Отключено; 1 - Включено. Поведение параметра: - *поддерживается*, если *Пороги* заданы и либо задан объект поля виджета панели мониторинга для *Show* со значением ”Value arc”, либо *Show arc* установлено в значение ”Включено”
Показывать дугу	0	th_show_arc	0 - *(по умолчанию)* Отключено; 1 - Включено. Поведение параметра: - *поддерживается*, если *Пороги* заданы
Размер дуги	0	th_arc_size	Возможные значения: от 1 до 100. По умолчанию: 5. Поведение параметра: - *поддерживается*, если *Show arc* установлено в значение ”Включено”

Примеры

Следующие примеры предназначены только для описания настройки объектов полей виджета панели мониторинга для виджета *Gauge*. Для получения дополнительной информации о настройке панели мониторинга см. `dashboard.create`.

Настройка виджета *Gauge*

Настройте виджет *Gauge*, отображающий значение элемента данных для элемента данных ”44474” (Interface enp0s3: Bits sent). Кроме того, выполните визуальную тонкую настройку виджета с помощью нескольких дополнительных 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 для соответствующих объектов полей виджета позволяют настраивать виджет *Geomap* в методах `dashboard.create` и `dashboard.update`.

Attention:
Свойства `fields` виджета не проверяются при создании или обновлении панели. Это позволяет пользователям изменять встроенные виджеты и создавать пользовательские виджеты, но также создает риск некорректного создания или обновления виджетов. Чтобы обеспечить успешное создание или обновление виджета *Geomap*, ознакомьтесь с поведением параметров, описанным в таблицах ниже.

Параметры

Для виджета *Geomap* 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.
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 вообще, либо должен быть настроен как минимум с одной группой узлов сети, к которой принадлежат 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` - 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			
Тип вычисления	0	evaltype	0 - *(по умолчанию)* И/Или; 2 - Или. Этот параметр не поддерживается при настройке виджета на панели шаблона.

Параметр	type	name	value
Имя тега	1	tags.0.tag	Любое строковое значение. Примечание: Число в имени свойства указывает порядок тега в списке вычисления тегов. Поведение параметра: - *обязательно* при настройке *Тегов*
Оператор	0	tags.0.operator	Этот параметр не поддерживается при настройке виджета на панели шаблона. 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: - *обязательно* при настройке *Тегов*
Значение тега	1	tags.0.value	Этот параметр не поддерживается при настройке виджета на панели шаблона. Любое строковое значение. Примечание: Число в имени свойства указывает порядок тега в списке вычисления тегов. Поведение параметра: - *обязательно* при настройке *Тегов* Этот параметр не поддерживается при настройке виджета на панели шаблона.
Начальный вид	1	default_view	Разделенные запятыми *широта*, *долгота*, *уровень масштабирования* (*необязательно*, возможные значения в диапазоне 0-30). Пример: `40.6892494,-74.0466891,10`.
Кластеризация			
Режим кластеризации	0	clustering_mode	0 - *(по умолчанию)* Авто; 1 - Уровень масштабирования.
Уровень масштабирования кластеризации	0	clustering_zoom_level	Возможные значения в диапазоне 0-30. По умолчанию: 0. Поведение параметра: - *поддерживается*, если *Режим кластеризации* установлен в "Уровень масштабирования"
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: - *обязательно*

Примеры

Следующие примеры предназначены только для описания настройки объектов полей виджета панели мониторинга для виджета *Geomap*. Дополнительную информацию о настройке панели мониторинга смотрите в `dashboard.create`.

Настройка виджета *Geomap*

Настройте виджет *Geomap*, отображающий 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 настраивать виджет *Graph* в методах `dashboard.create` и `dashboard.update`.

Attention:
Свойства `fields` виджета не проверяются при создании или обновлении панели мониторинга. Это позволяет пользователям изменять встроенные виджеты и создавать пользовательские виджеты, но также создает риск некорректного создания или обновления виджетов. Чтобы обеспечить успешное создание или обновление виджета *Graph*, ознакомьтесь с поведением параметров, описанным в таблицах ниже.

Параметры

Для виджета *График* 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	0	ds.0.dataset_type	0 - Список элементов данных; 1 - *(по умолчанию)* Шаблоны элементов данных.
Элементы данных	4	ds.0.itemids.0	ID элемента данных. При настройке виджета на шаблоне панели следует указывать только элементы данных, настроенные в шаблоне. Примечание: Чтобы настроить несколько элементов данных, создайте объект поля виджета панели для каждого элемента данных с увеличивающимся номером в имени свойства. Поведение параметра: - *обязателен*, если *Тип набора данных* установлен в значение "Список элементов данных" и *Элементы данных (виджет)* не задан

Параметр	type	name	value
Элементы данных (виджет)	1	ds.0.itemids.0._reference	Вместо 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 увеличивающимся номером в имени свойства. Поведение параметра: - *обязателен*, если *Тип набора данных* установлен в значение "Список элементов данных" и *Элементы данных* не заданы
Цвет	1	ds.0.color.0	Шестнадцатеричный код цвета (например, `FF0000`). Поведение параметра: - *обязателен*, если *Тип набора данных* установлен в значение "Список элементов данных"
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 шаблоне панели.
Шаблоны элементов данных	1	ds.0.items.0	Имя элемента данных или шаблон (например, "*: Number of processed *values per second"). При настройке виджета на шаблоне панели следует указывать только шаблоны для элементов данных, настроенных в шаблоне. Поведение параметра: - *обязателен*, если *Тип набора данных* установлен в значение "Шаблоны элементов данных"
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. Возможные значения: 0-11. По умолчанию: 0. Поведение параметра: - *поддерживается*, если *Тип набора данных* установлен в значение "Шаблоны элементов данных" и *Цвет* не задан
Отрисовка	0	ds.0.type	0 - *(по умолчанию)* Линия; 1 - Точки; 2 - Ступенчатая; 3 - Столбцы.
С накоплением	0	ds.0.stacked	0 - *(по умолчанию)* Отключено; 1 - Включено. Поведение параметра: - *поддерживается*, если *Отрисовка* установлена в значение "Линия", "Ступенчатая" или "Столбцы"

Параметр	type	name	value
Ширина	0	ds.0.width	Возможные значения: 1-10. По умолчанию: 1. Поведение параметра: - *поддерживается*, если *Отрисовка* установлена в значение ”Линия” или ”Ступенчатая”
Размер точки	0	ds.0.pointsize	Возможные значения: 1-10. По умолчанию: 3. Поведение параметра: - *поддерживается*, если *Отрисовка* установлена в значение ”Точки”
Прозрачность	0	ds.0.transparency	Возможные значения: 1-10. По умолчанию: 5.
Заливка	0	ds.0.fill	Возможные значения: 1-10. По умолчанию: 3. Поведение параметра: - *поддерживается*, если *Отрисовка* установлена в значение ”Линия” или ”Ступенчатая”
Отсутствующие данные	0	ds.0.missingdatafunc	0 - *(по умолчанию)* Нет; 1 - Соединять; 2 - Считать как 0; 3 - Последнее известное. Поведение параметра: - *поддерживается*, если *Отрисовка* установлена в значение ”Линия” или ”Ступенчатая”
Переопределить узел сети	1	ds.0.override_hostid	ABCDE - `hostid`- задать совместимый виджет (с его 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 шаблоне панели.
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 - min; 2 - max; 3 - avg; 4 - count; 5 - sum; 6 - first; 7 - last.
Интервал агрегации	1	ds.0.aggregate_interval	Допустимая строка времени (например, `3600`, `1h` и т. д.). Можно использовать суффиксы времени. По умолчанию: `1h`.

Параметр	type	name	value
Агрегировать	0	ds.0.aggregate_grouping	0 - *(по умолчанию)* Каждый элемент данных; 1 - Набор данных. Поведение параметра: - *поддерживается*, если *Функция агрегации* установлена в значение ”min”, ”max”, ”avg”, ”count”, ”sum”, ”first” или ”last”
Аппроксимация	0	ds.0.approximation	1 - min; 2 - *(по умолчанию)* avg; 4 - max; 7 - all.
Метка набора данных	1	ds.0.data_set_label	Любое строковое значение. По умолчанию: "" (пусто).
Теги элементов данных			
Тип вычисления	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 вычисления тегов. Поведение параметра: - *обязателен*, если настраиваются *Теги 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 вычисления тегов. Поведение параметра: - *обязателен*, если настраиваются *Теги элементов данных*
Значение тега	1	ds.0.item_tags.0.value	Любое строковое значение. Примечание: Число в имени свойства указывает порядок тега в списке вычисления тегов. Поведение параметра: - *обязателен*, если настраиваются *Теги элементов данных*

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.
Простые триггеры	0	simple_triggers	0 - *(по умолчанию)* Отключено; 1 - Включено.
Рабочее время	0	working_time	0 - *(по умолчанию)* Отключено; 1 - Включено.

Параметр	type	name	value
Имена узлов сети в метках	0	show_hostnames	0 - *(по умолчанию)* Авто; 1 - Показывать; 2 - Скрыть. Этот параметр не поддерживается при настройке виджета на шаблоне панели.
Линия процентиля (слева)			
Состояние	0	percentile_left	0 - *(по умолчанию)* Отключено; 1 - Включено. Поведение параметра: - *поддерживается*, если *Ось Y* (в настройке *Набор данных*) установлена в значение "Слева"
Значение	0	percentile_left_value	Возможные значения находятся в диапазоне от 1 до 100. Поведение параметра: - *поддерживается*, если *Ось Y* (в настройке *Набор данных*) установлена в значение "Слева"
Линия процентиля (справа)			
Состояние	0	percentile_right	0 - *(по умолчанию)* Отключено; 1 - Включено. Поведение параметра: - *поддерживается*, если *Ось Y* (в настройке *Набор данных*) установлена в значение "Справа"
Значение	0	percentile_right_value	Возможные значения находятся в диапазоне от 1 до 100. Поведение параметра: - *поддерживается*, если *Ось Y* (в настройке *Набор данных*) установлена в значение "Справа"

Период времени

Для настройки *Периода времени* поддерживаются следующие параметры.

Параметр	type	name	value
Период времени	1	time_period._reference	`DASHBOARD._timeperiod` — использовать селектор периода времени панели как источник данных; `ABCDE._timeperiod` — использовать совместимый виджет (с параметром *Reference*, установленным в "ABCDE") как источник 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 (`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`

Параметр	type	name	value
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`

Оси

Для настройки *осей* поддерживаются следующие параметры.

Параметр	type	name	value
Левая Y	0	lefty	0 - Отключено; 1 - *(по умолчанию)* Включено. Поведение параметра: - *поддерживается*, если для *Y-axis* (в настройках *Data set*) установлено значение ”Left”
Правая Y	0	righty	0 - *(по умолчанию)* Отключено; 1 - Включено. Поведение параметра: - *поддерживается*, если для *Y-axis* (в настройках *Data set*) установлено 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 измерения (тип)	0	lefty_units righty_units	0 - *(по умолчанию)* Авто; 1 - Статические.
Единицы измерения (значение)	1	lefty_static_units righty_static_units	Любое строковое значение. По умолчанию: "" (пусто).
Ось X	0	xaxis	0 - Отключено; 1 - *(по умолчанию)* Включено.

Условные обозначения

Для настройки *условных обозначений* поддерживаются следующие параметры.

Параметр	type	name	value
Показывать условные обозначения	0	legend	0 - Отключено; 1 - *(по умолчанию)* Включено.

Параметр	type	name	value
Отображать мин./сред./макс.	0	legend_statistic	0 - *(по умолчанию)* Отключено; 1 - Включено. Поведение параметра: - *поддерживается*, если для *Показывать условные обозначения* установлено значение "Включено"
Показывать функцию агрегации	0	legend_aggregation	0 - *(по умолчанию)* Отключено; 1 - Включено. Поведение параметра: - *поддерживается*, если для *Показывать условные обозначения* установлено значение "Включено"
Строки	0	legend_lines_mode	0 - *(по умолчанию)* Фиксированное; 1 - Переменное. Поведение параметра: - *поддерживается*, если для *Показывать условные обозначения* 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 *Показывать условные обозначения* установлено значение "Включено"
Количество столбцов	0	legend_columns	Допустимые значения: от 1 до 4. По умолчанию: 4. Поведение параметра: - *поддерживается*, если для *Показывать условные обозначения* установлено значение "Включено", а для *Отображать мин./сред./макс.* установлено значение "Отключено"

Проблемы

Для настройки *Проблем* поддерживаются следующие параметры.

Параметр	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 проблем	1	problemhosts.0	Имя узла сети. Примечание: Число в имени свойства указывает на настраиваемый узел сети. Чтобы настроить несколько узлов сети, создайте объект поля виджета панели с увеличивающимся номером в имени свойства для каждого узла сети. Этот параметр не поддерживается при настройке виджета на панели шаблона.

Параметр	type	name	value
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 с увеличивающимся номером в имени свойства.
Проблема	1	problem_name	Имя события проблемы (без учета регистра, полное имя или его часть).
Теги проблемы			
Тип вычисления	0	evaltype	0 - *(по умолчанию)* И/Или; 2 - Или.
Имя тега	1	tags.0.tag	Любое строковое значение. Примечание: Число в имени свойства указывает на порядок тега в списке вычисления тегов. Поведение параметра: - *обязательно* 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: Число в имени свойства указывает на порядок тега в списке вычисления тегов. Поведение параметра: - *обязательно* при настройке *Тегов проблемы*
Значение тега	1	tags.0.value	Любое строковое значение. Примечание: Число в имени свойства указывает на порядок тега в списке вычисления тегов. Поведение параметра: - *обязательно* при настройке *Тегов проблемы*

Переопределения

Для настройки *Переопределений* поддерживаются следующие параметры.

Note:
Первое число в имени свойства (например, or.0.hosts.0, or.0.items.0) обозначает конкретный набор данных, а второе число, если присутствует, обозначает настроенный узел сети или элемент данных.

Параметр	type	name	value
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 задавать только шаблоны для 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.
Отрисовка	0	or.0.type	0 - Линия; 1 - Точки; 2 - Ступенчатая; 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 точек	0	or.0.pointsize	Допустимые значения: от 1 до 10.
Отсутствующие данные	0	or.0.missingdatafunc	0 - Нет; 1 - Соединять; 2 - Считать как 0; 3 - Последнее известное значение.
Ось Y	0	or.0.axisy	0 - Левая; 1 - Правая.
Инвертировать значения	0	or.0.invert_values	0 - (по умолчанию) Отключено; 1 - Включено.
Сдвиг 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 настройки объектов полей виджета панели мониторинга для виджета *Graph*. Для получения дополнительной информации о настройке панели мониторинга смотрите `dashboard.create`.

Настройка виджета *Graph*

Настройте виджет *Graph* следующим образом:

- 2 набора данных, всего 9 элементов данных на 1 узле сети.
- Первый набор данных имеет тип "Item list" и состоит из 3 элементов данных, которые представлены линиями разного цвета, но с одинаковой шириной, прозрачностью и заполнением.
- Второй набор данных имеет тип "Item patterns", состоит из 6 элементов данных, имеет настроенную агрегацию и представлен линией с пользовательскими цветом, шириной, прозрачностью и заполнением.
- 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.
- График имеет две оси Y, из которых 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.

Запрос:

```
{
    "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
}
```

Ответ:

```
{
    "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 для соответствующих объектов полей виджета позволяют настраивать виджет *Graph (classic)* в методах `dashboard.create` и `dashboard.update`.

Attention:
Свойства `fields` виджета не проверяются при создании или обновлении панели мониторинга. Это позволяет пользователям изменять встроенные виджеты и создавать пользовательские виджеты, но также создает риск некорректного создания или обновления виджетов. Чтобы обеспечить успешное создание или обновление виджета *Graph (classic)*, ознакомьтесь с поведением параметров, описанным в таблицах ниже.

Параметры

Для виджета *Graph (classic)* 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.
Источник	0	source_type	0 - *(по умолчанию)* График; 1 - Простой график.
График	6	graphid.0	ID графика. Поведение параметра: - *обязателен*, если для *Источника* установлено значение "График"
График (виджет)	1	graphid._reference	Вместо ID графика: `ABCDE._graphid` — использовать совместимый виджет (с параметром *Reference*, установленным в значение "ABCDE") в качестве источника данных для графиков. Поведение параметра: - *обязателен*, если для *Источника* установлено значение "Simple graph" и *График* не задан
Элемент данных	4	itemid.0	ID элемента данных. Поведение параметра: - *обязателен*, если для *Источника* установлено значение "Simple graph" и *Элемент данных (виджет)* не задан
Элемент данных (виджет)	1	itemid._reference	Вместо ID элемента данных: `ABCDE._itemid` — использовать совместимый виджет (с параметром *Reference*, установленным в значение "ABCDE") в качестве источника данных для элементов данных. Поведение параметра: - *обязателен*, если для *Источника* установлено значение "Simple graph" и *Элемент данных* не задан
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 *Reference*, установленным в значение "ABCDE") в качестве источника данных. По умолчанию: `DASHBOARD._timeperiod` В качестве альтернативы можно задать период времени только в параметрах *От* и *До*.

Параметр	type	name	value
От	1	time_period.from	Допустимая строка времени в абсолютном формате (YYYY-MM-DD hh:mm:ss) или в 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 формате (YYYY-MM-DD hh:mm:ss) или в 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
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 *Reference*, установленным в значение "ABCDE") в качестве источника данных для узлов сети; DASHBOARD._hostid — использовать селектор узла сети панели мониторинга в качестве источника данных для узлов сети. Этот параметр не поддерживается при настройке виджета на панели мониторинга шаблона.
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: - *обязателен*

Примеры

Следующие примеры предназначены только для описания настройки объектов полей виджета панели мониторинга для виджета *Graph (classic)*. Для получения дополнительной информации о настройке панели мониторинга смотрите dashboard.create.

Настройка виджета *Graph (classic)*

Настройте виджет *Graph (classic)*, отображающий 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 для соответствующих объектов полей виджета панели мониторинга позволяют настроить виджет *Прототип графика* в методах `dashboard.create` и `dashboard.update`.

Attention:
Свойства `fields` виджета не проверяются при создании или обновлении панели мониторинга. Это позволяет пользователям изменять встроенные виджеты и создавать пользовательские виджеты, но также создает риск некорректного создания или обновления виджетов. Чтобы обеспечить успешное создание или обновление виджета *Прототип графика*, ознакомьтесь с поведением параметров, описанным в таблицах ниже.

Параметры

Для виджета *Прототип графика* 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.
Источник	0	source_type	2 - *(по умолчанию)* Прототип графика; 3 - Простой прототип графика.
Прототип графика	7	graphid.0	ID прототипа графика. Поведение параметра: - *обязателен*, если для *Источника* установлено значение "Прототип графика"
Прототип элемента данных	5	itemid.0	ID прототипа элемента данных. Поведение параметра: - *обязателен*, если для *Источника* установлено значение "Простой прототип графика"
Период времени	1	time_period._referenc	`DASHBOARD._timeperiod` - использовать селектор периода времени панели как источник данных; `ABCDE._timeperiod` - использовать совместимый виджет (с параметром *Reference*, установленным в "ABCDE") как источник 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 (`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`
Показывать легенду	0	show_legend	0 - Отключено; 1 - *(по умолчанию)* Включено.

Параметр	type	name	value
Переопределить узел сети	1	override_hostid._reference	ABCDE._hostid - использовать совместимый виджет (с параметром *Reference*, установленным в "ABCDE") как источник данных для узлов сети; DASHBOARD._hostid - использовать селектор узла сети панели как источник данных для узлов сети. Этот параметр не поддерживается при настройке виджета на панели шаблона.
Столбцы	0	columns	Допустимые значения: от 1 до 24. По умолчанию: 2.
Строки	0	rows	Допустимые значения: от 1 до 16. По умолчанию: 1.
Reference	1	reference	Любое строковое значение, состоящее из 5 символов (например, ABCDE или JBPNL). Это значение должно быть уникальным в пределах панели, к которой относится виджет. Поведение параметра: - *обязателен*

Примеры

Следующие примеры предназначены только для описания настройки объектов полей виджета панели мониторинга для виджета *Прототип графика*. Для получения дополнительной информации о настройке панели мониторинга смотрите dashboard.create.

Настройка виджета *Прототип графика*

Настройте виджет *Прототип графика*, 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. Ячеистая диаграмма

Описание

Эти параметры и возможные значения свойств для соответствующих объектов полей виджета позволяют настраивать виджет *Honeycomb* в методах `dashboard.create` и `dashboard.update`.

Attention:
Свойства `fields` виджета не проверяются при создании или обновлении панели. Это позволяет пользователям изменять встроенные виджеты и создавать пользовательские виджеты, но также создает риск некорректного создания или обновления виджетов. Чтобы обеспечить успешное создание или обновление виджета *Honeycomb*, ознакомьтесь с поведением параметров, описанным в таблицах ниже.

Параметры

Для виджета *Honeycomb* 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.
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, к которой принадлежат 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` — 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	evaltype_host	0 - *(по умолчанию)* И/Или; 2 - Или. Этот параметр не поддерживается при настройке виджета на панели шаблона.

Параметр	type	name	value
Имя тега	1	host_tags.0.tag	Любое строковое значение. Примечание: Число в имени свойства указывает порядок тега в списке вычисления тегов. Поведение параметра: - *обязательно* 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: Число в имени свойства указывает 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: Число в имени свойства указывает порядок тега в списке вычисления тегов. Поведение параметра: - *обязательно* при настройке *Тегов узлов сети* Этот параметр не поддерживается при настройке виджета на панели шаблона.
Шаблоны элементов данных	1	items.0	Имя или шаблон элемента данных. Примечание: Чтобы настроить несколько шаблонов элементов данных, создайте объект поля виджета панели с увеличивающимся номером в имени свойства для каждого шаблона элемента данных. Поведение параметра: - *обязательно*
Теги элементов данных			
Тип вычисления	0	evaltype_item	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: - *обязательно* при настройке *Тегов элементов данных*

Параметр	type	name	value
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: - *обязательно* 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: - *обязательно* при настройке *Тегов 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 с увеличивающимся номером в имени свойства для каждого значения. По умолчанию: 1, 2.
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: - *обязательно*

Advanced configuration

The following advanced configuration parameters are supported for the *Honeycomb* widget.

Note:
The number in the *Thresholds* property name (e.g. thresholds.0.color) references the threshold place in a list, sorted in ascending order. However, if thresholds are configured in a different order, the values will be sorted in ascending order after updating widget configuration in Zabbix frontend (e.g. `"thresholds.0.threshold":"5"` → `"thresholds.0.threshold":"1"`; `"thresholds.1.threshold":"1"` → `"thresholds.1.threshold": "5"`).

Parameter	type	name	value
Primary label			
Type	0	primary_label_type	0 - *(default)* Text; 1 - Value.

Parameter	type	name	value
Text	1	primary_label	Any string value, including macros. Supported macros: {HOST.*}, {ITEM.*}, {INVENTORY.*}, user macros. Default: {HOST.NAME} Parameter behavior: - *supported* if *Type* is set to ”Text”
Decimal places	0	primary_label_decimal_places	Possible values range from 0-6. Default: 2. Parameter behavior: - *supported* if *Type* is set to ”Value”
Size (type)	0	primary_label_size_type	0 - *(default)* Auto; 1 - Custom.
Size	0	primary_label_size	Possible values range from 1-100. Default: 20. Parameter behavior: - *supported* if *Size* (type) is set to ”Custom”
Bold	0	primary_label_bold	0 - *(default)* Disabled; 1 - Enabled.
Color	1	primary_label_color	Hexadecimal color code (e.g. `FF0000`). Default: based on `theme` of Settings object and User object: `1F2C33` for ”blue-theme” or ”hc-light”; `EEEEEE` for ”dark-theme” or ”hc-dark”.
Units (checkbox)	0	primary_label_units_show	0 - Disabled; 1 - *(default)* Enabled. Parameter behavior: - *supported* if *Type* is set to ”Value”
Units (value)	1	primary_label_units	Any string value. "" (empty) Parameter behavior: - *supported* if *Type* is set to ”Value” and *Units* (checkbox) is set to ”Enabled”
Position	0	primary_label_units_pos	0 - Before value; 1 - *(default)* After value. Parameter behavior: - *supported* if *Type* is set to ”Value” and *Units* (checkbox) is set to ”Enabled” This parameter is ignored if set to one of the following time-related units: unixtime, uptime, s.
Secondary label			
Type	0	secondary_label_type	0 - Text; 1 - *(default)* Value.
Text	1	secondary_label	Any string value, including macros. Supported macros: {HOST.*}, {ITEM.*}, {INVENTORY.*}, user macros. Default: {{ITEM.LASTVALUE}.fmtnum(2)} Parameter behavior: - *supported* if *Type* is set to ”Text”

Parameter	type	name	value
Decimal places	0	secondary_label_decimal_places	Possible values range from 0-6. Default: 2. Parameter behavior: - *supported* if *Type* is set to ”Value”
Size (type)	0	secondary_label_size_type	0 - *(default)* Auto; 1 - Custom.
Size	0	secondary_label_size	Possible values range from 1-100. Default: 30. Parameter behavior: - *supported* if *Size* (type) is set to ”Custom”
Bold	0	secondary_label_bold	0 - Disabled; 1 - *(default)* Enabled.
Color	1	secondary_label_color	Hexadecimal color code (e.g. `FF0000`). Default: based on `theme` of Settings object and User object: `1F2C33` for ”blue-theme” or ”hc-light”; `EEEEEE` for ”dark-theme” or ”hc-dark”.
Units (checkbox)	0	secondary_label_units_show	0 - Disabled; 1 - *(default)* Enabled. Parameter behavior: - *supported* if *Type* is set to ”Value”
Units (value)	1	secondary_label_units	Any string value. "" (empty) Parameter behavior: - *supported* if *Type* is set to ”Value” and *Units* (checkbox) is set to ”Enabled”
Position	0	secondary_label_position	0 - Before value; 1 - *(default)* After value. Parameter behavior: - *supported* if *Type* is set to ”Value” and *Units* (checkbox) is set to ”Enabled” This parameter is ignored if set to one of the following time-related units: unixtime, uptime, s.
Background color			
Background color	1	bg_color	Hexadecimal color code (e.g. `FF0000`). Default: based on `theme` of Settings object and User object: `D9E7ED` for ”blue-theme”; `3D5059` for ”dark-theme”; `AAD7E9` for ”hc-light”; `335463` for ”hc-dark”.
Thresholds			
Color interpolation	0	interpolation	0 - Disabled; 1 - *(default)* Enabled.
Color	1	thresholds.0.color	Hexadecimal color code (e.g. `FF0000`).
Threshold	1	thresholds.0.threshold	Any numeric value. Suffixes (e.g. ”1d”, ”2w”, ”4K”, ”8G”) are supported.

Examples

The following examples aim to only describe the configuration of the dashboard widget field objects for the *Honeycomb* widget. For more information on configuring a dashboard, see `dashboard.create`.

Настройка виджета *Honeycomb*

Настройте виджет *Honeycomb*, отображающий использование процессов сервера Zabbix. Кроме того, измените основную метку ячеек honeycomb и визуально донастройте виджет с помощью порогов.

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 для соответствующих объектов полей виджета панели мониторинга позволяют настраивать виджет *Доступность узла сети* в методах `dashboard.create` и `dashboard.update`.

Параметры

Для виджета *Доступность узла сети* 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.
Группы узлов сети	2	groupids.0	ID группы узлов сети. Примечание: Чтобы настроить несколько групп узлов сети, создайте объект поля виджета панели мониторинга для каждой группы узлов сети с увеличивающимся номером в имени свойства. Этот параметр не поддерживается при настройке виджета на панели мониторинга шаблона.
Группы узлов сети (виджет)	1	groupids._reference	Вместо ID группы узлов сети: `ABCDE._hostgroupids` — укажите 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.
Тип интерфейса	0	interface_type.0	0 - Нет; 1 - Агент Zabbix (пассивные проверки); 2 - SNMP; 3 - IPMI; 4 - JMX; 5 - Агент Zabbix (активные проверки). По умолчанию: 1, 2, 3, 4, 5 (все включены). Примечание: Чтобы настроить несколько значений, создайте объект поля виджета панели мониторинга для каждого значения с увеличивающимся номером в имени свойства.
Макет	0	layout	0 - *(по умолчанию)* Горизонтальный; 1 - Вертикальный.
Показывать узлы сети на 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 *Доступность узла сети*. Для получения дополнительной информации о настройке панели мониторинга смотрите `dashboard.create`.

Настройка виджета *Доступность узла сети*

Настройте виджет *Доступность узла сети*, который отображает информацию о доступности (в вертикальном макете) для узлов сети из группы узлов сети "4" с настроенными интерфейсами "Zabbix агент" 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 для соответствующих объектов полей виджета позволяют настраивать виджет *Карточка узла сети* в методах `dashboard.create` и `dashboard.update`.

Attention:
Свойства `fields` виджета не проверяются при создании или обновлении панели. Это позволяет пользователям изменять встроенные виджеты и создавать пользовательские виджеты, но также создает риск некорректного создания или обновления виджетов. Чтобы обеспечить успешное создание или обновление виджета *Карточка узла сети*, ознакомьтесь с поведением параметров, описанным 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.

Параметр	type	name	value
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 для соответствующих объектов полей виджета позволяют настраивать виджет *Навигатор узлов сети* в методах `dashboard.create` и `dashboard.update`.

Attention:
Свойства `fields` виджета не проверяются при создании или обновлении панели. Это позволяет пользователям изменять встроенные виджеты и создавать пользовательские виджеты, но также создает риск некорректного создания или обновления виджетов. Чтобы обеспечить успешное создание или обновление виджета *Навигатор узлов сети*, ознакомьтесь с поведением параметров, описанным в таблицах ниже.

Параметры

Для виджета *Навигатор узлов сети* 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.
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.

Параметр	type	name	value
Шаблоны узлов сети	1	hosts.0	Имя или шаблон узла сети. Примечание: Чтобы настроить несколько шаблонов узлов сети, создайте объект поля виджета панели с увеличивающимся номером в имени свойства для каждого шаблона узла сети. При использовании нескольких шаблонов узлов сети параметр *Группы узлов сети* либо не должен быть настроен вовсе, либо должен быть настроен как минимум с одной группой узлов сети, к которой принадлежат узлы сети, соответствующие заданным шаблонам узлов сети. Этот параметр не поддерживается при настройке виджета на панели шаблона.
Статус узла сети	0	status	-1 - *(по умолчанию)* Любой; 0 - Включен; 1 - Отключен. Этот параметр не поддерживается при настройке виджета на панели шаблона.
Теги узла 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 узла 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 на порядок тега в списке вычисления тегов. Поведение параметра: - *обязательно* при настройке *Тегов узла 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 узла сети* Этот параметр не поддерживается при настройке виджета на панели шаблона.

Параметр	type	name	value
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 с увеличивающимся номером в имени свойства для каждого значения.
Показывать узлы сети на обслуживании	0	maintenance	0 - *(по умолчанию)* Отключено; 1 - Включено.
Показывать проблемы	0	show_problems	0 - Все; 1 - *(по умолчанию)* Неподавленные; 2 - Нет.
Группировать по			
Атрибут	0	group_by.0.attribute	0 - Группа узлов сети; 1 - Значение тега; 2 - Важность. Примечание: Номер в имени свойства указывает на порядок атрибута в списке атрибутов группировки. Поведение параметра: - *обязательно* при настройке *Группировать по*
Значение	1	group_by.0.tag_name	Любое строковое значение. Примечание: Номер в имени свойства указывает на атрибут группировки, заданный в параметре *Атрибут*. Поведение параметра: - *обязательно* при настройке *Группировать по*, если *Атрибут* установлен в значение ”Значение тега”
Лимит узлов сети	0	show_lines	Допустимые значения: от 1 до 9999. По умолчанию: 100. Этот параметр не поддерживается при настройке виджета на панели шаблона.
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: - *обязательно*

Примеры

Следующие примеры предназначены только для описания конфигурации объектов полей виджета панели мониторинга для виджета *Навигатор узлов сети*. Для получения дополнительной информации о настройке панели мониторинга см. `dashboard.create`.

Настройка виджета *Host navigator*

Настройте виджет *Host navigator*, который отображает узлы сети, сгруппированные 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 позволяют настраивать виджет *Карточка элемента данных* в методах `dashboard.create` и `dashboard.update`.

Attention:
Свойства `fields` виджета не проверяются при создании или обновлении панели. Это позволяет пользователям изменять встроенные виджеты и создавать пользовательские виджеты, но также вносит риск некорректного создания или обновления виджетов. Чтобы обеспечить успешное создание или обновление виджета *Карточка элемента данных*, ознакомьтесь с поведением параметров, описанным 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

Параметр	type	name	value
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.
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 *Show* установлено значение "Latest data".

Параметр	type	name	value
Ширина	0	sparkline.width	Возможные значения находятся в диапазоне 0-10. По умолчанию: 1.
Заполнение	0	sparkline.fill	Возможные значения находятся в диапазоне 0-10. 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 как источник данных; `ABCDE._timeperiod` — использовать совместимый виджет (с параметром `reference`, равным `ABCDE`) как источник данных. По умолчанию: "" (пусто) В качестве альтернативы можно задать период времени только в параметрах *От* и *До*.

Параметр	type	name	value
От	1	sparkline.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.). По умолчанию: now-1h. Поведение параметра: - *поддерживается*, если *Период времени* не задан - *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 времени (`now`, `now/d`, `now/w-1w` и т. д.). По умолчанию: now. Поведение параметра: - *поддерживается*, если *Период времени* не задан - *обязателен*, если задан `time_period.to`
Данные истории	0	sparkline.history	0 — *(по умолчанию)* Авто; 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 элементов данных

Описание

Эти параметры и возможные значения свойств для соответствующих объектов полей виджета позволяют настраивать виджет *История элемента данных* в методах dashboard.create и dashboard.update.

Attention:
Свойства `fields` виджета не проверяются при создании или обновлении панели. Это позволяет пользователям изменять встроенные виджеты и создавать пользовательские виджеты, но также создает риск некорректного создания или обновления виджетов. Чтобы обеспечить успешное создание или обновление виджета *История элемента данных*, ознакомьтесь с поведением параметров, описанным в таблицах ниже.

Параметры

Для виджета *История элементов 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.
Макет	0	layout	0 - *(по умолчанию)* Горизонтальный; 1 - Вертикальный.
Столбцы (см. ниже)			
Показывать строки	0	show_lines	Возможные значения: от 1 до 100. По умолчанию: 25.
Переопределить узел сети	1	override_hostid._reference	`ABCDE._hostid` - установить совместимый виджет (с параметром *Reference*, установленным в "`ABCDE`") в качестве источника данных для узлов сети; `DASHBOARD._hostid` - установить селектор узла сети панели как источник данных для узлов сети. Этот параметр не поддерживается при настройке виджета на панели шаблона.
Расширенная конфигурация (см. ниже)			
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*

Столбцы

Столбцы имеют общие параметры и дополнительные параметры в зависимости от конфигурации параметра *Элемент данных*.

Note:
Для всех параметров, относящихся к столбцам, число в имени свойства (например, columns.0.name) указывает на столбец, для которого настраивается этот параметр.

Следующие параметры поддерживаются для всех столбцов.

Параметр	type	name	value
Имя	1	columns.0.name	Любое строковое значение. Поведение параметра: *- обязательный*

Параметр	type	name	value
Элемент данных	4	columns.0.itemid	ID элемента данных. При настройке виджета на панели шаблона следует указывать только элементы данных, настроенные в шаблоне. Поведение параметра: - *обязательный*
Базовый цвет	1	columns.0.base_color	Шестнадцатеричный код цвета (например, FF0000). По умолчанию: "" (пусто).

Следующие параметры столбца поддерживаются, если настроенный *элемент данных* имеет числовой тип.

Параметр	type	name	value
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 *Отображение* установлено в ”Полоса” или ”Индикаторы”
Макс	1	columns.0.max	Любое числовое значение. Поведение параметра: - *поддерживается*, если *Отображение* установлено в ”Полоса” или ”Индикаторы”
Пороговые значения			
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 столбца поддерживаются, если настроенный *Элемент данных* имеет тип символ, текст или журнал.

Параметр	type	name	value
Подсветки			
Подсветка	1	columns.0.highlights.color	Шестнадцатеричный код цвета (например, FF0000).
Порог	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 находятся в диапазоне 1-500. По умолчанию: 100. Поведение параметра: - *поддерживается*, если *Отображение* установлено в ”Одна строка”
Использовать моноширинный шрифт	0	columns.0.monospace_font	0 - *(по умолчанию)* Использовать шрифт по умолчанию; 1 - Использовать моноширинный шрифт.

Параметр	type	name	value
Отображать локальное время	0	columns.0.local_time	0 - *(по умолчанию)* Отображать временную метку; 1 - Отображать локальное время. Поведение параметра: - *поддерживается*, если *Элемент данных* установлен как элемент данных типа журнал, а *Показывать временную метку* установлено в ”Включено”

Следующие параметры столбца поддерживаются, если настроенный *Элемент данных* имеет двоичный тип.

Параметр	type	name	value
Показывать миниатюру	1	columns.0.show_thumbnail	0 - *(по умолчанию)* Отключено; 1 - Включено.

Расширенная настройка

Следующие параметры расширенной настройки поддерживаются для виджета *История элементов данных*.

Параметр	type	name	value
Новые значения	0	sortorder	0 - *(по умолчанию)* Сверху; 1 - Снизу.
Показывать метку времени	0	show_timestamp	0 - *(по умолчанию)* Отключено; 1 - Включено.
Показывать заголовок столбца	0	show_column_header	0 - Выкл.; 1 - Горизонтально; 2 - *(по умолчанию)* Вертикально.
Период времени	1	time_period._reference	`DASHBOARD._timeperiod` - использовать селектор периода времени панели как источник данных; `ABCDE._timeperiod` - использовать совместимый виджет (с параметром *Reference*, установленным в ”`ABCDE`”) как источник 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 (`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`

Примеры

Следующие примеры предназначены только для описания настройки объектов полей виджета панели мониторинга для виджета *История элемента данных*. Для получения дополнительной информации о настройке панели мониторинга смотрите `dashboard.create`.

Настройка виджета *История элемента данных*

Настройте виджет *История элемента данных*, отображающий 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 для соответствующих объектов полей виджета позволяют настраивать виджет *Навигатор элементов данных* в методах `dashboard.create` и `dashboard.update`.

Attention:
Свойства `fields` виджета не проверяются при создании или обновлении панели. Это позволяет пользователям изменять встроенные виджеты и создавать пользовательские виджеты, но также создает риск некорректного создания или обновления виджетов. Чтобы обеспечить успешное создание или обновление виджета *Навигатор элементов данных*, ознакомьтесь с поведением параметров, описанным в таблицах ниже.

Параметры

Для виджета *Навигатор элементов 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.
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 *Ссылка*, 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, к 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.hostid` — укажите селектор узла сети панели в качестве источника данных для узлов сети; `ABCDE._hostid` — укажите совместимый виджет (с параметром *Ссылка*, 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.
Шаблоны элементов данных	1	items.0	Имя или шаблон элемента данных. Примечание: Чтобы настроить несколько шаблонов элементов данных, создайте объект поля виджета панели с увеличивающимся номером в имени свойства для каждого шаблона элемента данных.
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: Номер в имени свойства указывает на порядок тега в списке вычисления тегов. Поведение параметра: - *обязательно* 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: Номер в имени свойства указывает на порядок тега в списке вычисления тегов. Поведение параметра: - *обязательно* при настройке *Тегов элементов данных*
Значение тега	1	item_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 элементов данных*

Параметр	type	name	value
Состояние	0	state	-1 - *(по умолчанию)* Все; 0 - Нормальное; 1 - Не поддерживается.
Показывать проблемы	0	show_problems	0 - Все; 1 - *(по умолчанию)* Неподавленные; 2 - Нет.
Группировать по			
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: Номер в имени свойства указывает на порядок атрибута в списке атрибутов группировки. Поведение параметра: - *обязательно* при настройке *Группировать по*
Значение	1	group_by.0.tag_name	Любое строковое значение. Примечание: Номер в имени свойства указывает на атрибут группировки, заданный в параметре *Атрибут*. Поведение параметра: - *обязательно* при настройке *Группировать по*, если *Атрибут* установлен в значение ”Значение тега узла сети” или ”Значение тега элемента данных”
Лимит элементов данных	0	show_lines	Допустимые значения: 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 панели, к которой принадлежит виджет. Поведение параметра: - *обязательно*

Примеры

Следующие примеры предназначены только для описания настройки объектов полей виджета панели мониторинга *Навигатор элементов данных*. Для получения дополнительной информации о настройке панели мониторинга см. `dashboard.create`.

Настройка виджета *Навигатор элементов данных*

Настройте виджет *Навигатор элементов данных*, который отображает до 1000 элементов данных, сгруппированных по их 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 позволяют настраивать виджет *Значение элемента данных* в методах `dashboard.create` и `dashboard.update`.

Attention:
Свойства `fields` виджета не проверяются при создании или обновлении панели. Это позволяет пользователям изменять встроенные виджеты и создавать пользовательские виджеты, но также создает риск некорректного создания или обновления виджетов. Чтобы обеспечить успешное создание или обновление виджета *Значение элемента данных*, ознакомьтесь с поведением параметров, описанным в таблицах ниже.

Параметры

Для виджета *Значение 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: - *обязательно*, если *Элемент данных (Виджет)* не задан
Элемент данных (Виджет)	1	itemid._reference	Вместо ID элемента данных: `ABCDE._itemid` - задаёт совместимый виджет (с его параметром *Ссылка*, установленным в "ABCDE") в качестве источника данных для элементов данных. Поведение параметра: - *обязательно*, если *Элемент данных* не задан
Показывать	0	show.0	1 - Описание; 2 - Значение; 3 - Время; 4 - Индикатор изменения; 5 - Спарклайн. По умолчанию: 1, 2, 3, 4. Примечание: Чтобы настроить несколько значений, создайте объект поля виджета панели мониторинга для каждого значения с увеличивающимся номером в имени свойства.

Параметр	type	name	value
Переопределить узел сети	1	override_hostid._reference	ABCDE._hostid - задаёт совместимый виджет (с его параметром *Ссылка*, установленным в "ABCDE") в качестве источника данных для узлов сети; DASHBOARD._hostid - задаёт селектор узла сети панели мониторинга в качестве источника данных для узлов сети. Этот параметр не поддерживается при настройке виджета на панели мониторинга шаблона.

Расширенная настройка

Для виджета *Значение элемента данных* поддерживаются следующие дополнительные параметры настройки.

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 в веб-интерфейсе Zabbix (например, "thresholds.0.threshold":"5" → "thresholds.0.threshold":"1"; "thresholds.1.threshold":"1" → "thresholds.1.threshold": "5").

Параметр	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 — использовать селектор периода времени панели как источник данных; ABCDE._timeperiod — использовать совместимый виджет (с параметром reference, равным ABCDE) как источник данных. По умолчанию: DASHBOARD._timeperiod В качестве альтернативы можно задать период времени только в параметрах *От* и *До*. Поведение параметра: - *поддерживается*, если *Функция агрегации* установлена в "min", "max", "avg", "count", "sum", "first", "last"
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 и *Функция агрегации* установлена в "min", "max", "avg", "count", "sum", "first", "last" - *обязателен*, если задан time_period.to

Параметр	type	name	value
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 и *Функция агрегации* установлена в ”min”, ”max”, ”avg”, ”count”, ”sum”, ”first”, ”last” - *обязателен*, если задан `time_period.from`
Данные истории	0	history	0 - *(по умолчанию)* Авто; 1 - История; 2 - Тренды.

Описание

Следующие расширенные параметры конфигурации поддерживаются, если для *Show* установлено значение ”Description”.

Параметр	type	name	value
Description	1	description	Любое строковое значение, включая макросы. Поддерживаемые макросы: {HOST.*}, {ITEM.*}, {INVENTORY.*}, пользовательские макросы. По умолчанию: {ITEM.NAME}.
Horizontal position	0	desc_h_pos	0 - Слева; 1 - *(по умолчанию)* По центру; 2 - Справа. Два или более элемента (Description, Value, Time) не могут иметь одинаковые значения *Horizontal position* и *Vertical position*.
Vertical position	0	desc_v_pos	0 - Сверху; 1 - Посередине; 2 - *(по умолчанию)* Снизу. Два или более элемента (Description, Value, Time) не могут иметь одинаковые значения *Horizontal position* и *Vertical position*.
Size	0	desc_size	Допустимые значения находятся в диапазоне 1-100. По умолчанию: 15.
Bold	0	desc_bold	0 - *(по умолчанию)* Отключено; 1 - Включено.
Color	1	desc_color	Шестнадцатеричный код цвета (например, FF0000). По умолчанию: "" (пусто).

Значение

Следующие расширенные параметры конфигурации поддерживаются, если для *Показывать* установлено значение ”Значение”.

Параметр	type	name	value
Десятичные знаки			
Десятичные знаки	0	decimal_places	Допустимые значения: 1-10. По умолчанию: 2.
Размер	0	decimal_size	Допустимые значения: 1-100. По умолчанию: 35.
Положение			

Параметр	type	name	value
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 *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 *Горизонтальное положение* и *Вертикальное положение*.
Размер	0	value_size	Допустимые значения: 1-100. По умолчанию: 45.
Жирный	0	value_bold	0 - Отключено; 1 - *(по умолчанию)* Включено.
Цвет	1	value_color	Шестнадцатеричный код цвета (например, FF0000). По умолчанию: "" (пусто).
Единицы			
Единицы (флажок)	0	units_show	0 - Отключено; 1 - *(по умолчанию)* Включено.
Единицы (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: 1-100. По умолчанию: 35.
Жирный	0	units_bold	0 - Отключено; 1 - *(по умолчанию)* Включено.
Цвет	1	units_color	Шестнадцатеричный код цвета (например, FF0000). По умолчанию: "" (пусто).

Время

Следующие дополнительные параметры конфигурации поддерживаются, если для *Show* установлено значение ”Time”.

Параметр	type	name	value
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 *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 *Горизонтальное положение* и *Вертикальное положение*.

Параметр	type	name	value
Размер	0	time_size	Допустимые значения находятся в диапазоне 1-100. По умолчанию: 15.
Жирный	0	time_bold	0 - *(по умолчанию)* Отключено; 1 - Включено.
Цвет	1	time_color	Шестнадцатеричный код цвета (например, FF0000). По умолчанию: "" (пусто).

Индикатор изменения

Следующие дополнительные параметры конфигурации поддерживаются, если для *Show* установлено значение «Индикатор изменения».

Параметр	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

Следующие дополнительные параметры конфигурации поддерживаются, если для *Show* установлено значение ”Sparkline”.

Параметр	type	name	value
Ширина	0	sparkline.width	Возможные значения находятся в диапазоне 0-10. По умолчанию: 1.
Заполнение	0	sparkline.fill	Возможные значения находятся в диапазоне 0-10. По умолчанию: 3.
Цвет	1	sparkline.color	Шестнадцатеричный код цвета (например, FF0000). По умолчанию: 42A5F5.
Период времени	1	sparkline.time_period.from	DASHBOARD._timeperiod — использовать селектор периода времени панели как источник данных; ABCDE._timeperiod — использовать совместимый виджет (с параметром reference, равным ABCDE) как источник данных. По умолчанию: DASHBOARD._timeperiod В качестве альтернативы 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 времени (now, now/d, now/w-1w и т. д.). Поведение параметра: - *обязателен*, если задан sparkline.time_period.to

Параметр	type	name	value
До	1	sparkline.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: - *обязателен*, если задан `sparkline.time_period.from`
Исторические данные	0	sparkline.history	0 — *(по умолчанию)* Авто; 1 — История; 2 — Тренды.

Примеры

Следующие примеры предназначены только для описания настройки объектов полей виджета панели мониторинга для виджета *Значение элемента данных*. Для получения дополнительной информации о настройке панели мониторинга смотрите `dashboard.create`.

Настройка виджета *Значение элемента данных*

Настройте виджет *Значение элемента данных*, который отображает значение элемента данных для элемента данных "42266" (доступность Zabbix агент). Кроме того, выполните визуальную тонкую настройку виджета с помощью нескольких дополнительных параметров, включая динамический цвет фона, который изменяется в зависимости от состояния доступности Zabbix агент.

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 для соответствующих объектов полей виджета позволяют настроить виджет *Карта* в методах dashboard.create и dashboard.update.

Attention:
Свойства fields виджета не проверяются при создании или обновлении панели мониторинга. Это позволяет пользователям изменять встроенные виджеты и создавать пользовательские виджеты, но также вносит риск некорректного создания или обновления виджетов. Чтобы обеспечить успешное создание или обновление виджета *Карта*, обратитесь к поведению параметров, описанному в таблицах ниже.

Параметры

Для виджета *Карта* 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.
Карта	8	sysmapid.0	ID карты. Поведение параметра: - *обязателен*, если *Карта (виджет)* не задана
Карта (виджет)	1	sysmapid._reference	ABCDE._mapid — задает виджет *Дерево навигации по картам* (с параметром *Ссылка*, установленным в "ABCDE") в качестве источника данных для карт. Поведение параметра: - *обязателен*, если *Карта* не задана
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: - *обязателен*

Примеры

Следующие примеры предназначены только для описания конфигурации объектов полей виджета панели мониторинга для виджета *Карта*. Для получения дополнительной информации о настройке панели мониторинга см. dashboard.create.

Настройка виджета *Карта*

Настройте виджет *Карта*, отображающий карту "1".

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": 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
}
```

Ответ:

```
{
    "jsonrpc": "2.0",
    "result": {
        "dashboardids": [
            "3"
        ]
    },
    "id": 1
}
```

Настройка связанного виджета *Карта*

Настройте виджет *Карта*, связанный с виджетом *Дерево навигации по картам*.

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 картам

Описание

Эти параметры и возможные значения свойств для соответствующих объектов полей виджета позволяют настраивать виджет *Дерево навигации по картам* в методах `dashboard.create` и `dashboard.update`.

Attention:
Свойства `fields` виджета не проверяются при создании или обновлении панели. Это позволяет пользователям изменять встроенные виджеты и создавать пользовательские виджеты, но также создает риск некорректного создания или обновления виджетов. Чтобы обеспечить успешное создание или обновление виджета *Дерево навигации по картам*, ознакомьтесь с поведением параметров, описанным в таблицах ниже.

Параметры

Следующие параметры поддерживаются для виджета *Дерево навигации по картам*.

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.
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 панели, к которой принадлежит виджет. Поведение параметра: - *обязательно*

Следующие параметры поддерживаются для настройки элементов дерева навигации по картам.

Параметр	type	name	value
Имя	1	navtree.1.name	Любое строковое значение. Примечание: Число в имени свойства задаёт 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	Позиция элемента в дереве навигации по картам. Примечание: Число в имени свойства указывает на номер элемента. Значение свойства указывает на позицию элемента в дереве навигации по картам. Позиция родительского элемента определяется в пределах всего дерева навигации по картам. Позиция дочернего элемента определяется в пределах родительского элемента.

Примеры

Следующие примеры предназначены только для описания настройки объектов полей виджета панели мониторинга для виджета *Дерево навигации карт*. Для получения дополнительной информации о настройке панели мониторинга см. `dashboard.create`.

Настройка виджета *Дерево навигации по картам*

Настройте виджет *Дерево навигации по картам*, который отображает следующее дерево навигации по картам:

- Элемент A
  - Элемент A1
  - Элемент A2
- Элемент B
  - Элемент B1 (содержит связанную карту "1", которую можно отобразить в связанном виджете *Карта*)
  - 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 для соответствующих объектов полей виджета позволяют настраивать виджет *Круговая диаграмма* в методах `dashboard.create` и `dashboard.update`.

Attention:
Свойства `fields` виджета не проверяются при создании или обновлении панели. Это позволяет пользователям изменять встроенные виджеты и создавать пользовательские виджеты, но также создает риск некорректного создания или обновления виджетов. Чтобы обеспечить успешное создание или обновление виджета *Круговая диаграмма*, ознакомьтесь с поведением параметров, описанным в таблицах ниже.

Параметры

Для виджета *Круговая 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.

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	0	ds.0.dataset_type	0 - Список элементов данных; 1 - *(по умолчанию)* Шаблоны элементов данных.
Элементы данных	4	ds.0.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 увеличивающимся номером в имени свойства. Поведение параметра: - *обязателен*, если *Тип набора данных* установлен в значение "Список элементов данных" и *Элементы данных (Виджет)* не задан
Элементы данных (Виджет)	1	ds.0.itemids.0._reference	Вместо ID элемента данных: `ABCDE._itemid` — укажите 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 увеличивающимся номером в имени свойства. Поведение параметра: - *обязателен*, если *Тип набора данных* установлен в значение "Список элементов данных" и *Элементы данных* не заданы
Цвет	1	ds.0.color.0	Шестнадцатеричный код цвета (например, `FF0000`). Поведение параметра: - *обязателен*, если *Тип набора данных* установлен в значение "Список элементов данных"
Тип элемента данных	0	ds.0.type.0	0 - *(по умолчанию)* Обычный; 1 - Итого. Значение "Итого" можно установить только для одного элемента данных во всей диаграмме. Поведение параметра: - *поддерживается*, если *Тип набора данных* установлен в значение "Список элементов данных"
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.

Параметр	type	name	value
Шаблоны элементов данных	1	ds.0.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”
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. Возможные значения: от 0 до 11. По умолчанию: 0. Поведение параметра: - *поддерживается*, если *Тип набора данных* установлен в значение ”Шаблоны элементов данных” и *Цвет* не задан
Функция агрегации	0	ds.0.aggregate_function	1 - min; 2 - max; 3 - avg; 4 - count; 5 - sum; 6 - first; 7 - *(по умолчанию)* last.
Агрегация набора данных	0	ds.0.dataset_aggregation	0 - *(по умолчанию)* нет; 1 - min; 2 - max; 3 - avg; 4 - count; 5 - sum. Поведение параметра: - *поддерживается*, если *Тип элемента данных* установлен в значение ”Итого”
Метка набора данных	1	ds.0.data_set_label	Любое строковое значение. По умолчанию: "" (пусто).
Теги элементов данных			
Тип вычисления	0	ds.0.item_tags_evaltype	0 - *(по умолчанию)* And/Or; 2 - Or.
Имя тега	1	ds.0.item_tags.0.tag	Любое строковое значение. Примечание: Число в имени свойства указывает на порядок тега в списке вычисления тегов. Поведение параметра: - *обязателен*, если настраиваются *Теги элементов данных*

Параметр	type	name	value
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 на порядок тега в списке вычисления тегов. Поведение параметра: - *обязателен*, если настраиваются *Теги элементов данных*
Значение тега	1	ds.0.item_tags.0.value	Любое строковое значение. Примечание: Число в имени свойства указывает на порядок тега в списке вычисления тегов. Поведение параметра: - *обязателен*, если настраиваются *Теги элементов данных*

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.
Отрисовка	0	draw_type	0 - *(по умолчанию)* Круговая; 1 - Кольцевая.
Ширина	0	width	20 - 20% от радиуса; 30 - 30% от радиуса; 40 - 40% от радиуса; 50 - *(по умолчанию)* 50% от радиуса. Поведение параметра: - *поддерживается*, если для *Отрисовка* установлено значение ”Кольцевая”
Ширина обводки	0	stroke	Допустимые значения: от 0 до 10. По умолчанию: 0. Поведение параметра: - *поддерживается*, если для *Отрисовка* установлено значение ”Кольцевая”
Показывать суммарное значение	0	total_show	0 - *(по умолчанию)* Отключено; 1 - Включено. Поведение параметра: - *поддерживается*, если для *Отрисовка* установлено значение ”Кольцевая”
Размер	0	value_size_type	0 - *(по умолчанию)* Авто; 1 - Пользовательский. Поведение параметра: - *поддерживается*, если для *Показывать суммарное значение* установлено значение ”Включено”

Параметр	type	name	value
Размер (значение для пользовательского размера)	0	value_size	Допустимые значения: от 1 до 100. По умолчанию: 20. Поведение параметра: - *поддерживается*, если для *Показывать суммарное значение* установлено значение ”Включено”
Количество десятичных знаков	0	decimal_places	Допустимые значения: от 0 до 6. По умолчанию: 2. Поведение параметра: - *поддерживается*, если для *Показывать суммарное значение* установлено значение ”Включено”
Единицы измерения (флажок)	0	units_show	0 - *(по умолчанию)* Отключено; 1 - Включено. Поведение параметра: - *поддерживается*, если для *Показывать суммарное значение* установлено значение ”Включено”
Единицы измерения (значение)	1	units	Любое строковое значение. Поведение параметра: - *поддерживается*, если для *Единицы измерения* (флажок) установлено значение ”Включено”
Жирный	0	value_bold	0 - *(по умолчанию)* Отключено; 1 - Включено. Поведение параметра: - *поддерживается*, если для *Показывать суммарное значение* установлено значение ”Включено”
Цвет	1	value_color	Шестнадцатеричный код цвета (например, FF0000). Поведение параметра: - *поддерживается*, если для *Показывать суммарное значение* установлено значение ”Включено”
Промежуток между секторами	0	space	Допустимые значения: от 0 до 10. По умолчанию: 1.
Объединять сектора меньше N% (флажок)	0	merge	0 - *(по умолчанию)* Отключено; 1 - Включено.
Объединять сектора меньше N% (значение)	0	merge_percent	Допустимые значения: от 1 до 10. По умолчанию: 1. Поведение параметра: - *поддерживается*, если для *Объединять сектора меньше N%* (флажок) установлено значение ”Включено”
Объединять сектора меньше N% (цвет)	1	merge_color	Шестнадцатеричный код цвета (например, FF0000). Поведение параметра: - *поддерживается*, если для *Объединять сектора меньше N%* (флажок) установлено значение ”Включено”

Период времени

Для настройки *Периода времени* поддерживаются следующие параметры.

Параметр	type	name	value
Период времени	1	time_period._referende	DASHBOARD._timeperiod — использовать селектор периода времени панели мониторинга в качестве источника данных; ABCDE._timeperiod — использовать совместимый виджет (с параметром *Reference*, установленным в значение "ABCDE") в качестве источника данных. По умолчанию: DASHBOARD._timeperiod В качестве альтернативы 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: - *поддерживается*, если *Период времени* не задан - *обязателен*, 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: - *поддерживается*, если *Период времени* не задан - *обязателен*, если задан time_period.from

Условные обозначения

Для настройки *условных обозначений* поддерживаются следующие параметры.

Параметр	type	name	value
Показывать условные обозначения	0	legend	0 - Отключено; 1 - *(по умолчанию)* Включено.
Показывать значение	0	legend_value	0 - *(по умолчанию)* Отключено; 1 - Включено. Поведение параметра: - *поддерживается*, если для *Показывать условные обозначения* установлено значение "Включено"
Показывать функцию агрегации	0	legend_aggregation	0 - *(по умолчанию)* Отключено; 1 - Включено. Поведение параметра: - *поддерживается*, если для *Показывать условные обозначения* установлено значение "Включено"
Строки	0	legend_lines_mode	0 - *(по умолчанию)* Фиксированное; 1 - Переменное. Поведение параметра: - *поддерживается*, если для *Показывать условные обозначения* установлено значение "Включено"
Количество строк/ Максимальное количество строк	0	legend_lines	Допустимые значения: 1-10. По умолчанию: 1. Поведение параметра: - *поддерживается*, если для *Показывать условные обозначения* установлено значение "Включено"

Параметр	type	name	value
Количество столбцов	0	legend_columns	Допустимые значения: 1-4. По умолчанию: 4. Поведение параметра: - *поддерживается*, если для *Показывать условные обозначения* установлено значение "Включено", а для *Показывать значение* установлено значение "Отключено"

Примеры

Следующие примеры предназначены только для описания настройки объектов полей виджета панели мониторинга *Круговая диаграмма*. Для получения дополнительной информации о настройке панели мониторинга см. `dashboard.create`.

Настройка виджета *Круговая диаграмма*

Настройте виджет *Круговая диаграмма* следующим образом:

- 2 набора данных, всего для 9 элементов данных на 1 узле сети.
- Первый набор данных имеет тип "Список элементов данных" и состоит из 3 элементов данных, все они имеют тип "Обычный" и представлены разными цветами.
- Второй набор данных имеет тип "Шаблоны элементов данных", состоит из 6 элементов данных, для каждого элемента данных настроена агрегация, и он представлен пользовательским цветом.
- Второй набор данных также имеет пользовательскую метку набора данных.
- Данные на круговой диаграмме отображаются в виде кольцевой диаграммы 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 данных 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 Проблемные узлы сети

Описание

Эти параметры и возможные значения свойств для соответствующих объектов полей виджета позволяют настроить виджет *Узлы сети с проблемами* в методах `dashboard.create` и `dashboard.update`.

Attention:
Свойства `fields` виджета не проверяются при создании или обновлении панели мониторинга. Это позволяет пользователям изменять встроенные виджеты и создавать пользовательские виджеты, но также создает риск некорректного создания или обновления виджетов. Чтобы обеспечить успешное создание или обновление виджета *Узлы сети с проблемами*, ознакомьтесь с поведением параметров, описанным в таблицах ниже.

Параметры

Для виджета *Узлы сети с проблемами* 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.
Группы узлов сети	2	groupids.0	ID группы узлов сети. Примечание: Чтобы настроить несколько групп узлов сети, создайте объект поля виджета панели с номером, увеличивающимся 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 *Ссылка*, 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
Узлы сети	3	hostids.0	ID узла сети. Примечание: Чтобы настроить несколько узлов сети, создайте объект поля виджета панели для каждого узла сети с номером, увеличивающимся в имени свойства. Для нескольких узлов сети параметр *Группы узлов сети* либо не должен быть настроен вообще, либо должен быть настроен как минимум с одной группой узлов сети, к 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 *Ссылка*, установленным в "ABCDE") в качестве источника данных для узлов сети. Этот параметр не поддерживается при настройке виджета на панели шаблона.
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 с номером, увеличивающимся в имени свойства. 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	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 с номером, увеличивающимся 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: - *обязательно* при настройке *Тегов проблем*

Параметр	type	name	value
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: - *обязательно* 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: - *обязательно* при настройке *Тегов проблем*
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 панели шаблона.
Отображение проблем	0	ext_ack	0 - *(по умолчанию)* Все; 1 - Только неподтвержденные; 2 - Разделено.
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 настройки объектов полей виджета панели мониторинга для виджета *Узлы сети с проблемами*. Для получения дополнительной информации о настройке панели мониторинга смотрите `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 именем, включающим строку ”CPU”, и которые имеют следующие уровни важности: ”Warning”, ”Average”, ”High”, ”Disaster”.

Запрос:

```
{
    "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 для соответствующих объектов полей виджета панели мониторинга позволяют настраивать виджет *Проблемы* в методах `dashboard.create` и `dashboard.update`.

Attention:
Свойства `fields` виджета не проверяются при создании или обновлении панели мониторинга. Это позволяет пользователям изменять встроенные виджеты и создавать пользовательские виджеты, но также создает риск некорректного создания или обновления виджетов. Чтобы обеспечить успешное создание или обновление виджета *Проблемы*, пожалуйста, ознакомьтесь с поведением параметров, описанным в таблицах ниже.

Параметры

Для виджета *Проблемы* 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.
Показывать	0	show	1 - *(по умолчанию)* Недавние проблемы; 2 - История; 3 - Проблемы.
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.

Параметр	type	name	value
Исключить группы узлов сети	2	exclude_groupids.0	ID группы узлов сети. Примечание: Чтобы исключить 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.
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 вообще, либо должен быть настроен как минимум с одной группой узлов сети, к которой принадлежат 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` — 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.
Проблема	1	problem	Имя события проблемы (без учета регистра, полное имя или его часть).
Важность	0	severities.0	0 - Не классифицировано; 1 - Информация; 2 - Предупреждение; 3 - Средняя; 4 - Высокая; 5 - Катастрофическая. По умолчанию: пусто (включены все). Примечание: Чтобы настроить несколько значений, создайте объект поля виджета панели с увеличивающимся номером в имени свойства для каждого значения.
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 на порядок тега в списке вычисления тегов. Поведение параметра: - *обязателен* при настройке *Тегов проблем*

Параметр	type	name	value
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 на 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 на 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 *Показывать теги* установлено в ”1”, ”2” или ”3”
Приоритет отображения тегов	1	tag_priority	Список тегов, разделенных запятыми. Поведение параметра: - *поддерживается*, если *Показывать теги* установлено в ”1”, ”2” или ”3”
Показывать оперативные данные	0	show_opdata	0 - *(по умолчанию)* Нет; 1 - Отдельно; 2 - С именем проблемы.
Показывать подавленные проблемы	0	show_suppressed	0 - *(по умолчанию)* Отключено; 1 - Включено.
Статус подтверждения	0	acknowledgement_status	0 - *(по умолчанию)* все; 1 - Неподтвержденные; 2 - Подтвержденные.
Мной	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 быть установлен в ”Отключено”. Значения ”Узел сети (по возрастанию)” и ”Узел сети (по убыванию)” не поддерживаются при настройке виджета на панели шаблона.

Параметр	type	name	value
Показывать временную шкалу	0	show_timeline	0 - Отключено; 1 - *(по умолчанию)* Включено. Поведение параметра: - *поддерживается*, если *Сортировать записи по* установлено в ”Время (по убыванию)” или ”Время (по возрастанию)”
Подсвечивать всю строку	0	highlight_row	0 - *(по умолчанию)* Отключено; 1 - Включено. Этот параметр не поддерживается в темах с высокой контрастностью.
Показывать строк	0	show_lines	Допустимые значения: от 1 до 100. По умолчанию: 25.
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: - *обязателен*

Примеры

Следующие примеры предназначены только для описания настройки объектов полей виджета панели мониторинга для виджета *Проблемы*. Для получения дополнительной информации о настройке панели мониторинга смотрите `dashboard.create`.

Настройка виджета *Problems*

Настройте виджет *Problems*, который отображает проблемы для группы узлов сети ”4”, удовлетворяющие следующим условиям:

- Проблемы, у которых есть тег с именем ”scope”, содержащий значения ”performance”, ”availability” или ”capacity”.
- Проблемы со следующими уровнями важности: ”Warning”, ”Average”, ”High”, ”Disaster”.

Кроме того, настройте виджет так, чтобы он показывал теги и операционные 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 важности

Описание

Эти параметры и возможные значения свойств для соответствующих объектов полей виджета позволяют настраивать виджет *Проблемы по важности* 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 некорректного создания или обновления виджетов. Чтобы обеспечить успешное создание или обновление виджета *Проблемы по важности*, ознакомьтесь с поведением параметров, описанным в таблицах ниже.

Параметры

Для виджета *Проблемы по важности* 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.
Группы узлов сети	2	groupids.0	ID группы узлов сети. Примечание: Чтобы настроить несколько групп узлов сети, создайте объект поля виджета панели с номером, увеличивающимся 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 *Ссылка*, установленным в "ABCDE") как источник данных для групп узлов сети. Этот параметр не поддерживается при настройке виджета на панели шаблона.
Исключить группы узлов сети	2	exclude_groupids.0	ID группы узлов сети. Примечание: Чтобы исключить несколько групп узлов сети, создайте объект поля виджета панели с номером, увеличивающимся в имени свойства, для каждой группы 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 номером, увеличивающимся 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, к которой принадлежат настроенные узлы сети. Этот параметр не поддерживается при настройке виджета на панели шаблона.

Параметр	type	name	value
Узлы сети (Виджет/Панель)	1	hostids._reference	Вместо ID узла сети: `DASHBOARD.hostids` — укажите селектор узлов сети панели как источник данных для узлов сети; `ABCDE._hostids` — укажите совместимый виджет (с параметром *Ссылка*, установленным в "ABCDE") как источник данных для узлов 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	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 с увеличивающимся номером 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 на порядок тега в списке вычисления тегов. Поведение параметра: - *обязательно* 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 на порядок тега в списке вычисления тегов. Поведение параметра: - *обязательно* при настройке *Тегов проблем*
Значение тега	1	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 проблем*
Показывать	0	show_type	0 - *(по умолчанию)* Группы узлов сети; 1 - Итоги. Этот параметр не поддерживается при настройке виджета на панели шаблона и по умолчанию установлен в значение "Итоги".

Параметр	type	name	value
Макет	0	layout	0 - *(по умолчанию)* Горизонтальный; 1 - Вертикальный. Поведение параметра: - *поддерживается*, если *Показывать* установлено в ”Итоги”
Показывать оперативные данные	0	show_opdata	0 - *(по умолчанию)* Нет; 1 - Отдельно; 2 - С именем 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. Поведение параметра: - *поддерживается*, если *Показывать* установлено в ”Группы узлов сети” Этот параметр не поддерживается при настройке виджета на панели шаблона.
Отображение проблем	0	ext_ack	0 - *(по умолчанию)* Все; 1 - Только неподтвержденные; 2 - Разделено.
Показывать временную шкалу	0	show_timeline	0 - Отключено; 1 - *(по умолчанию)* Включено.
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 настройки объектов полей виджета панели мониторинга *Проблемы по важности*. Для получения дополнительной информации о настройке панели мониторинга смотрите `dashboard.create`.

Настройка виджета *Проблемы по важности*

Настройте виджет *Проблемы по важности*, 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 для соответствующих объектов полей виджета позволяют настроить виджет *Точечная диаграмма* в методах `dashboard.create` и `dashboard.update`.

Attention:
Свойства `fields` виджета не проверяются при создании или обновлении панели. Это позволяет пользователям изменять встроенные виджеты и создавать пользовательские виджеты, но также создает риск некорректного создания или обновления виджетов. Чтобы обеспечить успешное создание или обновление виджета *Точечная диаграмма*, обратитесь к описанию 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"
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	Reference 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

Параметр	type	name	value
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	Ссылка на элемент 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	`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			
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.

Параметр	type	name	value
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.
Переопределить узел сети	1	ds.0.override_hostid._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.

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: 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).
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.

Параметр	type	name	value
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`
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 y_axis_min	Любое числовое значение. По умолчанию: "" (пусто).
Макс.	1	x_axis_max y_axis_max	Любое числовое значение. По умолчанию: "" (пусто).
Единицы измерения (тип)	0	x_axis_units y_axis_units	0 - *(по умолчанию)* Авто; 1 - Статически.

Параметр	type	name	value
Единицы измерения (значение)	1	x_axis_static_units y_axis_static_units	Любое строковое значение. По умолчанию: "" (пусто).

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"
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 для соответствующих объектов полей виджета позволяют настраивать виджет *Отчет SLA* в методах dashboard.create и dashboard.update.

Attention:
Свойства fields виджета не проверяются при создании или обновлении панели. Это позволяет пользователям изменять встроенные виджеты и создавать пользовательские виджеты, но также создает риск некорректного создания или обновления виджетов. Чтобы обеспечить успешное создание или обновление виджета *Отчет SLA*, ознакомьтесь с поведением параметров, описанным в таблицах ниже.

Параметры

Для виджета *Отчет SLA* поддерживаются следующие параметры.

Параметр	type	name	value
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	ID SLA. Поведение параметра: - *обязательно*
Сервис	9	serviceid.0	ID сервиса.
Показывать периоды	0	show_periods	Допустимые значения находятся в диапазоне 1-100. 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*. Для получения дополнительной информации о настройке панели мониторинга смотрите dashboard.create.

Настройка виджета *Отчет SLA*

Настройте виджет *Отчет SLA*, отображающий отчет SLA для SLA "4" сервиса "2" за последние 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 позволяют настраивать виджет *Информация о системе* в методах dashboard.create и dashboard.update.

Attention:
Свойства fields виджета не проверяются при создании или обновлении панели мониторинга. Это позволяет пользователям изменять встроенные виджеты и создавать пользовательские виджеты, но также создает риск некорректного создания или обновления виджетов. Чтобы обеспечить успешное создание или обновление виджета *Информация о системе*, ознакомьтесь с поведением параметров, описанным в таблицах ниже.

Параметры

Для виджета *Информация о системе* 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.
Показывать	0	info_type	0 - *(по умолчанию)* Статистика системы; 1 - Узлы высокой доступности.
Показывать подробности проверки обновлений программного обеспечения	0	show_software_update_check_details	0 - *(по умолчанию)* Отключено; 1 - Включено. Поведение параметра: - *supported* если AllowSoftwareUpdateCheck в конфигурации сервера Zabbix включен и для *Показывать* установлено значение "Статистика системы"

Примеры

Следующие примеры предназначены только для описания настройки объектов полей виджета панели мониторинга для виджета *System information*. Для получения дополнительной информации о настройке панели мониторинга см. dashboard.create.

Настройка виджета *System information*

Настройте виджет *System information*, отображающий 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 Популярные узлы сети

Описание

Эти параметры и возможные значения свойств для соответствующих объектов полей виджета позволяют настраивать виджет *Top hosts* в методах `dashboard.create` и `dashboard.update`.

Attention:
Свойства `fields` виджета не проверяются при создании или обновлении панели. Это позволяет пользователям изменять встроенные виджеты и создавать пользовательские виджеты, но также создает риск некорректного создания или обновления виджетов. Чтобы обеспечить успешное создание или обновление виджета *Top hosts*, ознакомьтесь с поведением параметров, описанным в таблицах ниже.

Параметры

Для виджета *Топ узлов сети* 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.
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, к которой принадлежат указанные узлы сети. Этот параметр не поддерживается при настройке виджета на панели шаблона.

Параметр	type	name	value
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	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: - *обязательно* при настройке *Тегов узлов сети* Этот параметр не поддерживается при настройке виджета на панели шаблона.
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: - *обязательно* 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: - *обязательно* 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.

Параметр	type	name	value
Порядок	0	order	2 - *(по умолчанию)* Топ N; 3 - Нижние N.
Лимит узлов сети	0	show_lines	Возможные значения: от 1 до 1000. По умолчанию: 10. Этот параметр не поддерживается при настройке виджета на панели шаблона.

Столбцы

Столбцы имеют общие параметры и дополнительные параметры в зависимости от конфигурации параметра *Data*.

Note:
Для всех параметров, относящихся к столбцам, число в имени свойства (например, columns.0.name) указывает на столбец, для которого настраивается этот параметр.

Следующие параметры поддерживаются для всех столбцов.

Параметр	type	name	value
Имя	1	columns.0.name	Любое строковое значение. Поведение параметра: - *обязательный*
Данные	0	columns.0.data	1 - Значение элемента данных; 2 - Имя узла сети; 3 - Текст. Поведение параметра: - *обязательный*
Базовый цвет	1	columns.0.base_color	Шестнадцатеричный код цвета (например, FF0000). Поведение параметра: - *обязательный*

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 пороги, а второе число 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 в веб-интерфейсе Zabbix (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	0 - *(по умолчанию)* Числовое; 1 - Текст; 2 - Двоичное.

Параметр	type	name	value
Отображение	0	columns.0.display	1 - *(по умолчанию)* Как есть; 2 - Полоса; 3 - Индикаторы; 6 - Спарклайн. Поведение параметра: - *поддерживается*, если для *Отображать значение элемента данных как* установлено значение ”Numeric”
Мин	1	columns.0.min	Любое числовое значение. Поведение параметра: - *поддерживается*, если для *Отображать значение элемента данных как* установлено значение ”Numeric” и для *Отображение* установлено значение ”Bar” или ”Indicators”
Макс	1	columns.0.max	Любое числовое значение. Поведение параметра: - *поддерживается*, если для *Отображать значение элемента данных как* установлено значение ”Numeric” и для *Отображение* установлено значение ”Bar” или ”Indicators”
Десятичные знаки	0	columns.0.decimal_places	Допустимые значения: 0-10. По умолчанию: 2. Поведение параметра: - *поддерживается*, если для *Отображать значение элемента данных как* установлено значение ”Numeric”
Спарклайн			
Ширина	0	columns.0.sparkline.width	Допустимые значения: 0-10. По умолчанию: 1. Поведение параметра: - *поддерживается*, если для *Отображать значение элемента данных как* установлено значение ”Numeric” и для *Отображение* установлено значение ”Sparkline”
Заполнение	0	columns.0.sparkline.fill	Допустимые значения: 0-10. По умолчанию: 3. Поведение параметра: - *поддерживается*, если для *Отображать значение элемента данных как* установлено значение ”Numeric” и для *Отображение* установлено значение ”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” и для *Отображение* установлено значение ”Sparkline”

Параметр	type	name	value
Период времени	1	columns.0.sparkline.time_period.reference	DASHBOARD._timeperiod - использовать селектор периода времени панели как источник данных; ABCDE._timeperiod - использовать совместимый виджет (с параметром reference, равным ABCDE) как источник данных. По умолчанию: DASHBOARD._timeperiod В качестве альтернативы можно задать период времени только в параметрах *От* и *До*. Поведение параметра: - *поддерживается*, если для *Отображать значение элемента данных как* установлено значение "Numeric" и для *Отображение* установлено значение "Sparkline"
От	1	columns.0.sparkline.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 для *Отображать значение элемента данных как* установлено значение "Numeric" и для *Отображение* установлено значение "Sparkline" - *обязателен*, 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 времени (now, now/d, now/w-1w и т. д.). Поведение параметра: - *поддерживается*, если для *Отображать значение элемента данных как* установлено значение "Numeric" и для *Отображение* установлено значение "Sparkline" - *обязателен*, если задан columns.0.sparkline.time_period.from
Исторические данные	0	columns.0.sparkline.history	0 - *(по умолчанию)* Авто; 1 - History; 2 - Trends. Поведение параметра: - *поддерживается*, если для *Отображать значение элемента данных как* установлено значение "Numeric" и для *Отображение* установлено значение "Sparkline"
Пороги			
Цвет	1	columnsthresholds.0.color	Шестнадцатеричный код цвета (например, FF0000). Поведение параметра: - *поддерживается*, если для *Отображать значение элемента данных как* установлено значение "Numeric"
Порог	1	columnsthresholds.0.threshold	Пороговое значение. Поведение параметра: - *поддерживается*, если для *Отображать значение элемента данных как* установлено значение "Numeric"
Подсветка			
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"

Параметр	type	name	value
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 - min; 2 - max; 3 - avg; 4 - count; 5 - sum; 6 - first; 7 - last.
Период времени	1	columns.0.time_period.reference	DASHBOARD._timeperiod - использовать селектор периода времени панели как источник данных; ABCDE._timeperiod - использовать совместимый виджет (с параметром reference, равным ABCDE) как источник данных. По умолчанию: DASHBOARD._timeperiod В качестве альтернативы 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", "last"
От	1	columns.0.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 и для *Функция агрегации* установлено значение "min", "max", "avg", "count", "sum", "first", "last" - *обязателен*, 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 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" - *обязателен*, если задан columns.0.time_period.from
Исторические данные	0	columns.0.history	0 - *(по умолчанию)* Авто; 1 - History; 2 - Trends. Поведение параметра: - *поддерживается*, если для *Отображать значение элемента данных как* установлено значение "Numeric"

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: - *обязателен*

Текст

Следующие параметры поддерживаются, если для параметра *Data* установлено значение "Текст".

Параметр	тип	имя	значение
Текст	1	columns.0.text	Любое строковое значение, включая макросы. Поддерживаемые макросы: {HOST.*}, {INVENTORY.*}. Поведение параметра: - *обязательно*, если для *Data* установлено значение "Текст"

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 верхние узлы сети по загрузке 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".

Запрос:

```
{
    "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
}
```


Ответ:

```
{
    "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 для соответствующих объектов полей виджета позволяют настраивать виджет *Top items* в методах `dashboard.create` и `dashboard.update`.

Attention:
Свойства `fields` виджета не проверяются при создании или обновлении панели мониторинга. Это позволяет пользователям изменять встроенные виджеты и создавать пользовательские виджеты, но также создает риск некорректного создания или обновления виджетов. Чтобы обеспечить успешное создание или обновление виджета *Top items*, ознакомьтесь с поведением параметров, описанным 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.

Параметр	type	name	value
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.
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.

Параметр	type	name	value
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 *Top items* поддерживаются следующие параметры расширенной настройки.

Параметр	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 объект поля виджета панели мониторинга для каждого шаблона элемента данных с увеличивающимся номером в имени свойства. При настройке виджета на панели мониторинга шаблона следует задавать только шаблоны для элементов данных, настроенных в шаблоне. Поведение параметра: - *обязательно*, если для *Сортировать по* установлено значение ”Значение элемента данных”
Порядок	0	host_ordering_order	2 - *(по умолчанию)* Верхние N; 3 - Нижние N.
Лимит	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.

Параметр	type	name	value
Шаблоны узлов сети	1	item_ordering_host.0	Имя или шаблон узла сети. Примечание: Чтобы настроить несколько шаблонов узлов сети, создайте объект поля виджета панели мониторинга для каждого шаблона узла сети с увеличивающимся номером в имени свойства. Поведение параметра: - *обязательно*, если для *Сортировать по* установлено значение "Узел сети"
Порядок	0	item_ordering_order	2 - *(по умолчанию)* Верхние N; 3 - Нижние N.
Лимит	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 присутствует, обозначает настроенную сущность (например, шаблон элемента данных, тег).

Параметр	type	name	value
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 объект поля виджета панели с отдельным шаблоном элемента данных для каждого из них, увеличивая второе число в имени свойства. При настройке виджета на панели шаблона следует задавать только шаблоны для элементов данных, настроенных в шаблоне. Поведение параметра: - *обязателен*
Теги элементов данных			
Тип вычисления	0	columns.0.item_tags_evaltype	0 - *(по умолчанию)* And/Or; 2 - Or.
Имя тега	1	columns.0.item_tags.0.tag	Любое строковое значение. Примечание: Второе число в имени свойства указывает на порядок тега в списке вычисления тегов. Поведение параметра: - *обязателен* при настройке *Тегов элементов данных*

Параметр	type	name	value
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 на порядок тега в списке вычисления тегов. Поведение параметра: - *обязателен* при настройке *Тегов элементов данных*
Значение тега	1	columns.0.item_tags.0.value	Любое строковое значение. Примечание: Второе число в имени свойства указывает на порядок тега в списке вычисления тегов. Поведение параметра: - *обязателен* 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 - Полоса; 3 - Индикаторы; 6 - Спарклайн.
Мин	1	columns.0.min	Любое числовое значение. Поддерживаются суффиксы (например, ”1d”, ”2w”, ”4K”, ”8G”). Поведение параметра: - *поддерживается*, если *Отображать значение как* установлено в ”Число”, а *Отображение* установлено в ”Полоса” или ”Индикаторы”
Макс	1	columns.0.max	Любое числовое значение. Поддерживаются суффиксы (например, ”1d”, ”2w”, ”4K”, ”8G”). Поведение параметра: - *поддерживается*, если *Отображать значение как* установлено в ”Число”, а *Отображение* установлено в ”Полоса” или ”Индикаторы”
Спарклайн			
Ширина	0	columns.0.sparkline.width	Допустимые значения: от 0 до 10. По умолчанию: 1. Поведение параметра: - *поддерживается*, если *Отображение* установлено в ”Спарклайн”
Заполнение	0	columns.0.sparkline.fill	Допустимые значения: от 0 до 10. По умолчанию: 3. Поведение параметра: - *поддерживается*, если *Отображение* установлено в ”Спарклайн”

Параметр	type	name	value
Цвет	1	columns.0.sparkline.color	Шестнадцатеричный код цвета (например, FF0000). По умолчанию: 42A5F5. Поведение параметра: - *поддерживается*, если *Отображение* установлено в "Спарклайн"
Период времени	1	columns.0.sparkline.time_period.reference	DASHBOARD._timeperiod — использовать селектор периода времени панели как источник данных; ABCDE._timeperiod — использовать совместимый виджет (с параметром reference, равным ABCDE) как источник данных. По умолчанию: DASHBOARD._timeperiod В качестве альтернативы можно задать период времени только в параметрах *От* и *До*. Поведение параметра: - *поддерживается*, если *Отображение* установлено в "Спарклайн"
От	1	columns.0.sparkline.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 *Отображение* установлено в "Спарклайн" - *обязателен*, 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 времени (now, now/d, now/w-1w и т. д.). Поведение параметра: - *поддерживается*, если *Отображение* установлено в "Спарклайн" - *обязателен*, если задан columns.0.sparkline.time_period.from
Исторические данные	0	columns.0.sparkline.history	0 - *(по умолчанию)* Авто; 1 - История; 2 - Тренды. Поведение параметра: - *поддерживается*, если *Отображение* установлено в "Спарклайн"
Пороги			
Цвет	1	columns.0.thresholds.color	Шестнадцатеричный код цвета (например, FF0000). Поведение параметра: - *поддерживается*, если *Отображать значение как* установлено в "Число"
Порог	1	columns.0.thresholds.threshold	Пороговое значение. Поддерживаются суффиксы (например, "1d", "2w", "4K", "8G"). Поведение параметра: - *поддерживается*, если *Отображать значение как* установлено в "Число"
Подсветка			

Параметр	type	name	value
Подсветка	1	columns.0.highlights.0.color	Шестнадцатеричный код цвета (например, `FF0000`). Поведение параметра: - *поддерживается*, если *Отображать значение как* установлено в "Текст"
Порог	1	columns.0.highlights.0.pattern	Любое регулярное выражение. Поведение параметра: - *поддерживается*, если *Отображать значение как* установлено в "Текст"
Десятичные знаки	0	columns.0.decimal_places	Допустимые значения: от 0 до 10. По умолчанию: 2.
Расширенная настройка			
Функция агрегации	0	columns.0.aggregate_function	Если *Отображать значение как* установлено в "Число": 0 - *(по умолчанию)* не используется; 1 - min; 2 - max; 3 - avg; 4 - count; 5 - sum; 6 - first; 7 - last. Если *Отображать значение как* установлено в "Текст": 0 - *(по умолчанию)* не используется; 4 - count; 6 - first; 7 - last.
Период времени	1	columns.0.time_period.reference	`DASHBOARD._timeperiod` — использовать селектор периода времени панели как источник данных; `ABCDE._timeperiod` — использовать совместимый виджет (с параметром `reference`, равным `ABCDE`) как источник данных. По умолчанию: `DASHBOARD._timeperiod` В качестве альтернативы можно задать период времени только в параметрах *От* и *До*. Поведение параметра: - *поддерживается*, если *Функция агрегации* установлена в "min", "max", "avg", "count", "sum", "first", "last"
От	1	columns.0.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 и *Функция агрегации* установлена в "min", "max", "avg", "count", "sum", "first", "last" - *обязателен*, 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 времени (`now`, `now/d`, `now/w-1w` и т. д.). Поведение параметра: - *поддерживается*, если *Период времени* не задан и *Функция агрегации* установлена в "min", "max", "avg", "count", "sum", "first", "last" - *обязателен*, если задан `columns.0.time_period.from`

Параметр	type	name	value
Исторические данные	0	columns.0.history	0 - *(по умолчанию)* Авто; 1 - История; 2 - Тренды. Поведение параметра: - *поддерживается*, если *Отображать значение как* установлено в "Число"
Агрегация	0	columns.0.aggregate_column	0 - *(по умолчанию)* Каждый элемент данных 1 - Объединено. Поведение параметра: - *поддерживается*, если *Отображать значение как* установлено в "Число", а *Отображение* установлено в "Как есть", "Полоса" или "Индикаторы".
Функция объединённой агрегации	0	columns.0.column_aggregate_function	1 - min; 2 - max; 3 - avg; 4 - count; 5 - sum *(по умолчанию)*. Поведение параметра: - *обязателен* при настройке *Агрегации*.
Имя объединённого столбца	1	columns.0.combined_column_name	Любое строковое значение. Поведение параметра: - *обязателен* при настройке *Агрегации*.

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 для соответствующих объектов полей виджета панели мониторинга позволяют настраивать виджет *Топ триггеров* в методах `dashboard.create` и `dashboard.update`.

Attention:
Свойства `fields` виджета не проверяются при создании или обновлении панели мониторинга. Это позволяет пользователям изменять встроенные виджеты и создавать пользовательские виджеты, но также создает риск некорректного создания или обновления виджетов. Чтобы обеспечить успешное создание или обновление виджета *Топ триггеров*, пожалуйста, ознакомьтесь с поведением параметров, описанным в таблицах ниже.

Параметры

Для виджета *Топ триггеров* поддерживаются следующие параметры.

Параметр	type	name	value
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	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.
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 вообще, либо должен быть настроен как минимум с одной группой узлов сети, к которой принадлежат 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).

Параметр	type	name	value
Важность	0	severities.0	Важности триггера. 0 - Не классифицировано; 1 - Информация; 2 - Предупреждение; 3 - Средняя; 4 - Высокая; 5 - Чрезвычайная. По умолчанию: пусто (все включены). Примечание: Чтобы настроить несколько значений, создайте объект поля виджета панели с увеличивающимся номером в имени свойства для каждого значения.
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 на порядок тега в списке вычисления тегов. Поведение параметра: - *обязательно* 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 на порядок тега в списке вычисления тегов. Поведение параметра: - *обязательно* при настройке *Тегов проблем*
Значение тега	1	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 проблем*
Период времени	1	time_period._reference	`DASHBOARD._timeperiod` — использовать селектор периода времени панели как источник данных; `ABCDE._timeperiod` — использовать совместимый виджет (с параметром *Reference*, установленным в "`ABCDE`") как источник 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 формате времени (`now`, `now/d`, `now/w-1w` и т. д.). Поведение параметра: - *поддерживается*, если *Период времени* не задан - *обязательно*, 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 формате времени (now, now/d, now/w-1w и т. д.). Поведение параметра: - *поддерживается*, если *Период времени* не задан - *обязательно*, если задан time_period.from
Лимит 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 *Топ триггеров*. Для получения дополнительной информации о настройке панели мониторинга см. dashboard.create.

Настройка виджета *Top triggers*

Настройте виджет *Top triggers*, который отображает 5 лучших триггеров для группы узлов сети "4" с количеством всех проблем для каждого триггера. Виджет отображает только триггеры с важностью "Предупреждение", "Средняя", "Высокая" или "Катастрофа", а также проблемы, имеющие тег с именем "scope", содержащий значения "performance", "availability" или "capacity".

Запрос:

```
{
    "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
}
```

Ответ:

```
{
    "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 для соответствующих объектов полей виджета позволяют настраивать 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 некорректного создания или обновления виджетов. Чтобы обеспечить успешное создание или обновление виджета *Обзор триггеров*, ознакомьтесь с поведением параметров, описанным в таблицах ниже.

Параметры

Для виджета *Обзор триггеров* 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.
Показывать	0	show	1 - *(по умолчанию)* Последние проблемы; 2 - Любые; 3 - Проблемы.

Параметр	type	name	value
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, к которой принадлежат 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` — укажите селектор узла 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 проблем			
Тип вычисления	0	evaltype	0 - *(по умолчанию)* И/Или; 2 - Или.
Имя тега	1	tags.0.tag	Любое строковое значение. Примечание: Число в имени свойства указывает порядок тега в списке вычисления тегов. Поведение параметра: - *обязательно* 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: Число в имени свойства указывает порядок тега в списке вычисления тегов. Поведение параметра: - *обязательно* при настройке *Тегов проблем*

Параметр	type	name	value
Значение тега	1	tags.0.value	Любое строковое значение. Примечание: Число в имени свойства указывает порядок тега в списке вычисления тегов. Поведение параметра: - *обязательно* при настройке *Тегов проблем*
Показывать подавленные проблемы	0	show_suppressed	0 - *(по умолчанию)* Отключено; 1 - Включено.
Макет	0	layout	0 - *(по умолчанию)* Горизонтальный; 1 - Вертикальный.

Примеры

Следующие примеры предназначены только для описания настройки объектов полей виджета панели мониторинга для виджета *Обзор триггеров*. Для получения дополнительной информации о настройке панели мониторинга смотрите `dashboard.create`.

Настройка виджета *Обзор триггеров*

Настройте виджет *Обзор триггеров*, который отображает состояния триггеров для всех групп узлов сети, имеющих триггеры с тегом с именем "scope" и содержащим значение "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 позволяют настраивать виджет *URL* в методах `dashboard.create` и `dashboard.update`.

Attention:
Свойства `fields` виджета не проверяются при создании или обновлении панели мониторинга. Это позволяет пользователям изменять встроенные виджеты и создавать пользовательские виджеты, но также создает риск некорректного создания или обновления виджетов. Чтобы обеспечить успешное создание или обновление виджета *URL*, ознакомьтесь с поведением параметров, описанным в таблицах ниже.

Параметры

Следующие параметры поддерживаются для виджета *URL*.

Параметр	type	name	value
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.

Параметр	type	name	value
URL	1	url	Корректная строка URL. Поведение параметра: - *обязательно*
Переопределить узел сети	1	override_hostid._reference	ABCDE._hostid - задать совместимый виджет (с его параметром *Reference*, установленным в "ABCDE") в качестве источника данных для узлов сети; DASHBOARD._hostid - задать селектор узла сети панели как источник данных для узлов сети. Этот параметр не поддерживается при настройке виджета на панели шаблона.

Примеры

Следующие примеры предназначены только для описания настройки объектов полей виджета панели мониторинга для виджета *URL*. Дополнительную информацию о настройке панели мониторинга смотрите в dashboard.create.

Настройка виджета *URL*

Настройте виджет *URL*, который отображает домашнюю 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 при создании или обновлении панели. Это позволяет пользователям изменять встроенные виджеты и создавать пользовательские виджеты, но также создает риск некорректного создания или обновления виджетов. Чтобы обеспечить успешное создание или обновление виджета *Веб-мониторинг*, ознакомьтесь с поведением параметров, описанным в таблицах ниже.

Параметры

Для виджета *Веб-мониторинг* 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.
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.

Параметр	type	name	value
Группы узлов сети (Виджет)	1	groupids._reference	Вместо ID группы узлов сети: `ABCDE._hostgroupids` — укажите 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.
Исключить группы узлов сети	2	exclude_groupids.0	ID группы узлов сети. Примечание: Чтобы исключить 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.
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. При указании нескольких узлов сети параметр *Группы узлов сети* либо не должен быть настроен вовсе, либо должен быть настроен как минимум с одной группой узлов сети, к 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 *Ссылка*, установленным в "ABCDE") в качестве источника данных для узлов сети. Этот параметр не поддерживается при настройке виджета на панели шаблона.
Теги сценария			
Тип вычисления	0	evaltype	0 - *(по умолчанию)* И/Или; 2 - Или.
Имя тега	1	tags.0.tag	Любое строковое значение. Примечание: Число в имени свойства указывает порядок тега в списке вычисления тегов. Поведение параметра: - *обязательно* при настройке *Тегов сценария*
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: - *обязательно* при настройке *Тегов сценария*

Параметр	type	name	value
Значение тега	1	tags.0.value	Любое строковое значение. Примечание: Число в имени свойства указывает порядок тега в списке вычисления тегов. Поведение параметра: - *обязательно* при настройке *Тегов сценария*
Показывать узлы сети в обслуживании	0	maintenance	0 - Отключено; 1 - *(по умолчанию)* Включено.
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 настройки объектов полей виджета панели мониторинга для виджета *Веб-мониторинг*. Для получения дополнительной информации о настройке панели мониторинга смотрите dashboard.create.

Настройка виджета *Web monitoring*

Настройте виджет *Web monitoring*, отображающий 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`.

Дашборд шаблона

Объект дашборда шаблона имеет следующие свойства.

Свойство	Тип	Описание
dashboardid	ID	ID дашборда шаблона. Поведение свойства: - *только для чтения* - *обязательно* для операций обновления

Свойство	Тип	Описание
name	string	Имя дашборда шаблона. Поведение свойства: - *обязательно* для операций создания
templateid	ID	ID шаблона, которому принадлежит дашборд. Поведение свойства: - *константа* - *обязательно* для операций создания
display_period	integer	Период отображения страницы по умолчанию (в секундах). Возможные значения: 10, 30, 60, 120, 600, 1800, 3600. По умолчанию: 30.
auto_start	integer	Автоматически запускать слайд-шоу. Возможные значения: 0 - не запускать слайд-шоу автоматически; 1 - *(по умолчанию)* запускать слайд-шоу автоматически.
uuid	string	Универсальный уникальный идентификатор, используемый для связывания импортированных дашбордов шаблонов с уже существующими. Генерируется автоматически, если не указан.

Страница панели шаблона

Объект страницы панели шаблона имеет следующие свойства.

Свойство	Тип	Описание
dashboard_pageid	ID	ID страницы панели. Поведение свойства: - *только для чтения*
name	string	Имя страницы панели. По умолчанию: пустая строка.
display_period	integer	Период отображения страницы панели (в секундах). Возможные значения: 0, 10, 30, 60, 120, 600, 1800, 3600. По умолчанию: 0 (будет использоваться период отображения страницы по умолчанию).
widgets	array	Массив объектов виджета панели шаблона.

Виджет панели шаблона

Объект виджета панели шаблона имеет следующие свойства.

Свойство	Тип	Описание
widgetid	ID	ID виджета панели. Поведение свойства: - *только для чтения*

Свойство	Тип	Описание
type	string	Тип виджета панели. Возможные значения: actionlog - Журнал действий; clock - Часы; discovery - Состояние обнаружения; favgraphs - Избранные графики; favmaps - Избранные карты; gauge - Измеритель; 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 картам; piechart - Круговая диаграмма; plaintext - Простой текст; problemhosts - Узлы сети с проблемами; problems - Проблемы; problemsbysv - Проблемы по важности; scatterplot - Диаграмма рассеяния; 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 - Веб-мониторинг. Поведение свойства: - *обязательное*
name	string	Пользовательское имя виджета.
x	integer	Горизонтальная позиция от левой стороны панели. Возможные значения находятся в диапазоне от 0 до 71.
y	integer	Вертикальная позиция от верхней стороны панели. Возможные значения находятся в диапазоне от 0 до 63.
width	integer	Ширина виджета. Возможные значения находятся в диапазоне от 1 до 72.
height	integer	Высота виджета. Возможные значения находятся в диапазоне 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 поля виджета панели шаблона.

Поле виджета панели шаблона

Объект поля виджета панели шаблона имеет следующие свойства.

Свойство	Тип	Описание
type	integer	Тип поля виджета. Возможные значения: 0 - Целое число; 1 - Строка; 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. Поведение свойства: - *обязательно*
name	string	Имя поля виджета. Возможные значения: см. Поля виджетов панели. Обратите внимание, что некоторые параметры, связанные с узлом сети (например, *Группы узлов сети*, *Исключить группы узлов сети* и *Узлы сети* в виджете *Проблемы*, *Группы узлов сети* в виджете *Доступность узла сети* и т. д.), недоступны при настройке виджета на панели шаблона. Это связано с тем, что панели шаблонов отображают данные только того узла сети, с которым связан шаблон. Поведение свойства: - *обязательно*
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, *Группы узлов сети*, *Исключить группы узлов сети* и *Узлы сети* в виджете *Проблемы*, *Группы узлов сети* в виджете *Доступность узла сети* и т. д.), недоступны при настройке виджета на панели шаблона. Это связано с тем, что панели шаблонов отображают данные только того узла сети, с которым связан шаблон. Поведение свойства: - *обязательно*

templatedashboard.create

Описание

`object templatedashboard.create( / templateDashboards)`

Этот метод позволяет создавать новые шаблоны информационных панелей.

Note:
Этот метод доступен только пользователям с типом доступа *Администратор* и *Супер-администратор*. Разрешения на вызов метода можно отозвать в настройках ролей пользователя. Дополнительную информацию см. в разделе Роли пользователей.

Параметры

`(object/array)` Шаблоны панелей мониторинга для создания.

В дополнение к стандартным свойствам панели мониторинга шаблона метод принимает следующие параметры.

Параметр	Тип	Описание
pages	array	Страницы панели мониторинга шаблона, которые должны быть созданы для панели мониторинга. Страницы панели мониторинга будут упорядочены в том же порядке, в котором они указаны. Поведение параметра: - *обязательный*

Возвращаемые значения

( ) Возвращает объект, который содержит ID созданных шаблонов панелей, указанных в свойстве dashboardids. Порядок возвращаемых идентификаторов соответствует порядку переданных шаблонов панелей.

Примеры

Создание шаблона панели

Создайте шаблон панели с именем «Graphs» с одним виджетом Graph на одной странице панели.

Запрос:

```
{
    "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
}
```

Ответ:

```
{
    "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(  templateDashboardIds)`

Этот метод позволяет удалять панели шаблонов.

> **Note:**
> Этот метод доступен только пользователям с типом доступа *Администратор* и *Супер-администратор*. Разрешения на вызов метода можно отозвать в настройках ролей 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

Удалите две панели 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( )

Метод позволяет получить шаблоны панелей мониторинга в соответствии с заданными параметрами.

> **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) Параметры, определяющие желаемый вывод.

Метод поддерживает следующие параметры.

Параметр	Тип	Описание
dashboardids	ID/array	Возвращать только шаблоны панелей с указанными ID.
templateids	ID/array	Возвращать только шаблоны панелей, принадлежащие указанным шаблонам.
selectPages	query	Возвращать свойство pages со страницами шаблона панели, в правильном порядке.
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 панелей шаблона

Получение всех панелей 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(  /  templateDashboards)`

Этот метод позволяет обновлять существующие шаблоны информационных панелей.

Note:
Этот метод доступен только пользователям с типом доступа *Администратор* и *Супер-администратор*. Разрешения на вызов метода можно отозвать в настройках ролей пользователя. Дополнительную информацию см. в разделе Роли пользователей.

Параметры

`(object/array)` Свойства шаблона панели, которые необходимо обновить.

Свойство `dashboardid` должно быть указано для каждой панели, все остальные свойства являются необязательными. Будут обновлены только указанные свойства.

В дополнение к стандартным свойствам шаблона панели, метод принимает следующие параметры.

Параметр	Type	Описание
pages	array	Страницы шаблона панели для замены существующих страниц панели. Страницы панели обновляются по свойству `dashboard_pageid`. Для объектов без свойства `dashboard_pageid` будут созданы новые страницы панели, а существующие страницы панели будут удалены, если не будут использованы повторно. Страницы панели будут упорядочены в том же порядке, в котором они указаны. Будут обновлены только указанные свойства страниц панели. Для свойства `pages` требуется как минимум один объект страницы панели.

Возвращаемые значения

`(object)` Возвращает объект, содержащий идентификаторы обновлённых панелей шаблона в свойстве `dashboardids`.

Примеры

Переименование дашборда шаблона

Переименуйте дашборд 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 панели шаблона

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": "                "
            },
            {
                "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*.

Пользователь

Этот класс предназначен для работы с пользователями.

Ссылки на объекты:

- User
- Media

Доступные методы:

- user.checkauthentication - проверка и продление пользовательских сессий
- 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 TOTP-секретов пользователей
- user.unblock - разблокировка пользователей
- user.update - обновление пользователей

Объект пользователя

Следующие объекты напрямую связаны с API `user`.

Пользователь

Объект пользователя имеет следующие свойства.

Свойство	Type	Описание
userid	ID	ID 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` имеет корректное значение, отличное от ”0”), включен статус подготовки каталога пользователей (`provision_status` объекта User directory object установлен в ”1”), а также включен статус аутентификации для всей LDAP- или SAML-подготовки (`ldap_jit_status` объекта Authentication object установлен в ”Enabled for configured LDAP IdPs” или `saml_jit_status` объекта Authentication object установлен в ”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	ID роли пользователя. Обратите внимание, что пользователи без роли могут входить в Zabbix только с использованием аутентификации LDAP или SAML, при условии, что их информация LDAP/SAML соответствует сопоставлениям групп пользователей, настроенным в Zabbix.
attempt_clock	timestamp	Время последней неуспешной попытки входа. Поведение свойства: - *только для чтения*
attempt_failed	integer	Количество недавних неуспешных попыток входа. Поведение свойства: - *только для чтения*
attempt_ip	string	IP-адрес, с которого была выполнена последняя неуспешная попытка входа. Поведение свойства: - *только для чтения*

Свойство	Type	Описание
autologin	integer	Следует ли включить автоматический вход. Возможные значения: 0 - *(по умолчанию)* автоматический вход отключен; 1 - автоматический вход включен.
autologout	string	Время жизни сеанса пользователя. Принимает секунды и единицу времени с суффиксом. Если установлено значение 0s, срок действия сеанса никогда не истечет. По умолчанию: 15m.
lang	string	Код языка пользователя, например `en_US`. По умолчанию: `default` - системное значение по умолчанию.
name	string	Имя пользователя.
provisioned	integer	Был ли пользователь подготовлен. Возможные значения: 0 - не подготовлен; 1 - подготовлен. Поведение свойства: - *только для чтения*
refresh	string	Период автоматического обновления. Принимает секунды или единицу времени с суффиксом (например, 30s, 90s, 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` - Blue; `dark-theme` - Dark.
ts_provisioned	timestamp	Время выполнения последней операции подготовки. Поведение свойства: - *только для чтения* - *поддерживается* для пользователей типа *Super admin*
url	string	URL страницы, на которую пользователь будет перенаправлен после входа в систему.
userdirectoryid	ID	ID каталога пользователей, с которым связан пользователь. Используется для подготовки (создания или обновления), а также для входа пользователя, связанного с каталогом пользователей. Для операций входа значение этого свойства будет иметь приоритет над свойством `userdirectoryid` групп пользователей, к которым принадлежит пользователь. По умолчанию: 0. Поведение свойства: - *только для чтения* - *поддерживается* для пользователей типа *Super admin*

Свойство	Type	Описание
timezone	string	Часовой пояс пользователя, например `Europe/London`, `UTC`. По умолчанию: `default` - системное значение по умолчанию. Полный список поддерживаемых часовых поясов см. в документации PHP.

Способ оповещения

Объект способа оповещения имеет следующие свойства.

Свойство	Тип	Описание
mediaid	ID	ID способа оповещения пользователя. Поведение свойства: - *только для чтения*
mediatypeid	ID	ID типа способа оповещения, используемого способом оповещения пользователя. Поведение свойства: - *обязательно*
sendto	string/array	Адрес, имя пользователя или другой идентификатор получателя. Если `type` у типа способа оповещения установлен в значение "Email", значения представлены в виде массива. Для других типов способов оповещения значение представлено в виде строки. Поведение свойства: - *обязательно*
active	integer	Включен ли способ оповещения. Возможные значения: 0 - *(по умолчанию)* включен; 1 - отключен.
severity	integer	Важности триггеров, о которых следует отправлять уведомления. Возможные значения битовой маски: 1 - Не классифицировано; 2 - Информация; 4 - Предупреждение; 8 - Средняя; 16 - Высокая; 32 - Катастрофическая. Это поле битовой маски; допустима любая сумма возможных значений битовой маски (например, 48 для Средняя, Высокая и Катастрофическая). По умолчанию: 63.
period	string	Время, когда могут отправляться уведомления, в виде периода времени или пользовательских макросов, разделенных точкой с запятой. По умолчанию: 1-7,00:00-24:00.
provisioned	integer	Был ли пользователь подготовлен. Возможные значения: 0 - не подготовлен; 1 - подготовлен.

Свойство	Тип	Описание
userdirectory_mediaid	ID	ID сопоставления способа оповещения каталога пользователей для подготовленного способа оповещения. Поведение свойства: - *только для чтения* - *поддерживается* для пользователей типа *Super admin*

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.

Параметр	Тип	Описание
extend	boolean	Следует ли продлевать пользовательскую сессию. Значение по умолчанию: ”true”. Установка значения ”false” позволяет проверить пользовательскую сессию без её продления. Поведение параметра: - *поддерживается*, если задан `sessionid`
sessionid	string	Пользовательский токен аутентификации. Поведение параметра: - *обязателен*, если `token` не задан
token	string	Пользовательский API-токен. Поведение параметра: - *обязателен*, если `sessionid` не задан

Возвращаемые значения

`(object)` Возвращает объект, содержащий информацию о пользователе.

В дополнение к стандартным свойствам пользователя возвращается следующая информация.

Свойство	Type	Описание
auth_type	integer	Аутентификация по умолчанию для пользователя. Список возможных значений см. в свойстве `authentication_type` объекта Authentication object.
debug_mode	integer	Включён или отключён ли для пользователя режим отладки. Список возможных значений см. в свойстве `debug_mode` объекта User group object.
deprovisioned	boolean	Принадлежит ли пользователь к группе деактивированных пользователей.

Свойство	Type	Описание
gui_access	string	Метод аутентификации пользователя в веб-интерфейс. Список возможных значений см. в свойстве `gui_access` объекта User group object.
secret	string	Случайная строка из 32 символов. Генерируется при входе пользователя в систему. Свойство `secret` не возвращается, если пользовательская сессия проверяется с использованием API-токена.
sessionid	string	Токен аутентификации, который должен использоваться в последующих API-запросах. Свойство `sessionid` не возвращается, если пользовательская сессия проверяется с использованием API-токена.
type	integer	Тип пользователя. Список возможных значений см. в свойстве `type` объекта Role object.
userip	string	IP-адрес пользователя.

Примеры

Проверка аутентификации с использованием токена аутентификации

Проверьте и продлите пользовательский сеанс с использованием 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

Проверьте пользовательскую сессию с помощью API-токена пользователя и верните 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( / users)`

Этот метод позволяет создавать новых пользователей.

> **Note:**
> Этот метод доступен только пользователю с типом доступа *Супер-администратор*. Разрешения на вызов метода можно отозвать в настройках ролей пользователя. Дополнительную информацию см. в разделе Роли пользователей.

Параметры

`(object/array)` Пользователи для создания.

В дополнение к стандартным свойствам пользователя, метод принимает следующие параметры.

Параметр	Тип	Описание
usrgrps	array	Группы пользователей, в которые нужно добавить пользователя. Для групп пользователей должно быть определено только свойство `usrgrpid`.
medias	array	Способы оповещения пользователя, которые будут созданы.

Возвращаемые значения

`( )` 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 новое средство оповещения.

Запрос:

```
{
    "jsonrpc": "2.0",
    "method": "user.create",
    "params": {
        "username": "John",
        "passwd": "Doe123",
        "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
- Способ оповещения
- Группа пользователей
- Роль

Источник

CUser::create() в *ui/include/classes/api/services/CUser.php*.

user.delete

Описание

`object user.delete(    users)`

Этот метод позволяет удалять пользователей.

> **Note:**
> Этот метод доступен только пользователю с типом доступа *Супер-администратор*. Разрешения на вызов метода можно отозвать в настройках ролей пользователя. Дополнительную информацию см. в разделе Роли пользователей.

Параметры

( ) ID удаляемых пользователей.

Возвращаемые значения

( ) Возвращает объект, который содержит ID удаленных пользователей, указанных в свойстве `userids`.

Примеры

Удаление нескольких пользователей

Удалите 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(          )`

Метод позволяет получать пользователей по заданным параметрам.

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)` Параметры, определяющие желаемый вывод.

Метод поддерживает следующие параметры.

Note:
При запросе пользовательских способов оповещения, прав доступа или информации о роли пользователи типа *Admin* и *User* могут получать данные только о собственном пользователе. Пример см. в разделе Получение пользователей от имени *Admin*.

Параметр	Тип	Описание
mediaids	ID/array	Возвращать только пользователей, использующих указанные способы оповещения.
mediatypeids	ID/array	Возвращать только пользователей, использующих указанные типы способов оповещения.
userids	ID/array	Возвращать только пользователей с указанными ID.
usrgrpids	ID/array	Возвращать только пользователей, принадлежащих указанным группам пользователей.
getAccess	flag	Добавляет дополнительную информацию о правах доступа пользователя. Добавляет следующие свойства для каждого пользователя: `gui_access` - `(integer)` метод аутентификации пользователя в веб-интерфейс. 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 способы оповещения, используемые пользователем, в свойстве `medias`.
selectMediatypes	query	Возвращать типы способов оповещения, используемые пользователем, в свойстве `mediatypes`. Ограничения в зависимости от типа пользователя см. в `mediatype.get`.

Параметр	Тип	Описание
selectUsrgrps	query	Возвращать группы пользователей, к которым принадлежит пользователь, в свойстве `usrgrps`. Ограничения в зависимости от типа пользователя см. в `usergroup.get`.
selectRole	query	Возвращать роль пользователя в свойстве `role`.
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. Не поддерживает свойства типа данных `text` data type. Возможные свойства объекта User для пользователей типа *Admin* и *User* при запросе данных о пользователях из их группы пользователей: `userid`, `name`, `surname`, `username`.
output	query	Свойства объекта User, которые должны быть возвращены. Пользователи типа *Admin* и *User* могут получать только следующие свойства: - Для 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 из своей группы пользователей: `userid`, `name`, `surname`, `username`. По умолчанию: `extend`.
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 "%…%"`. Поддерживает только свойства типа данных `string` и `text` data type. Возможные свойства объекта User для пользователей типа *Admin* и *User* при запросе данных о пользователях из их группы пользователей: `name`, `surname`, `username`.
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

Получение 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",
            "refresh": "0s",
            "theme": "default",
            "attempt_failed": "0",
            "attempt_ip": "",
            "attempt_clock": "0",
            "rows_per_page": "50",
            "timezone": "default",
            "roleid": "3",
            "userdirectoryid": "0",
            "ts_provisioned": "0"
        },
        {
            "userid": "2",
            "username": "guest",
            "name": "",
            "surname": "",
            "url": "",
            "autologin": "0",
            "autologout": "15m",
            "lang": "default",
            "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"
        },
        {
            "userid": "3",
            "username": "user",
            "name": "Zabbix",
```

```
            "surname": "User",
            "url": "",
            "autologin": "0",
            "autologout": "0",
            "lang": "ru_RU",
            "refresh": "15s",
            "theme": "dark-theme",
            "attempt_failed": "0",
            "attempt_ip": "",
            "attempt_clock": "0",
            "rows_per_page": "100",
            "timezone": "default",
            "roleid": "1",
            "userdirectoryid": "0",
            "ts_provisioned": "0"
        }
    ],
    "id": 1
}
```

Получение пользователей от имени *Admin*

Как пользователь типа *Admin*, получите подробные данные о собственном пользователе и ограниченные данные о пользователях из вашей 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",
            "usrgrps": [
                {
                    "usrgrpid": "7",
                    "name": "Zabbix administrators",
                    "gui_access": "0",
                    "users_status": "0",
                    "debug_mode": "0",
                    "mfa_status": "0"
                }
            ]
        },
        {
            "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",
            "gui_access": "0",
            "debug_mode": "0",
            "users_status": "0",
            "usrgrps": [
                {
                    "usrgrpid": "7",
                    "name": "Zabbix administrators",
                    "gui_access": "0",
                    "users_status": "0",
                    "debug_mode": "0",
                    "mfa_status": "0"
                }
            ],
            "medias": [
                {
                    "mediaid": "2",
                    "mediatypeid": "1",
                    "sendto": [
                        "john.doe@example.com"
                    ],
                    "active": "0",
                    "severity": "63",
                    "period": "1-7,00:00-24:00",
                    "provisioned": 0
                }
            ],
            "mediatypes": [
                {
                    "mediatypeid": "1",
                    "type": "0",
                    "name": "Email",
                    "status": "0",
                    "description": "",
                    "maxattempts": "3"
                }
            ],
            "role": {
                "roleid": "2",
                "name": "Admin role",
                "type": "2",
                "readonly": "0"
            }
        }
    ],
    "id": 1
}
```


Получение данных пользователя

Получите данные пользователя с ID "12".

Запрос:

```
{
    "jsonrpc": "2.0",
    "method": "user.get",
    "params": {
        "output": ["userid", "username"],
        "selectRole": "extend",
        "userids": "12"
    },
    "id": 1
}
```

Ответ:

```
{
  "jsonrpc": "2.0",
  "result": [
      {
          "userid": "12",
          "username": "John",
          "role": {
              "roleid": "5",
              "name": "Operator",
              "type": "1",
              "readonly": "0"
          }
      }
  ],
  "id": 1
}
```

См. также

- Способ оповещения
- Тип способа оповещения
- 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 выполнить вход в API и сгенерировать токен аутентификации.

Warning:
При использовании этого метода также необходимо выполнить user.logout, чтобы предотвратить создание большого количества записей об открытых сессиях.

Attention:
Этот метод доступен только неаутентифицированным пользователям, которые не принадлежат ни к одной группе пользователей с включённой многофакторной аутентификацией.

Параметры

`(object)` Параметры, содержащие имя пользователя и пароль.

Метод принимает следующие параметры.

Параметр	Тип	Описание
password	string	Пароль пользователя. Поведение параметра: - *обязательный*
username	string	Имя пользователя. Поведение параметра: - *обязательный*
userData	flag	Возвращать информацию об аутентифицированном пользователе.

Возвращаемые значения

`(string/object)` Если используется параметр `userData`, возвращается объект, содержащий информацию об аутентифицированном пользователе.

В дополнение к стандартным свойствам пользователя возвращается следующая информация:

Свойство	Type	Описание
auth_type	integer	Аутентификация пользователя по умолчанию. Список возможных значений см. в свойстве `authentication_type` объекта Authentication object.
debug_mode	integer	Включён или отключён режим отладки для пользователя. Список возможных значений см. в свойстве `debug_mode` объекта User group object.
deprovisioned	boolean	Принадлежит ли пользователь к группе деактивированных пользователей.
gui_access	string	Метод аутентификации пользователя в веб-интерфейс. Список возможных значений см. в свойстве `gui_access` объекта User group object.
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. Список возможных значений см. в свойстве `type` объекта Role object.
userip	string	IP-адрес пользователя.

Note:
Если пользователь был успешно аутентифицирован после одной или нескольких неудачных попыток, метод вернёт 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(    )`

Этот метод позволяет выйти из API и аннулирует текущий токен аутентификации.

> **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

`(    )` 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(    /     users)`

Этот метод позволяет предоставлять пользователей LDAP.

Note:
Этот метод доступен только пользователю с типом доступа *Супер-администратор*. Разрешения на вызов метода можно отозвать в настройках ролей пользователя. Дополнительную информацию 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( / users)`

Этот метод позволяет сбросить TOTP-секреты пользователя.

Note:
Этот метод доступен только пользователю с типом доступа *Супер-администратор*. Разрешения на вызов метода можно отозвать в настройках ролей пользователя. Дополнительную информацию см. в разделе Роли пользователей.

Параметры

`(array)` Идентификаторы пользователей, для которых необходимо сбросить секреты TOTP.

Note:
Сеансы пользователей для указанных пользователей также будут удалены (кроме пользователя, отправляющего запрос).

Возвращаемые значения

(object) Возвращает объект, содержащий идентификаторы пользователей, для которых были сброшены секреты TOTP, в свойстве userids.

Примеры

Сброс TOTP-секретов для нескольких пользователей

Сбросьте TOTP-секреты 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(    userids)`

Этот метод позволяет разблокировать пользователей.

> **Note:**
> Этот метод доступен только пользователю с типом доступа *Супер-администратор*. Разрешения на вызов метода можно отозвать в настройках ролей пользователя. Дополнительную информацию см. в разделе Роли пользователей.

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(  /  users)`

Этот метод позволяет обновлять существующих пользователей.

Note:
Этот метод доступен пользователям любого типа. Разрешения на вызов метода можно отозвать в настройках ролей пользователя. Дополнительную информацию см. в разделе Роли пользователей.

Note:
Сложность пароля пользователя проверяется в соответствии с правилами политики паролей, определенными API аутентификации. Подробнее см. в API аутентификации.

Параметры

`(object/array)` Свойства пользователя, которые необходимо обновить.

Свойство `userid` должно быть определено для каждого пользователя, все остальные свойства являются необязательными. Будут обновлены только переданные свойства, все остальные останутся без изменений.

В дополнение к стандартным свойствам пользователя, метод принимает следующие параметры.

Параметр	Тип	Описание
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 object задано и пользователь изменяет собственный пароль

Параметр	Тип	Описание
usrgrps	array	Группы пользователей для замены существующих групп пользователя. Для групп пользователей должно быть определено только свойство `usrgrpid`.
medias	array	Способы оповещения пользователя для замены существующих, не подготовленных автоматически способов оповещения. Подготовленные автоматически способы оповещения можно не указывать при обновлении способов оповещения.

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*.

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.

Свойство	Type	Описание
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 Vault.
value	string	Значение макроса. Поведение свойства: - *только для записи*, если `type` установлен 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.

Свойство	Тип	Описание
hostmacroid	ID	ID макроса. Поведение свойства: - *только для чтения* - *обязательно* для операций обновления
automatic	integer	Определяет, управляется ли макрос правилом обнаружения. Возможные значения: 0 - *(по умолчанию)* Макрос управляется пользователем; 1 - Макрос управляется правилом обнаружения. Пользователю не разрешено создавать автоматический макрос. Чтобы обновить автоматический макрос, его необходимо преобразовать в ручной.
config	object/array	Конфигурация макроса, отвечающая за то, как макрос отображается в мастере узла сети.
description	string	Описание макроса.
hostid	ID	ID узла сети, прототипа узла сети или шаблона, которому принадлежит макрос. Поведение свойства: - *постоянное* - *обязательно* для операций создания
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 Vault.
value	string	Значение макроса. Поведение свойства: - *только для записи*, если `type` установлен в ”Secret macro” - *обязательно* для операций создания

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*

Свойство	Type	Описание
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.
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( / hostMacros)`

Этот метод позволяет создавать новые макросы хоста.

> **Note:**
> Этот метод доступен только пользователям с типом доступа *Администратор* и *Супер-администратор*. Разрешения на вызов метода можно отозвать в настройках ролей пользователя. Дополнительную информацию см. в разделе Роли пользователей.

Параметры

`(object/array)` Макросы узла сети для создания.

Метод принимает макросы узла сети со стандартными свойствами макросов 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".

Запрос:

```
{
    "jsonrpc": "2.0",
    "method": "usermacro.create",
    "params": {
        "hostid": "10198",
        "macro": "{$SNMP_COMMUNITY}",
        "value": "public"
    },
    "id": 1
}
```

Ответ:

```
{
    "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(  /  globalMacros)`

Этот метод позволяет создавать новые глобальные макросы.

> **Note:**
> Этот метод доступен только пользователю с типом доступа *Супер-администратор*. Разрешения на вызов метода можно отозвать в настройках ролей пользователя. Дополнительную информацию 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(    hostMacroIds)`

Этот метод позволяет удалять макросы хоста.

> **Note:**
> Этот метод доступен только пользователям с типом доступа *Администратор* и *Супер-администратор*. Разрешения на вызов метода можно отозвать в настройках ролей 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

Удалите два макроса 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.createglobal(  /  globalMacros)`

Этот метод позволяет создавать новые глобальные макросы.

> **Note:**
> Этот метод доступен только пользователю с типом доступа *Супер-администратор*. Разрешения на вызов метода можно отозвать в настройках ролей пользователя. Дополнительную информацию см. в разделе Роли пользователей.

Параметры

`(array)` Идентификаторы глобальных макросов для удаления.

Возвращаемые значения

`(object)` Возвращает объект, содержащий идентификаторы удалённых глобальных макросов в свойстве `globalmacroids`.

Примеры

Удаление нескольких глобальных макросов

Удалите два глобальных макроса.

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(          )`

Метод позволяет получить хостовые и глобальные макросы в соответствии с заданными параметрами.

> **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)` Параметры, определяющие желаемый вывод.

Метод поддерживает следующие параметры.

Параметр	Тип	Описание
globalmacro	flag	Возвращать глобальные макросы вместо макросов узла сети.
globalmacroids	ID/array	Возвращать только глобальные макросы с указанными ID.
groupids	ID/array	Возвращать только макросы узла сети, принадлежащие узлам сети или шаблонам из указанных групп узлов сети или групп шаблонов.
hostids	ID/array	Возвращать только макросы, принадлежащие указанным узлам сети или шаблонам.
hostmacroids	ID/array	Возвращать только макросы узла сети с указанными ID.
inherited	boolean	Если установлено значение `true`, возвращать только пользовательские макросы прототипа узла сети, унаследованные от шаблона.
selectHostGroups	query	Возвращать группы узлов сети, которым принадлежит макрос узла сети, в свойстве `hostgroups`. Используется только при получении макросов узла сети.
selectHosts	query	Возвращать узлы сети, которым принадлежит макрос узла сети, в свойстве `hosts`. Используется только при получении макросов узла сети.
selectTemplateGroups	query	Возвращать группы шаблонов, которым принадлежит макрос шаблона, в свойстве `templategroups`. Используется только при получении макросов шаблона.
selectTemplates	query	Возвращать шаблоны, которым принадлежит макрос узла сети, в свойстве `templates`. Используется только при получении макросов узла сети.
sortfield	string/array	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 для узла сети

Получите все макросы узла сети, определённые 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

Получите 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(  /  hostMacros)`

Этот метод позволяет обновлять существующие макросы хоста.

> **Note:**
> Этот метод доступен только пользователям с типом доступа *Администратор* и *Супер-администратор*. Разрешения на вызов метода можно отозвать в настройках ролей 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 являются необязательными. Будут обновлены только переданные свойства, все остальные останутся без изменений.

Возвращаемые значения

`(object)` Возвращает объект, содержащий идентификаторы обновлённых 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, который был создан правилом обнаружения

Преобразуйте созданный правилом обнаружения "автоматический" макрос в "ручной" и измените его значение на "new-value".

Запрос:

```
{
    "jsonrpc": "2.0",
    "method": "usermacro.update",
    "params": {
        "hostmacroid": "1",
        "value": "new-value",
        "automatic": "0"
    },
```

```
    "id": 1
}
```

Ответ:

```
{
    "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(    /     globalMacros)`

Этот метод позволяет обновлять существующие глобальные макросы.

Note:
Этот метод доступен только пользователю с типом доступа *Супер-администратор*. Разрешения на вызов метода можно отозвать в настройках ролей пользователя. Дополнительную информацию см. в разделе Роли пользователей.

Параметры

`(object/array)` Свойства глобального макроса, которые необходимо обновить.

Свойство `globalmacroid` должно быть определено для каждого глобального макроса, все остальные свойства являются необязательными. Будут обновлены только переданные свойства, все остальные останутся без изменений.

Возвращаемые значения

`(object)` Возвращает объект, содержащий идентификаторы обновлённых 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 низкоуровневого обнаружения.

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
    - Обнаружение операции переопределения правила LLD
    - Период операции переопределения правила LLD
    - История операции переопределения правила LLD
    - Тренды операции переопределения правила LLD
    - Важность 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

Свойство	Тип	Описание
delay	string	Интервал обновления правила LLD. Принимает секунды или единицу времени с суффиксом (например, 30s, 1m, 2h, 1d) и, при необходимости, один или несколько пользовательских интервалов, разделенных точкой с запятой. Пользовательские интервалы могут представлять собой сочетание гибких интервалов и интервалов по расписанию. Поддерживает пользовательские макросы. Если они используются, значение должно быть одним макросом. Несколько макросов или макросы, смешанные с текстом, не поддерживаются. Гибкие интервалы могут быть записаны как два макроса, разделенные 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` установлено в ”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` установлено в ”Zabbix agent (active)” (7) и `key_` не содержит ”mqtt.get”
hostid	ID	ID узла сети, которому принадлежит правило LLD. Поведение свойства: - *константа* - *обязательно* для операций создания
flags	integer	Происхождение правила обнаружения. Возможные значения: 1 - правило низкоуровневого обнаружения; 5 - правило низкоуровневого обнаружения, преобразованное из прототипа. Поведение свойства: - *только для чтения*
interfaceid	ID	ID интерфейса узла сети для правила LLD. Поведение свойства: - *обязательно*, если правило LLD принадлежит узлу сети и `type` установлено в ”Zabbix agent”, ”IPMI agent”, ”JMX agent” или ”SNMP agent” - *поддерживается*, если правило LLD принадлежит узлу сети и `type` установлено в ”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

Свойство	Тип	Описание
type	integer	Тип правила LLD. Возможные значения: 0 - Zabbix агент; 2 - Zabbix trapper; 3 - Простая проверка; 5 - Внутренний элемент Zabbix; 7 - Zabbix агент (активный); 10 - Внешняя проверка; 11 - Монитор базы данных; 12 - IPMI агент; 13 - SSH агент; 14 - TELNET агент; 16 - JMX агент; 18 - Зависимый элемент данных; 19 - HTTP агент; 20 - SNMP агент; 21 - Скрипт; 22 - Браузер; 23 - Вложенный. Этот тип разрешен только если родительским объектом правила является шаблон (предполагается, что он будет связан с прототипом узла сети при обнаружении) или обнаруженный узел сети (у которого правило LLD находится в корне). 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` установлено в ”HTTP agent” - *только для чтения* для унаследованных объектов
allow_traps	integer	Разрешить заполнение значения аналогично элементу данных trapper. Возможные значения: 0 - *(по умолчанию)* Не разрешать прием входящих данных; 1 - Разрешать прием входящих данных. Поведение свойства: - *поддерживается*, если `type` установлено в ”HTTP agent”

Свойство	Тип	Описание
authtype	integer	Метод аутентификации. Возможные значения, если `type` установлено в ”SSH agent”: 0 - *(по умолчанию)* пароль; 1 - открытый ключ. Возможные значения, если `type` установлено в ”HTTP agent”: 0 - *(по умолчанию)* нет; 1 - basic; 2 - NTLM; 3 - Kerberos; 4 - Digest. Поведение свойства: - *поддерживается*, если `type` установлено в ”SSH agent” или ”HTTP agent” - *только для чтения* для унаследованных объектов (если `type` установлено в ”HTTP agent”)
description	string	Описание правила LLD.
error	string	Текст ошибки, если возникают проблемы при обновлении значения правила LLD. 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` установлено в ”HTTP agent” - *только для чтения* для унаследованных объектов
headers	array	Массив заголовков, которые будут отправлены при выполнении HTTP-запроса. Поведение свойства: - *поддерживается*, если `type` установлено в ”HTTP agent” - *только для чтения* для унаследованных объектов
http_proxy	string	Строка подключения к HTTP(S) прокси. Поведение свойства: - *поддерживается*, если `type` установлено в ”HTTP agent” - *только для чтения* для унаследованных объектов
ipmi_sensor	string	Сенсор IPMI. Поведение свойства: - *обязательно*, если `type` установлено в ”IPMI agent” и `key_` не установлено в ”ipmi.get” - *поддерживается*, если `type` установлено в ”IPMI agent” - *только для чтения* для унаследованных объектов
jmx_endpoint	string	Пользовательская строка подключения JMX агент. По умолчанию: service:jmx:rmi:///jndi/rmi://{HOST.CONN}:{HOST.PORT}/jmxrmi Поведение свойства: - *поддерживается*, если `type` установлено в ”JMX agent”

Свойство	Тип	Описание
lifetime	string	Период времени, по истечении которого элементы данных, которые больше не обнаруживаются, будут удалены. Принимает секунды, единицу времени с суффиксом или пользовательский макрос. По умолчанию: 7d.
lifetime_type	integer	Сценарий удаления потерянных ресурсов LLD. Возможные значения: 0 - *(по умолчанию)* Удалить после достижения порога lifetime; 1 - Не удалять; 2 - Удалить немедленно.
enabled_lifetime	string	Период времени, по истечении которого элементы данных, которые больше не обнаруживаются, будут отключены. Принимает секунды, единицу времени с суффиксом или пользовательский макрос. По умолчанию: 0.
enabled_lifetime_type	integer	Сценарий отключения потерянных ресурсов LLD. Возможные значения: 0 - Отключить после достижения порога lifetime; 1 - Не отключать; 2 - *(по умолчанию)* Отключить немедленно.
master_itemid	ID	ID мастер-элемента данных. Правило обнаружения не может быть мастер-элементом данных для другого правила обнаружения. Поведение свойства: - *обязательно*, если `type` установлено в ”Dependent item” - *только для чтения* для унаследованных объектов
output_format	integer	Следует ли преобразовывать ответ в JSON. Возможные значения: 0 - *(по умолчанию)* Сохранять как есть; 1 - Преобразовать в JSON. Поведение свойства: - *поддерживается*, если `type` установлено в ”HTTP agent” - *только для чтения* для унаследованных объектов
params	string	Дополнительные параметры в зависимости от типа правила LLD: - выполняемый скрипт для правил LLD SSH и Telnet; - SQL-запрос для правил LLD монитора базы данных; - формула для вычисляемых правил LLD; - скрипт для правил LLD типа script и browser. Поведение свойства: - *обязательно*, если `type` установлено в ”Database monitor”, ”SSH agent”, ”TELNET agent”, ”Script” или ”Browser” - *только для чтения* для унаследованных объектов (если `type` установлено в ”Script” или ”Browser”)
parameters	object/array	Дополнительные параметры, если `type` установлено в ”Script” или ”Browser”. Массив объектов со свойствами `name` и `value`, где `name` должно быть уникальным. Поведение свойства: - *поддерживается*, если `type` установлено в ”Script” или ”Browser” - *только для чтения* для унаследованных объектов

Свойство	Тип	Описание
password	string	Пароль для аутентификации. Поведение свойства: - *обязательно*, если `type` установлено в ”JMX agent” и `username` задано - *поддерживается*, если `type` установлено в ”Simple check”, ”Database monitor”, ”SSH agent”, ”TELNET agent” или ”HTTP agent” - *только для чтения* для унаследованных объектов (если `type` установлено в ”HTTP agent”)
post_type	integer	Тип тела post-данных, хранящихся в свойстве `posts`. Возможные значения: 0 - *(по умолчанию)* Необработанные данные; 2 - JSON-данные; 3 - XML-данные. Поведение свойства: - *поддерживается*, если `type` установлено в ”HTTP agent” - *только для чтения* для унаследованных объектов
posts	string	Данные тела HTTP(S)-запроса. Поведение свойства: - *обязательно*, если `type` установлено в ”HTTP agent” и `post_type` установлено в ”JSON data” или ”XML data” - *поддерживается*, если `type` установлено в ”HTTP agent” и `post_type` установлено в ”Raw data” - *только для чтения* для унаследованных объектов
privatekey	string	Имя файла закрытого ключа. Поведение свойства: - *обязательно*, если `type` установлено в ”SSH agent” и `authtype` установлено в ”public key”
publickey	string	Имя файла открытого ключа. Поведение свойства: - *обязательно*, если `type` установлено в ”SSH agent” и `authtype` установлено в ”public key”
query_fields	array	Массив полей запроса, которые будут отправлены при выполнении HTTP-запроса. Поведение свойства: - *поддерживается*, если `type` установлено в ”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` установлено в ”HTTP agent” - *только для чтения* для унаследованных объектов

Свойство	Тип	Описание
retrieve_mode	integer	Какая часть ответа должна быть сохранена. Возможные значения, если `request_method` установлено в ”GET”, ”POST” или ”PUT”: 0 - *(по умолчанию)* Тело; 1 - Заголовки; 2 - Будут сохранены и тело, и заголовки. Возможные значения, если `request_method` установлено в ”HEAD”: 1 - Заголовки. Поведение свойства: - *поддерживается*, если `type` установлено в ”HTTP agent” - *только для чтения* для унаследованных объектов
snmp_oid	string	SNMP OID. Поведение свойства: - *обязательно*, если `type` установлено в ”SNMP agent” - *только для чтения* для унаследованных объектов
ssl_cert_file	string	Путь к файлу открытого SSL-ключа. Поведение свойства: - *поддерживается*, если `type` установлено в ”HTTP agent” - *только для чтения* для унаследованных объектов
ssl_key_file	string	Путь к файлу закрытого SSL-ключа. Поведение свойства: - *поддерживается*, если `type` установлено в ”HTTP agent” - *только для чтения* для унаследованных объектов
ssl_key_password	string	Пароль для файла SSL-ключа. Поведение свойства: - *поддерживается*, если `type` установлено в ”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	Статус правила LLD. Возможные значения: 0 - *(по умолчанию)* включенное правило LLD; 1 - отключенное правило LLD.
status_codes	string	Диапазоны требуемых HTTP-кодов состояния, разделенные запятыми. Также поддерживает пользовательские макросы как часть списка, разделенного запятыми. Пример: 200,200-{$M},{$M},200-400 Поведение свойства: - *поддерживается*, если `type` установлено в ”HTTP agent” - *только для чтения* для унаследованных объектов
templateid	ID	ID родительского правила LLD шаблона. Поведение свойства: - *только для чтения*

Свойство	Тип	Описание
timeout	string	Тайм-аут запроса опроса данных элемента данных. Принимает секунды или единицу времени с суффиксом (например, 30s, 1m). Также принимает пользовательские макросы. Диапазон возможных значений: 1-600s. По умолчанию: "" - использовать настройки прокси/глобальные настройки. Поведение свойства: - *поддерживается*, если `type` установлено в "Zabbix agent" (0), "Simple check" (3) и `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) и `snmp_oid` начинается с "`walk[`" или "`get[`", "Script" (21), "Browser" (22) - *только для чтения* для унаследованных объектов
trapper_hosts	string	Разрешенные узлы сети. Поведение свойства: - *поддерживается*, если `type` установлено в "Zabbix trapper", или если `type` установлено в "HTTP agent" и `allow_traps` установлено в "Allow to accept incoming data"
username	string	Имя пользователя для аутентификации. Поведение свойства: - *обязательно*, если `type` установлено в "SSH agent", "TELNET agent", или если `type` установлено в "JMX agent" и `password` задано - *поддерживается*, если `type` установлено в "Simple check", "Database monitor" или "HTTP agent" - *только для чтения* для унаследованных объектов (если `type` установлено в "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
verify_host	integer	Следует ли проверять, что имя узла сети для подключения совпадает с именем в сертификате узла сети. Возможные значения: 0 - *(по умолчанию)* Не проверять; 1 - Проверять. Поведение свойства: - *поддерживается*, если `type` установлено в "HTTP agent" - *только для чтения* для унаследованных объектов
verify_peer	integer	Следует ли проверять подлинность сертификата узла сети. Возможные значения: 0 - *(по умолчанию)* Не проверять; 1 - Проверять. Поведение свойства: - *поддерживается*, если `type` установлено в "HTTP agent" - *только для чтения* для унаследованных объектов

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 правила LLD

Объект фильтра 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 - And/Or; 1 - And; 2 - Or; 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 правила LLD

Объект условия фильтра правила LLD определяет отдельную проверку, выполняемую для значения макроса LLD. Он имеет следующие свойства:

Свойство	Тип	Описание
macro	string	Макрос LLD, для которого выполняется проверка. Поведение свойства: - *обязательно*
value	string	Значение для сравнения. Поведение свойства: - *обязательно*, если `operator` установлен в ”совпадает с регулярным выражением” или ”не совпадает с регулярным выражением”
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 последующем их запросе. Поведение свойства: - *обязательно*, если `evaltype` объекта фильтра правила LLD установлен в ”пользовательское выражение”
operator	integer	Оператор условия. Возможные значения: 8 - *(по умолчанию)* совпадает с регулярным выражением; 9 - не совпадает с регулярным выражением; 12 - существует; 13 - не существует.

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:

Свойство	Type	Описание
lld_macro	string	Макрос LLD. Поведение свойства: - *обязательно*
path	string	Селектор значения, которое будет присвоено соответствующему макросу. Поведение свойства: - *обязательно*

Предобработка правила LLD

Объект предобработки правила LLD имеет следующие свойства.

Свойство	Тип	Описание
type	integer	Тип опции предобработки. Возможные значения: 5 - Регулярное выражение; 11 - XML XPath; 12 - JSONPath; 14 - Соответствует регулярному выражению; 15 - Не соответствует регулярному выражению; 16 - Проверка на ошибку в JSON; 17 - Проверка на ошибку в XML; 20 - Отбрасывать неизменившееся с интервалом; 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: - *обязательно*
params	string	Дополнительные параметры, используемые опцией предобработки. Несколько параметров разделяются символом новой строки (\n). Поведение свойства: - *обязательно*, если `type` установлен в ”Регулярное выражение” (5), ”XML XPath” (11), ”JSONPath” (12), ”Соответствует регулярному выражению” (14), ”Не соответствует регулярному выражению” (15), ”Проверка на ошибку в JSON” (16), ”Проверка на ошибку в XML” (17), ”Отбрасывать неизменившееся с интервалом” (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, используемого в случае сбоя шага предобработки. Возможные значения: 0 - Сообщение об ошибке задаётся сервером Zabbix; 1 - Отбросить значение; 2 - Установить пользовательское значение; 3 - Установить пользовательское сообщение об ошибке. Поведение свойства: - *обязательно*, если `type` установлен в ”Регулярное выражение” (5), ”XML XPath” (11), ”JSONPath” (12), ”Соответствует регулярному выражению” (14), ”Не соответствует регулярному выражению” (15), ”Проверка на ошибку в JSON” (16), ”Проверка на ошибку 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: - *обязательно*, если `error_handler` установлен в ”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	шаблон[1]	вывод[2]		0, 1, 2, 3
11	XML XPath	путь[3]			0, 1, 2, 3
12	JSONPath	путь[3]			0, 1, 2, 3
14	Соответствует регулярному выражению	шаблон[1]			0, 1, 2, 3
15	Не соответствует регулярному выражению	шаблон[1]			0, 1, 2, 3
16	Проверка на ошибку в JSON	путь[3]			0, 1, 2, 3
17	Проверка на ошибку в XML	путь[3]			0, 1, 2, 3
20	Отбрасывать неизменившееся с интервалом	секунды[4, 5]			
21	JavaScript	скрипт[2]			
23	Prometheus в JSON	шаблон[5, 6]			0, 1, 2, 3
24	CSV в JSON	символ[2]	символ[2]	0,1	0, 1, 2, 3

Тип предобработки	Название	Параметр 1	Параметр 2	Параметр 3	Поддерживаемые обработчики ошибок
25	Замена	строка поиска[2]	замена[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, следующий 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", разделённых символом новой строки.

Переопределения правил LLD

Объект переопределений 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*

Свойство	Type	Описание
stop	integer	Остановить обработку следующих переопределений при совпадении. Возможные значения: 0 - *(по умолчанию)* 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, при совпадении с которыми для обнаруженного объекта применяется переопределение. Он имеет следующие свойства:

Свойство	Тип	Описание
conditions	object/array	Набор условий фильтра переопределения, используемых для сопоставления обнаруженных объектов. Условия будут отсортированы в порядке их размещения в формуле. Поведение свойства: - *обязательно*
evaltype	integer	Метод вычисления условий фильтра переопределения. Возможные значения: 0 - And/Or; 1 - And; 2 - Or; 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` установлено в ”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:

Свойство	Type	Описание
macro	string	Макрос LLD, для которого выполняется проверка. Property behavior: - *required*
value	string	Значение для сравнения. Property behavior: - *required* если `operator` установлен в ”совпадает с регулярным выражением” или ”не совпадает с регулярным выражением”
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* если `evaltype` объекта LLD rule override filter object установлен в ”пользовательское выражение”
operator	integer	Оператор условия. Возможные значения: 8 - *(по умолчанию)* совпадает с регулярным выражением; 9 - не совпадает с регулярным выражением; 12 - существует; 13 - не существует.

Операция переопределения правила LLD

Операция переопределения правила LLD представляет собой комбинацию условий и действий, выполняемых 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; 4 - Прототип обнаружения. Поведение свойства: - *обязательно*
operator	integer	Условие переопределения operator. 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).

Свойство	Type	Описание
opperiod	object	Объект периода операции переопределения (интервал 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.
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 правила LLD

Статус операции переопределения правила LLD, который устанавливается для обнаруженного объекта. Имеет следующие свойства:

Свойство	Тип	Описание
status	integer	Переопределяет статус для выбранного объекта. Возможные значения: 0 - Создать включённым; 1 - Создать отключённым. Поведение свойства: - *обязательно*

Обнаружение операции переопределения правила LLD

Статус обнаружения операции переопределения 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 правила LLD

Период операции переопределения правила LLD — это значение интервала обновления, которое задается для обнаруженного элемента данных. Он имеет следующие свойства:

Свойство	Тип	Описание
delay	string	Переопределяет интервал обновления прототипа элемента данных. Принимает секунды или единицу времени с суффиксом (например, 30s, 1m, 2h, 1d), а также, при необходимости, один или несколько пользовательских интервалов, разделенных точкой с запятой. Пользовательские интервалы могут представлять собой сочетание гибких интервалов и интервалов по расписанию. Принимает пользовательские макросы или макросы LLD. Если они используются, значение должно быть одним макросом. Несколько макросов или макросы в сочетании с текстом не поддерживаются. Гибкие интервалы могут быть записаны как два макроса, разделенные прямой 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: *- required*

История операции переопределения правила LLD

Значение истории операции переопределения правила LLD, которое задается для обнаруженного элемента данных. Оно 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*

Тренды операции переопределения правила LLD

Значение трендов операции переопределения правила LLD, которое устанавливается для обнаруженного элемента данных. Оно имеет следующие свойства:

Свойство	Type	Описание
trends	string	Переопределяет тренды прототипа элемента данных, то есть единицу времени, определяющую, как долго должны храниться данные трендов. Также принимает пользовательский макрос и макрос LLD. Поведение свойства: *- обязательно*

Серьезность операции переопределения правила LLD

Значение серьезности операции переопределения правила LLD, которое устанавливается для обнаруженного триггера. Оно имеет следующие свойства:

Свойство	Type	Описание
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: - *обязательно*

Тег операции переопределения правила LLD

Объект тега операции переопределения 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 правила LLD

Объект шаблона операции переопределения 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 правила LLD

Значение режима инвентаризации операции переопределения правила LLD, которое устанавливается для обнаруженного узла сети. Оно 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*

discoveryrule.create

Описание

```
object discoveryrule.create(    /     lldRules)
```

Этот метод позволяет создавать новые правила LLD.

Note:
Этот метод доступен только пользователям с типом доступа *Администратор* и *Супер-администратор*. Разрешения на вызов метода можно отозвать в настройках ролей пользователя. Дополнительную информацию см. в разделе Роли пользователей.

Параметры

`(object/array)` Правила LLD для создания.

В дополнение к стандартным свойствам правила LLD, метод принимает следующие параметры.

Параметр	Тип	Описание
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 сгруппированы с использованием логического оператора ”и”.

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 пользовательского фильтра выражения

Создайте правило LLD с фильтром, который будет использовать пользовательское выражение для вычисления условий. Правило LLD должно обнаруживать только те объекты, значение макроса "{#MACRO1}" у которых соответствует обоим регулярным выражениям "regex1" и "regex2", а значение "{#MACRO2}" соответствует либо "regex3", либо "regex4". Идентификаторы формулы "A", "B", "C" и "D" были 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    ",
        "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 предобработкой

Запрос:

```
{
    "jsonrpc": "2.0",
    "method": "discoveryrule.create",
    "params": {
        "name": "                      ",
        "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": "                     ",
        "key_": "lld.with.overrides",
        "hostid": "10001",
        "type": 0,
        "delay": "60s",
        "interfaceid": "1155",
        "overrides": [
            {
                "name": "             MySQL",
```

```
            "step": "1",
            "stop": "1",
            "filter": {
                "evaltype": "2",
                "conditions": [
                    {
                        "macro": "{#UNIT.NAME}",
                        "operator": "8",
                        "value": "^mysqld\\.service$"
                    },
                    {
                        "macro": "{#UNIT.NAME}",
                        "operator": "8",
                        "value": "^mariadb\\.service$"
                    }
                ]
            },
            "operations": [
                {
                    "operationobject": "3",
                    "operator": "2",
                    "value": "             ",
                    "opstatus": {
                        "status": "0"
                    },
                    "optemplate": [
                        {
                            "templateid": "10170"
                        }
                    ],
                    "optag": [
                        {
                            "tag": "database",
                            "value": "mysql"
                        }
                    ]
                }
            ]
        },
        {
            "name": "             PostgreSQL",
            "step": "2",
            "stop": "1",
            "filter": {
                "evaltype": "0",
                "conditions": [
                    {
                        "macro": "{#UNIT.NAME}",
                        "operator": "8",
                        "value": "^postgresql\\.service$"
                    }
                ]
            },
            "operations": [
                {
                    "operationobject": "3",
                    "operator": "2",
                    "value": "             ",
                    "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 со скриптом

Создайте простой сбор данных с помощью правила LLD со скриптом.

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 пользовательским 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 удалять правила LLD.

> **Note:**
> Этот метод доступен только для типов пользователей *Администратор* и *Супер-администратор*. Разрешения на вызов метода могут быть отозваны в настройках роли пользователей. Смотрите Роли пользователей для получения дополнительной информации.

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)`

Этот метод позволяет получать правила LLD 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 смотрите в разделе Роли пользователей.

Параметры

`(object)` Параметры, определяющие желаемый вывод.

Метод поддерживает следующие параметры.

Параметр	Тип	Описание
itemids	ID/array	Возвращать только правила LLD с указанными ID.
groupids	ID/array	Возвращать только правила LLD, принадлежащие узлам сети из указанных групп.
hostids	ID/array	Возвращать только правила LLD, принадлежащие указанным узлам сети.
inherited	boolean	Если установлено в `true`, возвращать только правила LLD, унаследованные от шаблона.
interfaceids	ID/array	Возвращать только правила LLD, использующие указанные интерфейсы узла сети.
monitored	boolean	Если установлено в `true`, возвращать только включенные правила LLD, принадлежащие отслеживаемым узлам сети.
templated	boolean	Если установлено в `true`, возвращать только правила LLD, принадлежащие шаблонам.
templateids	ID/array	Возвращать только правила LLD, принадлежащие указанным шаблонам.

Параметр	Тип	Описание
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, принадлежащими правилу LLD. Поддерживает `count`.
selectFilter	query	Возвращать свойство `filter` с данными фильтра, используемого правилом LLD.
selectGraphs	query	Возвращает свойство `graphs` с прототипами графиков, принадлежащими правилу LLD. Поддерживает `count`.
selectHostPrototypes	query	Возвращать свойство `hostPrototypes` с прототипами узлов сети, принадлежащими правилу LLD. Поддерживает `count`.
selectHosts	query	Возвращать свойство `hosts` с массивом узлов сети, которым принадлежит правило LLD.
selectItems	query	Возвращать свойство `items` с прототипами элементов данных, принадлежащими правилу LLD. Поддерживает `count`.
selectTriggers	query	Возвращать свойство `triggers` с прототипами триггеров, принадлежащими 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 предобработки правила LLD.
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 указанному фильтру. Принимает объект, где ключи — это имена свойств, а значения — либо одиночное значение, либо массив значений для сопоставления. Не поддерживает свойства типа данных `text` data type. Поддерживает дополнительные свойства: `host` - техническое имя узла сети, которому принадлежит правило 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	

Параметр	Тип	Описание
search	object	
searchByAny	boolean	
searchWildcardsEnabled	boolean	
sortorder	string/array	
startSearch	boolean	

Возвращаемые значения

`(integer/array)` Возвращает:

- массив объектов
- количество полученных объектов, если был использован параметр `countOutput`.

Примеры

Получение правил обнаружения с узла сети

Получите 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 условия его фильтра. Фильтр использует тип вычисления "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

Получить одно правило LLD, у которого имеются различные настройки 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

Исходный код

CDiscoveryRule::get() в *ui/include/classes/api/services/CDiscoveryRule.php*.

discoveryrule.update

Описание

`object discoveryrule.update(  /  lldRules)`

Этот метод позволяет обновлять существующие правила LLD.

Note:
Этот метод доступен только пользователям с типом доступа *Администратор* и *Супер-администратор*. Разрешения на вызов метода можно отозвать в настройках ролей пользователя. Дополнительную информацию см. в разделе Роли пользователей.

Параметры

`(object/array)` свойства правила LLD, которые должны быть обновлены.

Свойство `itemid` должно быть определено для каждого правила LLD, все остальные свойства являются необязательными. Будут обновлены только переданные свойства, все остальные останутся без изменений.

В дополнение к стандартным свойствам правила LLD, метод принимает следующие параметры.

Параметр	Тип	Описание
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 в правило LLD

Добавьте фильтр так, 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 траппинга

Отключите траппинг LLD 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 предобработки правила LLD

Обновите правило LLD с правилом предобработки “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 другим скриптом 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 сущность, которая больше не обнаруживается, через 12 часов и удалять её 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 обнаружения низкого уровня см. API правил 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 сетевого обнаружения. Он имеет следующие свойства.

Свойство	Тип	Описание
druleid	ID	ID правила обнаружения. Поведение свойства: - *только для чтения* - *обязательно* для операций обновления
iprange	string	Один или несколько диапазонов IP-адресов для проверки, разделённых запятыми. Дополнительную информацию о поддерживаемых форматах диапазонов IP-адресов смотрите в разделе настройки сетевого обнаружения. Поведение свойства: - *обязательно* для операций создания

Свойство	Тип	Описание
name	string	Имя правила обнаружения. Поведение свойства: - *обязательно* для операций создания
delay	string	Интервал выполнения правила обнаружения. Допускаются секунды или единица времени с суффиксом (например, 30s, 1m, 2h, 1d), либо пользовательский макрос. По умолчанию: 1h.
proxyid	ID	ID прокси, используемого для обнаружения.
status	integer	Включено ли правило обнаружения. Возможные значения: 0 - *(по умолчанию)* включено; 1 - отключено.
concurrency_max	integer	Максимальное количество одновременных проверок на одно правило 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 какие-либо проблемы. Поведение свойства: - *только для чтения*

drule.create

Описание

`object drule.create(  /  discoveryRules)`

Этот метод позволяет создавать новые правила обнаружения.

> **Note:**
> Этот метод доступен только пользователям с типом доступа *Администратор* и *Супер-администратор*. Разрешения на вызов метода можно отозвать в настройках ролей 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	Type	Описание
dchecks	array	Проверки обнаружения для создания для правила обнаружения. Поведение параметра: - *обязательный*

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 находить 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(    discoveryRuleIds)`

Этот метод позволяет удалять правила обнаружения.

> **Note:**
> Этот метод доступен только пользователям с типом доступа *Администратор* и *Супер-администратор*. Разрешения на вызов метода можно отозвать в настройках ролей 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

Удалите два правила 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(           )`

Этот метод позволяет получать правила обнаружения в соответствии с заданными параметрами.

> **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)` Параметры, определяющие желаемый вывод.

Метод поддерживает следующие параметры.

Параметр	Тип	Описание
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`.

Параметр	Тип	Описание
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

Получение всех правил 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

```
drule.update(object/array discoveryRules)
```

Этот метод позволяет обновлять существующие правила обнаружения.

Note:
Этот метод доступен только пользователям с ролью *Admin* или *Super admin*. Разрешения на вызов метода можно отозвать в настройках роли пользователя. См. Роли пользователей для получения дополнительной информации.

Параметры

`(object/array)` Свойства правил обнаружения, которые необходимо обновить.

Свойство `druleid` должно быть определено для каждого правила обнаружения, все остальные свойства являются необязательными. Будут обновлены только переданные свойства, все остальные останутся без изменений.

В дополнение к стандартным свойствам правила обнаружения, метод принимает следующие параметры.

Параметр	Тип	Описание
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 оповещения
- 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.

Свойство	Тип	Описание
eventid	ID	ID события проблемы.
source	integer	Тип события проблемы. Возможные значения: 0 - событие, созданное триггером; 3 - внутреннее событие; 4 - событие, созданное при обновлении состояния сервиса.
object	integer	Тип объекта, связанного с событием проблемы. Возможные значения, если `source` установлено в ”событие, созданное триггером”: 0 - триггер. Возможные значения, если `source` установлено в ”внутреннее событие”: 0 - триггер; 4 - элемент данных; 5 - правило LLD. Возможные значения, если `source` установлено в ”событие, созданное при обновлении состояния сервиса”: 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.

Свойство	Тип	Описание
r_ns	integer	Наносекунды в момент создания события восстановления.
cause_eventid	ID	ID причинного события.
correlationid	ID	ID правила корреляции, если это событие было восстановлено глобальным правилом корреляции.
userid	ID	ID пользователя, который закрыл проблему (если проблема была закрыта вручную).
name	string	Имя разрешённой проблемы.
acknowledged	integer	Состояние подтверждения проблемы. Возможные значения: 0 - не подтверждена; 1 - подтверждена.
severity	integer	Текущая важность проблемы. Возможные значения: 0 - не классифицирована; 1 - информация; 2 - предупреждение; 3 - средняя; 4 - высокая; 5 - катастрофическая.
suppressed	integer	Подавлена ли проблема. Возможные значения: 0 - проблема в нормальном состоянии; 1 - проблема подавлена.
opdata	string	Операционные данные с раскрытыми макросами.
urls	array	Активные URL-адреса способа оповещения.

URL типа медиа

Объект URL типа медиа имеет следующие свойства.

Свойство	Type	Описание
name	string	Имя URL, определённого для типа медиа.
url	string	Значение URL, определённого для типа медиа.

Результаты будут содержать записи только для активных типов медиа с включённой записью меню событий. Макросы, используемые в свойствах, будут раскрыты, но если одно из свойств содержит нераскрытый макрос, оба свойства будут исключены из результатов. Поддерживаемые макросы перечислены в разделе *Поддерживаемые макросы*.

Тег проблемы

Объект тега проблемы имеет следующие свойства.

Свойство	Type	Описание
tag	string	Имя тега проблемы.
value	string	Значение тега проблемы.

problem.get

Описание

`integer/array problem.get(object parameters)`

Этот метод позволяет получать проблемы в соответствии с заданными параметрами.

Этот метод предназначен для получения неустранённых проблем. Также, если указано, можно дополнительно получать недавно устранённые проблемы. Период, который определяет, насколько давно считается "недавно", задаётся в *Administration* → *General*. Проблемы, которые были устранены до этого периода, не хранятся в таблице проблем. Чтобы получить проблемы, которые были устранены раньше, используйте метод `event.get`.

Attention:
Этот метод может возвращать проблемы удалённой сущности, если эти проблемы ещё не были удалены сборщиком устаревших данных.

Note:
Этот метод доступен пользователям любого типа. Права на вызов метода могут быть отозваны в настройках роли пользователя. Подробнее см. в разделе User roles.

Параметры

`(object)` Параметры, определяющие желаемый вывод.

Метод поддерживает следующие параметры.

Параметр	Тип	Описание
eventids	ID/array	Возвращать только проблемы с указанными ID.
groupids	ID/array	Возвращать только проблемы, созданные объектами, принадлежащими указанным группам узлов сети.
hostids	ID/array	Возвращать только проблемы, созданные объектами, принадлежащими указанным узлам сети.
objectids	ID/array	Возвращать только проблемы, созданные указанными объектами.
source	integer	Возвращать только проблемы указанного типа. Список поддерживаемых типов событий смотрите на странице объекта события проблемы. По умолчанию: 0 — проблема, созданная триггером.
object	integer	Возвращать только проблемы, созданные объектами указанного типа. Список поддерживаемых типов объектов смотрите на странице объекта события проблемы. 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 карты в качестве битовой 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 важностями событий. Применяется только если object — триггер.
evaltype	integer	Метод вычисления тегов. Возможные значения: 0 - *(по умолчанию)* And/Or; 2 - Or.

Параметр	Тип	Описание
tags	array	Возвращать только проблемы с указанными тегами. Формат: `[{"tag": "<tag>", "value": "<value>", "operator": "<operator>"}, ...]`. Пустой массив возвращает все проблемы. Возможные значения operator: 0 - *(по умолчанию)* Содержит; 1 - Равно; 2 - Не содержит; 3 - Не равно; 4 - Существует; 5 - Не существует.
recent	boolean	Если установлено в `true`, возвращать только недавно устраненные проблемы (зависит от `ok_period`).
eventid_from	string	Возвращать только проблемы с ID больше или равными указанному ID.
eventid_till	string	Возвращать только проблемы с ID меньше или равными указанному ID.
time_from	timestamp	Возвращать только проблемы, созданные после указанного времени или в указанный момент времени.
time_till	timestamp	Возвращать только проблемы, созданные до указанного времени или в указанный момент времени.
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, обновившего событие; `eventid` - `(ID)` ID обновленного события; `clock` - `(timestamp)` время обновления события; `message` - `(string)` текст сообщения; `action` - `(integer)` тип действия обновления (см. `event.acknowledge`); `old_severity` - `(integer)` важность события до этого действия обновления; `new_severity` - `(integer)` важность события после этого действия обновления; `suppress_until` - `(timestamp)` время, до которого событие будет подавлено; `taskid` - `(ID)` ID задачи, если для текущего события выполняется изменение ранга. 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, подавившего 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 типа данных `text` data type.
sortfield	string/array	Сортировать результат по указанным свойствам. Возможные значения: `eventid`.

Параметр	Тип	Описание
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 событий проблем триггера

Получение недавних событий 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   localhost              5    ",
            "acknowledged": "1",
            "severity": "3",
            "cause_eventid": "0",
            "opdata": "",
```

```
            "acknowledges": [
                {
                    "acknowledgeid": "14443",
                    "userid": "1",
                    "eventid": "1245463",
                    "clock": "1472457281",
                    "message": "         ",
                    "action": "6",
                    "old_severity": "0",
                    "new_severity": "0",
                    "suppress_until": "1472511600",
                    "taskid": "0"
                }
            ],
            "suppression_data": [
                {
                    "maintenanceid": "15",
                    "suppress_until": "1472511600",
                    "userid": "0"
                }
            ],
            "suppressed": "1",
            "tags": [
                {
                    "tag": "test-tag",
                    "value": "test-value"
                }
            ]
        }
    ],
    "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",
            "opdata": "",
            "acknowledges": [
                {
                    "userid": "10",
                    "action": "2"
                }
            ],
            "suppressed": "0"
        }
    ],
    "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.

Свойство	Тип	Описание
dcheckid	ID	ID проверки обнаружения.
druleid	ID	ID правила обнаружения, к которому относится проверка.
key_	string	Ключ элемента данных (если `type` установлен в ”Zabbix agent”) или SNMP OID (если `type` установлен в ”SNMPv1 agent”, ”SNMPv2 agent” или ”SNMPv3 agent”). Поведение свойства: - *обязательно* если `type` установлен в ”Zabbix agent”, ”SNMPv1 agent”, ”SNMPv2 agent” или ”SNMPv3 agent”

Свойство	Тип	Описание
ports	string	Один или несколько диапазонов портов для проверки, разделённых запятыми. По умолчанию: 0. Поведение свойства: - *поддерживается* если `type` установлен в ”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) или ”Telnet” (15)
snmp_community	string	SNMP community. Поведение свойства: - *обязательно* если `type` установлен в ”SNMPv1 agent” или ”SNMPv2 agent”
snmpv3_authpassphrase	string	Парольная фраза аутентификации. Поведение свойства: - *поддерживается* если `type` установлен в ”SNMPv3 agent” и `snmpv3_securitylevel` установлен в ”authNoPriv” или ”authPriv”
snmpv3_authprotocol	integer	Протокол аутентификации. Возможные значения: 0 - *(по умолчанию)* MD5; 1 - SHA1; 2 - SHA224; 3 - SHA256; 4 - SHA384; 5 - SHA512. Поведение свойства: - *поддерживается* если `type` установлен в ”SNMPv3 agent” и `snmpv3_securitylevel` установлен в ”authNoPriv” или ”authPriv”
snmpv3_contextname	string	Имя контекста SNMPv3. Поведение свойства: - *поддерживается* если `type` установлен в ”SNMPv3 agent”
snmpv3_privpassphrase	string	Парольная фраза приватности. Поведение свойства: - *поддерживается* если `type` установлен в ”SNMPv3 agent” и `snmpv3_securitylevel` установлен в ”authPriv”
snmpv3_privprotocol	integer	Протокол приватности. Возможные значения: 0 - *(по умолчанию)* DES; 1 - AES128; 2 - AES192; 3 - AES256; 4 - AES192C; 5 - AES256C. Поведение свойства: - *поддерживается* если `type` установлен в ”SNMPv3 agent” и `snmpv3_securitylevel` установлен в ”authPriv”

Свойство	Тип	Описание
snmpv3_securitylevel	string	Уровень безопасности. Возможные значения: 0 - noAuthNoPriv; 1 - authNoPriv; 2 - authPriv. Поведение свойства: - *поддерживается* если `type` установлен в ”SNMPv3 agent”
snmpv3_securityname	string	Имя безопасности. Поведение свойства: - *поддерживается* если `type` установлен в ”SNMPv3 agent”
type	integer	Тип проверки. Возможные значения: 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. Поведение свойства: - *обязательно*
uniq	integer	Следует ли использовать эту проверку как критерий уникальности устройства. Для правила обнаружения можно настроить только одну уникальную проверку. Возможные значения: 0 - *(по умолчанию)* не использовать эту проверку как критерий уникальности; 1 - использовать эту проверку как критерий уникальности. Поведение свойства: - *поддерживается* если `type` установлен в ”Zabbix agent”, ”SNMPv1 agent”, ”SNMPv2 agent” или ”SNMPv3 agent”
host_source	integer	Источник имени узла сети. Возможные значения: 1 - *(по умолчанию)* DNS; 2 - IP; 3 - значение обнаружения этой проверки.
name_source	integer	Источник видимого имени. Возможные значения: 0 - *(по умолчанию)* не указано; 1 - DNS; 2 - IP; 3 - значение обнаружения этой проверки.

Свойство	Тип	Описание
allow_redirect	integer	Разрешить ситуацию, когда целевой объект, проверяемый с помощью ICMP ping, отвечает с другого IP-адреса. Возможные значения: 0 - *(по умолчанию)* считать перенаправленные ответы так, как если бы целевой узел сети был недоступен (ошибка); 1 - считать перенаправленные ответы так, как если бы целевой узел сети был доступен (успех). Поведение свойства: - *поддерживается* если `type` установлен в ”ICMP ping”

dcheck.get

Описание

`integer/array dcheck.get(parameters)`

Метод позволяет получить проверки обнаружения по заданным параметрам.

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)` Параметры, определяющие желаемый вывод.

Метод поддерживает следующие параметры.

Параметр	Тип	Описание
dcheckids	ID/array	Возвращать только проверки обнаружения с указанными ID.
druleids	ID/array	Возвращать только проверки обнаружения, принадлежащие указанным правилам обнаружения.
dserviceids	ID/array	Возвращать только проверки обнаружения, которые обнаружили указанные обнаруженные сервисы.
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

Получение всех проверок обнаружения, используемых 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 прокси

Следующие объекты напрямую связаны с API `proxy`.

Прокси

Объект прокси имеет следующие свойства.

Свойство	Тип	Описание
proxyid	ID	ID прокси. Поведение свойства: - *только для чтения* - *обязательно* для операций обновления
name	string	Имя прокси. Поведение свойства: - *обязательно* для операций создания
proxy_groupid	ID	ID 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, когда прокси в последний раз подключался 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 Zabbix proxy — пассивный
port	string	Номер порта для подключения. Поддерживаются пользовательские макросы. По умолчанию: 10051. Поведение свойства: - *поддерживается*, если режим работы Zabbix proxy — пассивный
allowed_addresses	string	IP-адреса или DNS-имена активного Zabbix proxy, разделённые запятыми.

Свойство	Тип	Описание
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 битовой маски; допустима 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 (среди autoregistration, hosts и proxies). Не включайте конфиденциальную информацию в идентификатор PSK, так как он передаётся по сети в незашифрованном виде, чтобы сообщить получателю, какой PSK использовать. Поведение свойства: - *только для записи* - *обязательно*, если `tls_connect` установлено в ”PSK” или `tls_accept` содержит бит ”PSK”
tls_psk	string	Предварительно общий ключ (PSK); должен содержать не менее 32 шестнадцатеричных цифр. Поведение свойства: - *только для записи* - *обязательно*, если `tls_connect` установлено в ”PSK” или `tls_accept` содержит бит ”PSK”
custom_timeouts	integer	Следует ли переопределять 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. Допускаются секунды или единица времени с суффиксом (например, 30s, 1m). Также допускаются пользовательские макросы. Диапазон возможных значений: 1-600s. По умолчанию: ””. Поведение свойства: - *обязательно*, если `custom_timeouts` установлено в 1.
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		

Свойство	Тип	Описание
version	integer	Версия прокси. Трёхчастный номер версии Zabbix, где 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*
compatibility	integer	Версия прокси по сравнению с версией Zabbix server. Возможные значения: 0 — Не определено; 1 — Текущая версия (прокси и сервер имеют одинаковую мажорную версию); 2 — Устаревшая версия (версия прокси старше версии сервера, но частично поддерживается); 3 — Неподдерживаемая версия (версия прокси старше версии предыдущего LTS-релиза сервера или мажорная версия сервера старше мажорной 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*

proxy.create

Описание

`object proxy.create( / proxies)`

Этот метод позволяет создавать новые прокси.

Note:
Этот метод доступен только пользователю с типом доступа *Супер-администратор*. Разрешения на вызов метода можно отозвать в настройках ролей пользователя. Дополнительную информацию см. в разделе Роли пользователей.

Параметры

`(object/array)` Прокси для создания.

В дополнение к стандартным свойствам прокси, метод принимает следующие параметры.

Параметр	Тип	Описание
hosts	array	Узлы сети, которые должны мониториться прокси. Если узел сети уже мониторится другим прокси, он будет переназначен текущему прокси. Для узлов сети должно быть определено только свойство `hostid`.

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
}
```

См. также

- Узел сети
- Группа прокси

Источник

CProxy::create() в *ui/include/classes/api/services/CProxy.php*.

proxy.delete

Описание

`object proxy.delete(    proxies)`

Этот метод позволяет удалять прокси.

> **Note:**
> Этот метод доступен только пользователю с типом доступа *Супер-администратор*. Разрешения на вызов метода можно отозвать в настройках ролей пользователя. Дополнительную информацию см. в разделе Роли пользователей.

Параметры

`(    )` ID удаляемых прокси.

Возвращаемые значения

`(object)` Возвращает объект, содержащий идентификаторы удалённых прокси в свойстве `proxyids`.

Примеры

Удаление нескольких прокси

Удалить два 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(  )`

Этот метод позволяет получать прокси в соответствии с заданными параметрами.

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)` Параметры, определяющие желаемый вывод.

Метод поддерживает следующие параметры.

Параметр	Тип	Описание
proxyids	ID/array	Возвращать только прокси с указанными ID.
proxy_groupids	ID/array	Возвращать только прокси, принадлежащие указанным группам прокси.
selectAssignedHosts	query	Возвращать свойство `assignedHosts` с узлами сети, назначенными прокси. Поддерживает `count`.
selectHosts	query	Возвращать свойство `hosts` с узлами сети, которые мониторит 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	Эти параметры описаны 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 одно из следующего:

- массив объектов;
- количество полученных объектов, если был использован параметр `countOutput`.

Примеры

Получение всех прокси

Получите 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(  /  proxies)`

Этот метод позволяет обновлять существующие прокси.

> **Note:**
> Этот метод доступен только пользователю с типом доступа *Супер-администратор*. Разрешения на вызов метода можно отозвать в настройках ролей пользователя. Дополнительную информацию см. в разделе Роли пользователей.

Параметры

`(object/array)` Свойства прокси, которые требуется обновить.

Свойство `proxyid` должно быть определено для каждого прокси, все остальные свойства являются необязательными. Будут обновлены только переданные свойства, все остальные останутся без изменений.

В дополнение к стандартным свойствам прокси, метод принимает следующие параметры.

Параметр	Type	Описание
hosts	array	Узлы сети, которые должны мониториться прокси. Если узел сети уже мониторится другим прокси, он будет переназначен текущему прокси. Для узлов сети должно быть определено только свойство `hostid`.

Возвращаемые значения

`(object)` Возвращает объект, содержащий идентификаторы обновлённых прокси в свойстве `proxyids`.

Примеры

Изменение узлов сети, мониторинг которых выполняется прокси

Обновите прокси, чтобы он отслеживал 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.

Свойство	Type	Описание
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 - с накоплением; 2 - круговой; 3 - разнесённый.
percent_left	float	Левый процентиль. По умолчанию: 0.
percent_right	float	Правый процентиль. По умолчанию: 0.
show_3d	integer	Показывать ли обнаруженные круговые и разнесённые 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)* показывать.
show_work_period	integer	Показывать ли рабочее время на обнаруженном графике. Возможные значения: 0 - скрыть; 1 - *(по умолчанию)* показывать.

Свойство	Type	Описание
templateid	ID	ID прототипа графика родительского 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. Генерируется автоматически, если не указан. Поведение свойства: - *поддерживается*, если прототип графика принадлежит шаблону

graphprototype.create

Описание

`object graphprototype.create( / graphPrototypes)`

Этот метод позволяет создавать новые прототипы графиков.

> **Note:**
> Этот метод доступен только пользователям с типом доступа *Администратор* и *Супер-администратор*. Разрешения на вызов метода можно отозвать в настройках ролей пользователя. Дополнительную информацию см. в разделе Роли пользователей.

Параметры

`(object/array)` Прототипы графиков для создания.

В дополнение к стандартным свойствам прототипа графика, метод принимает следующие параметры.

Параметр	Тип	Описание
gitems	array	Элементы графика, которые будут созданы 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. Поведение параметра: *- обязательный*

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": "Disk space usage {#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( graphPrototypeIds)`

Этот метод позволяет удалять прототипы графиков.

> **Note:**
> Этот метод доступен только пользователям с типом доступа *Администратор* и *Супер-администратор*. Разрешения на вызов метода можно отозвать в настройках ролей 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

Удалить два прототипа 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( )`

Этот метод позволяет получать прототипы графиков в соответствии с заданными параметрами.

> **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)` Параметры, определяющие желаемый вывод.

Метод поддерживает следующие параметры.

Параметр	Type	Описание
discoveryids	ID/array	Возвращать только прототипы графиков, принадлежащие указанным правилам обнаружения.
graphids	ID/array	Возвращать только прототипы графиков с указанными ID.
groupids	ID/array	Возвращать только прототипы графиков, принадлежащие узлам сети или шаблонам в указанных группах узлов сети или группах шаблонов.
hostids	ID/array	Возвращать только прототипы графиков, принадлежащие указанным узлам сети.
inherited	boolean	Если установлено в `true`, возвращать только прототипы графиков, унаследованные от шаблона.
itemids	ID/array	Возвращать только прототипы графиков, содержащие указанные прототипы элементов данных.
templated	boolean	Если установлено в `true`, возвращать только прототипы графиков, принадлежащие шаблонам.
templateids	ID/array	Возвращать только прототипы графиков, принадлежащие 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. Имеет следующие свойства: `parent_graphid` - `(ID)` ID прототипа графика, из которого был создан график; `status` - `(int)` состояние обнаружения графика: 0 - *(по умолчанию)* прототип графика обнаружен, 1 - прототип графика больше не обнаруживается; `ts_delete` - `(timestamp)` время, когда прототип графика, который больше не обнаруживается, будет удалён.
selectDiscoveryRule	query	Возвращать свойство `discoveryRule` с правилом LLD, которому принадлежит прототип графика.
selectDiscoveryRulePrototype	query	Возвращать свойство `discoveryRulePrototype` с родительским прототипом правила LLD, 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, которым принадлежит прототип графика.
selectHosts	query	Возвращать свойство `hosts` с узлами сети, 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, которым принадлежит прототип графика.
selectTemplates	query	Возвращать свойство `templates` с шаблонами, 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 одиночное 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 графика; `hostid` - ID узла сети, которому принадлежит прототип графика.
sortfield	string/array	Сортировать результат по указанным свойствам. Возможные значения: `graphid`, `name`, `graphtype`, `discovered`.

Параметр	Type	Описание
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

(/) 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": "                    {#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(  /  graphPrototypes)`

Этот метод позволяет обновлять существующие прототипы графиков.

> **Note:**
> Этот метод доступен только пользователям с типом доступа *Администратор* и *Супер-администратор*. Разрешения на вызов метода можно отозвать в настройках ролей 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 являются необязательными. Будут обновлены только переданные свойства, все остальные останутся без изменений.

В дополнение к стандартным свойствам прототипа графика, метод принимает следующие параметры.

Параметр	Тип	Описание
gitems	array	Элементы графика для замены существующих элементов графика. Если для элемента графика определено свойство `gitemid`, он будет обновлён, 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 до 1100 на 400 пикселей.

Запрос:

```
{
    "jsonrpc": "2.0",
    "method": "graphprototype.update",
    "params": {
        "graphid": "439",
        "width": 1100,
        "height": 400
    },
    "id": 1
}
```

Ответ:

```
{
    "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 LLD

Объект прототипа правила низкоуровневого обнаружения имеет следующие свойства.

Свойство	Тип	Описание
itemid	ID	ID прототипа правила LLD. Поведение свойства: - *только для чтения* - *обязательно* для операций обновления

Свойство	Тип	Описание
ruleid	ID	ID родительского правила LLD/прототипа правила LLD. Поведение свойства: - *только для чтения* - *обязательно* для операций создания
delay	string	Интервал обновления прототипа правила LLD. Принимает секунды или единицу времени с суффиксом (например, 30s, 1m, 2h, 1d) и, при необходимости, один или несколько пользовательских интервалов, разделенных точкой с запятой. Пользовательские интервалы могут представлять собой сочетание гибких интервалов и интервалов по расписанию. Поддерживает пользовательские макросы. Если они используются, значение должно быть одним макросом. Несколько макросов или макросы, смешанные с текстом, не поддерживаются. Гибкие интервалы могут быть записаны как два макроса, разделенные 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` установлено в ”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` установлено в ”Zabbix agent (active)” (7) и `key_` не содержит ”mqtt.get”
hostid	ID	ID узла сети, которому принадлежит прототип правила LLD. Поведение свойства: - *неизменяемое* - *обязательно* для операций создания
flags	integer	Происхождение прототипа правила LLD. Возможные значения: 3 - прототип правила низкоуровневого обнаружения; 7 - обнаруженный прототип правила низкоуровневого обнаружения. Поведение свойства: - *только для чтения*
interfaceid	ID	ID интерфейса узла сети для прототипа правила LLD. Поведение свойства: - *обязательно*, если прототип правила LLD принадлежит узлу сети и `type` установлено в ”Zabbix agent”, ”IPMI agent”, ”JMX agent” или ”SNMP agent” - *поддерживается*, если прототип правила LLD принадлежит узлу сети и `type` установлено в ”Simple check”, ”External check”, ”SSH agent”, ”TELNET agent” или ”HTTP agent”
key_	string	Ключ прототипа правила LLD. Требуется как минимум один макрос LLD. Поведение свойства: - *обязательно* для операций создания - *только для чтения* для унаследованных объектов

Свойство	Тип	Описание
name	string	Имя прототипа правила LLD. Поведение свойства: - *обязательно* для операций создания - *только для чтения* для унаследованных объектов
type	integer	Тип прототипа правила LLD. Возможные значения: 0 - Zabbix агент; 2 - Zabbix trapper; 3 - Простая проверка; 5 - Внутренний Zabbix; 7 - Zabbix агент (активный); 10 - Внешняя проверка; 11 - Монитор базы данных; 12 - IPMI агент; 13 - SSH агент; 14 - TELNET агент; 16 - JMX агент; 18 - Зависимый элемент данных; 19 - HTTP агент; 20 - SNMP агент; 21 - Скрипт; 22 - Браузер; 23 - Вложенный. 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` установлено в ”HTTP agent” - *только для чтения* для унаследованных объектов
allow_traps	integer	Разрешить заполнять значение аналогично элементу данных trapper. Возможные значения: 0 - *(по умолчанию)* Не разрешать принимать входящие данные; 1 - Разрешить принимать входящие данные. Поведение свойства: - *поддерживается*, если `type` установлено в ”HTTP agent”
authtype	integer	Метод аутентификации. Возможные значения, если `type` установлено в ”SSH agent”: 0 - *(по умолчанию)* пароль; 1 - открытый ключ. Возможные значения, если `type` установлено в ”HTTP agent”: 0 - *(по умолчанию)* отсутствует; 1 - basic; 2 - NTLM. Поведение свойства: - *поддерживается*, если `type` установлено в ”SSH agent” или ”HTTP agent” - *только для чтения* для унаследованных объектов (если `type` установлено в ”HTTP agent”)

Свойство	Тип	Описание
description	string	Описание прототипа правила LLD.
error	string	Текст ошибки, если возникают проблемы при обновлении значения прототипа правила LLD. 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` установлено в ”HTTP agent” - *только для чтения* для унаследованных объектов
headers	array	Массив заголовков, которые будут отправлены при выполнении HTTP-запроса. Поведение свойства: - *поддерживается*, если `type` установлено в ”HTTP agent” - *только для чтения* для унаследованных объектов
http_proxy	string	Строка подключения к HTTP(S) прокси. Поведение свойства: - *поддерживается*, если `type` установлено в ”HTTP agent” - *только для чтения* для унаследованных объектов
ipmi_sensor	string	Сенсор IPMI. Поведение свойства: - *обязательно*, если `type` установлено в ”IPMI agent” и `key_` не установлено в ”ipmi.get” - *поддерживается*, если `type` установлено в ”IPMI agent” - *только для чтения* для унаследованных объектов
jmx_endpoint	string	Пользовательская строка подключения JMX агент. По умолчанию: service:jmx:rmi:///jndi/rmi://{HOST.CONN}:{HOST.PORT}/jmxrmi Поведение свойства: - *поддерживается*, если `type` установлено в ”JMX agent”
lifetime	string	Период времени, по истечении которого элементы данных, которые больше не обнаруживаются, будут удалены. Принимает секунды, единицу времени с суффиксом или пользовательский макрос. По умолчанию: `7d`.
lifetime_type	integer	Сценарий удаления потерянных ресурсов LLD. Возможные значения: 0 - *(по умолчанию)* Удалить после достижения порога lifetime; 1 - Не удалять; 2 - Удалить немедленно.
enabled_lifetime	string	Период времени, по истечении которого элементы данных, которые больше не обнаруживаются, будут отключены. Принимает секунды, единицу времени с суффиксом или пользовательский макрос. По умолчанию: `0`.

Свойство	Тип	Описание
enabled_lifetime_type	integer	Сценарий отключения потерянных ресурсов LLD. Возможные значения: 0 - Отключить после достижения порога lifetime; 1 - Не отключать; 2 - *(по умолчанию)* Отключить немедленно.
master_itemid	ID	ID мастер-элемента данных. Правило обнаружения не может быть мастер-элементом данных для другого правила обнаружения. Поведение свойства: - *обязательно*, если `type` установлено в ”Dependent item” - *только для чтения* для унаследованных объектов
output_format	integer	Следует ли преобразовывать ответ в JSON. Возможные значения: 0 - *(по умолчанию)* Хранить как есть; 1 - Преобразовать в JSON. Поведение свойства: - *поддерживается*, если `type` установлено в ”HTTP agent” - *только для чтения* для унаследованных объектов
params	string	Дополнительные параметры в зависимости от типа прототипа правила LLD: - выполняемый скрипт для правил LLD SSH и Telnet; - SQL-запрос для правил LLD монитора базы данных; - формула для вычисляемых правил LLD; - скрипт для правил LLD типа script и browser. Поведение свойства: - *обязательно*, если `type` установлено в ”Database monitor”, ”SSH agent”, ”TELNET agent”, ”Script” или ”Browser” - *только для чтения* для унаследованных объектов (если `type` установлено в ”Script” или ”Browser”)
parameters	object/array	Дополнительные параметры, если `type` установлено в ”Script” или ”Browser”. Массив объектов со свойствами `name` и `value`, где `name` должно быть уникальным. Поведение свойства: - *поддерживается*, если `type` установлено в ”Script” или ”Browser” - *только для чтения* для унаследованных объектов
password	string	Пароль для аутентификации. Поведение свойства: - *обязательно*, если `type` установлено в ”JMX agent” и задано `username` - *поддерживается*, если `type` установлено в ”Simple check”, ”Database monitor”, ”SSH agent”, ”TELNET agent” или ”HTTP agent” - *только для чтения* для унаследованных объектов (если `type` установлено в ”HTTP agent”)
post_type	integer	Тип тела post-данных, хранящихся в свойстве `posts`. Возможные значения: 0 - *(по умолчанию)* Необработанные данные; 2 - JSON-данные; 3 - XML-данные. Поведение свойства: - *поддерживается*, если `type` установлено в ”HTTP agent” - *только для чтения* для унаследованных объектов

Свойство	Тип	Описание
posts	string	Данные тела HTTP(S)-запроса. Поведение свойства: - *обязательно*, если `type` установлено в ”HTTP agent” и `post_type` установлено в ”JSON data” или ”XML data” - *поддерживается*, если `type` установлено в ”HTTP agent” и `post_type` установлено в ”Raw data” - *только для чтения* для унаследованных объектов
privatekey	string	Имя файла закрытого ключа. Поведение свойства: - *обязательно*, если `type` установлено в ”SSH agent” и `authtype` установлено в ”public key”
publickey	string	Имя файла открытого ключа. Поведение свойства: - *обязательно*, если `type` установлено в ”SSH agent” и `authtype` установлено в ”public key”
query_fields	array	Массив полей запроса, которые будут отправлены при выполнении HTTP-запроса. Поведение свойства: - *поддерживается*, если `type` установлено в ”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` установлено в ”HTTP agent” - *только для чтения* для унаследованных объектов
retrieve_mode	integer	Какая часть ответа должна быть сохранена. Возможные значения, если `request_method` установлено в ”GET”, ”POST” или ”PUT”: 0 - *(по умолчанию)* Тело; 1 - Заголовки; 2 - Будут сохранены и тело, и заголовки. Возможные значения, если `request_method` установлено в ”HEAD”: 1 - Заголовки. Поведение свойства: - *поддерживается*, если `type` установлено в ”HTTP agent” - *только для чтения* для унаследованных объектов
snmp_oid	string	SNMP OID. Поведение свойства: - *обязательно*, если `type` установлено в ”SNMP agent” - *только для чтения* для унаследованных объектов
ssl_cert_file	string	Путь к файлу открытого SSL-ключа. Поведение свойства: - *поддерживается*, если `type` установлено в ”HTTP agent” - *только для чтения* для унаследованных объектов

Свойство	Тип	Описание
ssl_key_file	string	Путь к файлу закрытого SSL-ключа. Поведение свойства: - *поддерживается*, если `type` установлено в ”HTTP agent” - *только для чтения* для унаследованных объектов
ssl_key_password	string	Пароль для файла SSL-ключа. Поведение свойства: - *поддерживается*, если `type` установлено в ”HTTP agent” - *только для чтения* для унаследованных объектов
state	integer	Состояние прототипа правила LLD. Возможные значения: 0 - *(по умолчанию)* нормальное; 1 - не поддерживается. Поведение свойства: - *только для чтения*
status	integer	Статус прототипа правила LLD. Возможные значения: 0 - *(по умолчанию)* включенный прототип правила LLD; 1 - отключенный прототип правила LLD.
status_codes	string	Диапазоны требуемых HTTP-кодов состояния, разделенные запятыми. Также поддерживает пользовательские макросы как часть списка, разделенного запятыми. Пример: 200,200-{$M},{$M},200-400 Поведение свойства: - *поддерживается*, если `type` установлено в ”HTTP agent” - *только для чтения* для унаследованных объектов
templateid	ID	ID родительского шаблона прототипа правила LLD. Поведение свойства: - *только для чтения*
timeout	string	Тайм-аут запроса опроса данных элемента данных. Принимает секунды или единицу времени с суффиксом (например, 30s, 1m). Также принимает пользовательские макросы. Диапазон возможных значений: 1-600s. По умолчанию: ”” - использовать настройки прокси/глобальные настройки. Поведение свойства: - *поддерживается*, если `type` установлено в ”Zabbix agent” (0), ”Simple check” (3) и `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) и `snmp_oid` начинается с ”`walk[`” или ”`get[`”, ”Script” (21), ”Browser” (22) - *только для чтения* для унаследованных объектов
trapper_hosts	string	Разрешенные узлы сети. Поведение свойства: - *поддерживается*, если `type` установлено в ”Zabbix trapper”, или если `type` установлено в ”HTTP agent” и `allow_traps` установлено в ”Allow to accept incoming data”

Свойство	Тип	Описание
username	string	Имя пользователя для аутентификации. Поведение свойства: - *обязательно*, если `type` установлено в ”SSH agent”, ”TELNET agent”, или если `type` установлено в ”JMX agent” и задано `password` - *поддерживается*, если `type` установлено в ”Simple check”, ”Database monitor” или ”HTTP agent” - *только для чтения* для унаследованных объектов (если `type` установлено в ”HTTP agent”)
uuid	string	Универсальный уникальный идентификатор, используемый для связывания импортированных прототипов правил LLD с уже существующими. Генерируется автоматически, если не указан. Поведение свойства: - *поддерживается*, если прототип правила LLD принадлежит шаблону
verify_host	integer	Следует ли проверять, что имя узла сети для подключения совпадает с именем в сертификате узла сети. Возможные значения: 0 - *(по умолчанию)* Не проверять; 1 - Проверять. Поведение свойства: - *поддерживается*, если `type` установлено в ”HTTP agent” - *только для чтения* для унаследованных объектов
verify_peer	integer	Следует ли проверять подлинность сертификата узла сети. Возможные значения: 0 - *(по умолчанию)* Не проверять; 1 - Проверять. Поведение свойства: - *поддерживается*, если `type` установлено в ”HTTP agent” - *только для чтения* для унаследованных объектов

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*
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	Type	Описание
type	integer	Тип варианта предобработки. Возможные значения: 5 - Регулярное выражение; 11 - XML XPath; 12 - JSONPath; 14 - Соответствует регулярному выражению; 15 - Не соответствует регулярному выражению; 16 - Проверка ошибки в JSON; 17 - Проверка ошибки в XML; 20 - Отбрасывать неизменившееся с интервалом; 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: - *обязательно*
params	string	Дополнительные параметры, используемые вариантом предобработки. Несколько параметров разделяются символом новой строки (\n). Поведение свойства: - *обязательно*, если `type` установлен в ”Регулярное выражение” (5), ”XML XPath” (11), ”JSONPath” (12), ”Соответствует регулярному выражению” (14), ”Не соответствует регулярному выражению” (15), ”Проверка ошибки в JSON” (16), ”Проверка ошибки в XML” (17), ”Отбрасывать неизменившееся с интервалом” (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)
error_handler	integer	Тип действия, используемого 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: - *обязательно*, если `type` установлен в ”Регулярное выражение” (5), ”XML XPath” (11), ”JSONPath” (12), ”Соответствует регулярному выражению” (14), ”Не соответствует регулярному выражению” (15), ”Проверка ошибки в JSON” (16), ”Проверка 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: - *обязательно*, если `error_handler` установлен в ”Установить пользовательское значение” или ”Установить пользовательское сообщение об ошибке”

Для каждого типа предобработки поддерживаются следующие параметры и обработчики ошибок.

Тип предобработки	Имя	Параметр 1	Параметр 2	Параметр 3	Поддерживаемые обработчики ошибок
5	Регулярное выражение	шаблон[1]	вывод[2]		0, 1, 2, 3
11	XML XPath	путь[3]			0, 1, 2, 3
12	JSONPath	путь[3]			0, 1, 2, 3
14	Соответствует регулярному выражению	шаблон[1]			0, 1, 2, 3
15	Не соответствует регулярному выражению	шаблон[1]			0, 1, 2, 3
16	Проверка на ошибки в JSON	путь[3]			0, 1, 2, 3
17	Проверка на ошибки в XML	путь[3]			0, 1, 2, 3
20	Отбрасывать неизменившееся с интервалом	секунды[4, 5]			
21	JavaScript	скрипт[2]			
23	Prometheus в JSON	шаблон[5, 6]			0, 1, 2, 3
24	CSV в JSON	символ[2]	символ[2]	0,1	0, 1, 2, 3
25	Замена	строка поиска[2]	замена[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

Тип предобработки	Имя	Параметр 1	Параметр 2	Параметр 3	Поддерживаемые обработчики ошибок
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, следующий 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.
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*

Свойство	Тип	Описание
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*
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”

Свойство	Type	Описание
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.
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, которое задается для обнаруженного элемента данных. Он имеет следующие свойства:

Свойство	Type	Описание
delay	string	Переопределяет интервал обновления прототипа элемента данных. Допускает секунды или единицу времени с суффиксом (например, 30s, 1m, 2h, 1d), а также, при необходимости, один или несколько пользовательских интервалов, разделенных точкой с запятой. Пользовательские интервалы могут представлять собой сочетание гибких интервалов и интервалов по расписанию. Допускает пользовательские макросы или макросы LLD. Если они используются, значение должно быть одним макросом. Несколько макросов или макросы в сочетании с текстом не поддерживаются. Гибкие интервалы могут быть записаны как два макроса, разделенные 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: - *required*

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 типам пользователей *Admin* и *Super admin*. Разрешения на вызов метода могут быть отозваны в настройках роли пользователя. Для получения дополнительной информации смотрите 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 пользователям типов *Admin* и *Super admin*. Разрешения на вызов метода могут быть отозваны в настройках роли пользователя. Дополнительную информацию смотрите 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)`

Этот метод позволяет получать прототипы правил LLD в соответствии с заданными параметрами.

Note:
Этот метод доступен пользователям любого типа. Права на вызов метода могут быть отозваны 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.

Параметр	Тип	Описание
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.
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	

Параметр	Тип	Описание
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 пользователям типов *Admin* и *Super admin*. Разрешения на вызов метода могут быть отозваны в настройках ролей пользователей. Для получения дополнительной информации смотрите 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.

Свойство	Тип	Описание
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
expression	string	Выражение триггера. Должно содержать как минимум один прототип элемента данных. Прототипы элементов данных, у которых `value_type` установлен в 5 (binary) или 6 (JSON), не могут использоваться в выражениях триггера. Поведение свойства: - *обязательно* для операций создания
event_name	string	Имя события, генерируемого триггером.
opdata	string	Оперативные данные.
comments	string	Дополнительные комментарии к прототипу триггера.
priority	integer	Важность прототипа триггера. Возможные значения: 0 - *(по умолчанию)* не классифицирован; 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*

Свойство	Тип	Описание
templateid	ID	ID родительского прототипа триггера шаблона. Поведение свойства: - *только для чтения*
type	integer	Может ли прототип триггера генерировать несколько событий проблем. 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 OK-события. 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, у которых `value_type` установлен в 5 (binary) или 6 (JSON), не могут использоваться в выражениях восстановления триггера.
correlation_mode	integer	Закрытие OK-событием. 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)* новые триггеры будут обнаруживаться; 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. Генерируется автоматически, если не указан. Поведение свойства: - *поддерживается*, если прототип триггера принадлежит шаблону

Тег прототипа триггера

Объект тега прототипа триггера имеет следующие свойства.

Свойство	Type	Описание
tag	string	Имя тега прототипа триггера. Поведение свойства: - *обязательно*

Свойство	Type	Описание
value	string	Значение тега прототипа 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; 4 - прототип элемента данных. Поведение свойства: - *только для чтения*.
objectid	ID	ID объекта, от которого был унаследован тег. Поведение свойства: - *только для чтения*.

triggerprototype.create

Описание

`object triggerprototype.create( / triggerPrototypes)`

Этот метод позволяет создавать новые прототипы триггеров.

Note:
Этот метод доступен только пользователям с типом доступа *Администратор* и *Супер-администратор*. Разрешения на вызов метода можно отозвать в настройках ролей пользователя. Дополнительную информацию см. в разделе Роли пользователей.

Параметры

`(object/array)` Прототипы триггеров для создания.

В дополнение к стандартным свойствам прототипа триггера этот метод принимает следующие параметры.

Параметр	Тип	Описание
dependencies	array	Триггеры и прототипы триггеров, от которых зависит прототип триггера. Для 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 места на диске.

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(    triggerPrototypeIds)`

Этот метод позволяет удалять прототипы триггеров.

> **Note:**
> Этот метод доступен только пользователям с типом доступа *Администратор* и *Супер-администратор*. Разрешения на вызов метода можно отозвать в настройках ролей пользователя. Дополнительную информацию см. в разделе Роли пользователей.

Параметры

`(array)` Идентификаторы прототипов триггеров для удаления.

Возвращаемые значения

`(object)` Возвращает объект, содержащий идентификаторы удалённых прототипов триггеров в свойстве `triggerids`.

Примеры

Удаление нескольких прототипов триггеров

Удалите два прототипа 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)`

Этот 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 смотрите в разделе Роли пользователей.

Параметры

`(object)` Параметры, определяющие желаемый вывод.

Метод поддерживает следующие параметры.

Параметр	Тип	Описание
active	flag	Возвращать только включённые прототипы триггеров, принадлежащие отслеживаемым узлам сети.
discoveryids	ID/array	Возвращать только прототипы триггеров, принадлежащие указанным правилам LLD.
functions	string/array	Возвращать только триггеры, использующие указанные функции. Список поддерживаемых функций см. на странице Поддерживаемые функции.
group	string	Возвращать только прототипы триггеров, принадлежащие узлам сети или шаблонам из групп узлов сети или групп шаблонов с указанным именем.
groupids	ID/array	Возвращать только прототипы триггеров, принадлежащие узлам сети или шаблонам из указанных групп узлов сети или групп шаблонов.
host	string	Возвращать только прототипы триггеров, принадлежащие узлам сети с указанным именем.
hostids	ID/array	Возвращать только прототипы триггеров, принадлежащие указанным узлам сети.
inherited	boolean	Если установлено в `true`, возвращать только прототипы триггеров, унаследованные от шаблона.
maintenance	boolean	Если установлено в `true`, возвращать только включённые прототипы триггеров, принадлежащие узлам сети, находящимся на обслуживании.
min_severity	integer	Возвращать только прототипы триггеров с важностью больше или равной указанной важности.

Параметр	Тип	Описание
monitored	flag	Возвращать только включённые прототипы триггеров, принадлежащие отслеживаемым узлам сети и содержащие только включённые элементы данных.
templated	boolean	Если установлено в `true`, возвращать только прототипы триггеров, принадлежащие шаблонам.
templateids	ID/array	Возвращать только прототипы триггеров, принадлежащие 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 обнаруживается; `ts_delete` - `(timestamp)` время, когда прототип триггера, который больше не обнаруживается, будет удалён; `ts_disable` - `(timestamp)` время, когда прототип триггера, который больше не обнаруживается, будет отключён; `disable_source` - `(int)` индикатор того, был ли прототип триггера отключён правилом LLD или вручную: 0 - *(по умолчанию)* отключён автоматически, 1 - отключён правилом LLD.
selectDiscoveryRule	query	Возвращать правило LLD, которому принадлежит прототип триггера, в свойстве `discoveryRule`.
selectDiscoveryRulePrototype	query	Возвращать свойство `discoveryRulePrototype` с родительским прототипом правила LLD, которому принадлежит прототип триггера.
selectFunctions	query	Возвращать функции, используемые в прототипе триггера, в свойстве `functions`. Объекты функций 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, передаваемый функции. Параметр запроса заменяется символом $ в возвращаемой строке.
selectHostGroups	query	Возвращать группы узлов сети, которым принадлежит прототип триггера, в свойстве `hostgroups`.
selectHosts	query	Возвращать узлы сети, 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 есть ссылки в выражениях 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, которым принадлежит прототип триггера, в свойстве `templategroups`.

Параметр	Тип	Описание
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 триггера; `hostid` - ID узла сети, которому принадлежит прототип триггера.
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: `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 одно из следующего:

- массив объектов;
- количество полученных объектов, если был использован параметр `countOutput`.

Примеры

Получение прототипов триггеров из правила LLD

Получить 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": "      inode     20%      {#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": "                    20%      {#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 определённого 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( / triggerPrototypes)`

Этот метод позволяет обновлять существующие прототипы триггеров.

> **Note:**
> Этот метод доступен только пользователям с типом доступа *Администратор* и *Супер-администратор*. Разрешения на вызов метода можно отозвать в настройках ролей пользователя. Дополнительную информацию см. в разделе Роли пользователей.

Параметры

`(object/array)` Свойства прототипа триггера, которые необходимо обновить.

Свойство `triggerid` должно быть определено для каждого прототипа триггера, все остальные свойства являются необязательными. Будут обновлены только переданные свойства, все остальные останутся без изменений.

В дополнение к стандартным свойствам прототипа триггера метод принимает следующие параметры.

Параметр	Type	Описание
dependencies	array	Триггеры и прототипы триггеров, от которых зависит прототип триггера. Для триггеров должно быть определено только свойство `triggerid`.
tags	array	Теги прототипа триггера.

Возвращаемые значения

`(object)` Возвращает объект, содержащий идентификаторы обновлённых прототипов триггеров в свойстве `triggerids`.

Примеры

Включение прототипа триггера

Включите прототип триггера, то есть установите его статус в ”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

Заменить 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 с группой
- 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.

Свойство	Тип	Описание
hostid	ID	ID прототипа узла сети. Поведение свойства: - *только для чтения* - *обязательно* для операций обновления
host	string	Техническое имя прототипа узла сети. Поведение свойства: - *обязательно* для операций создания - *только для чтения* для унаследованных объектов
name	string	Видимое имя прототипа узла сети. По умолчанию: значение свойства `host`. Поведение свойства: - *только для чтения* для унаследованных объектов
status	integer	Состояние прототипа узла сети. Возможные значения: 0 - *(по умолчанию)* отслеживаемый узел сети; 1 - неотслеживаемый узел сети.
flags	integer	Происхождение прототипа узла сети. Возможные значения: 2 - прототип узла сети; 6 - обнаруженный прототип узла сети Поведение свойства: - *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 прототипа узла сети шаблона. Поведение свойства: - *только для чтения*
discover	integer	Состояние 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 прототипом узла сети. Возможные значения: 0 - *(по умолчанию)* наследовать интерфейсы от родительского узла сети; 1 - использовать пользовательские интерфейсы прототипов узлов сети. Поведение свойства: - *только для чтения* для унаследованных объектов

Свойство	Тип	Описание
uuid	string	Универсальный уникальный идентификатор, используемый для связывания импортированных прототипов узлов сети с уже существующими. Генерируется автоматически, если не указан. Поведение свойства: - *поддерживается*, если прототип узла сети принадлежит шаблону

Связь с группой

Объект связи с группой связывает прототип узла сети с группой узлов сети. Он имеет следующие свойства.

Свойство	Тип	Описание
groupid	ID	ID группы узлов сети. Поведение свойства: - *обязательно*

Прототип группы

Объект прототипа группы определяет группу, которая будет создана для обнаруженного узла сети. Он имеет следующие свойства.

Свойство	Type	Описание
group_prototypeid	ID	ID прототипа группы. Property behavior: - *только для чтения*
name	string	Имя прототипа группы. Property behavior: - *обязательно* для операций создания

Тег прототипа узла сети

Объект тега прототипа узла сети имеет следующие свойства.

Свойство	Type	Описание
tag	string	Имя тега прототипа узла сети. Поведение свойства: - *обязательно*
value	string	Значение тега прототипа узла сети. Поведение свойства: - *только для чтения*.
object	integer	Тип объекта, от которого был унаследован тег. Возможные значения: 0 - шаблон. Поведение свойства: - *только для чтения*.
objectid	ID	ID объекта, от которого был унаследован тег. Поведение свойства: - *только для чтения*.

Пользовательский интерфейс

Пользовательские интерфейсы поддерживаются, если `custom_interfaces` объекта Host prototype object установлено в значение ”use host prototypes custom interfaces”. Объект пользовательского интерфейса имеет следующие свойства.

Свойство	Type	Описание
type	integer	Тип интерфейса. Возможные значения: 1 - Агент; 2 - SNMP; 3 - IPMI; 4 - JMX. Property behavior: - *required*
useip	integer	Следует ли выполнять подключение по IP. Возможные значения: 0 - подключаться, используя DNS-имя узла сети; 1 - подключаться, используя IP-адрес узла сети. Property behavior: - *required*
ip	string	IP-адрес, используемый интерфейсом. Может содержать макросы. Property behavior: - *required* если `useip` установлено в значение ”connect using host IP address”
dns	string	DNS-имя, используемое интерфейсом. Может содержать макросы. Property behavior: - *required* если `useip` установлено в значение ”connect using host DNS name”
port	string	Номер порта, используемый интерфейсом. Может содержать пользовательские макросы и макросы LLD. Property behavior: - *required*
main	integer	Используется ли интерфейс как интерфейс по умолчанию на узле сети. Только один интерфейс определенного типа может быть установлен как интерфейс по умолчанию на узле сети. Возможные значения: 0 - не по умолчанию; 1 - по умолчанию. Property behavior: - *required*
details	object	Дополнительный объект для сведений о пользовательском интерфейсе. Property behavior: - *required* если `type` установлено в значение ”SNMP”

Сведения о пользовательском интерфейсе

Объект details имеет следующие свойства.

Свойство	Тип	Описание
version	integer	Версия интерфейса SNMP. Возможные значения: 1 - SNMPv1; 2 - SNMPv2c; 3 - SNMPv3. Поведение свойства: - *обязательно*
bulk	integer	Следует ли использовать bulk-запросы SNMP. Возможные значения: 0 - не использовать bulk-запросы; 1 - *(по умолчанию)* - использовать bulk-запросы.
community	string	Сообщество SNMP. Поведение свойства: - *обязательно*, если `version` установлено в ”SNMPv1” или ”SNMPv2c”
max_repetitions	integer	Максимальное значение повторений для собственных SNMP bulk-запросов (GetBulkRequest-PDUs). Используется только для элементов данных `discovery[]` и `walk[]` в SNMPv2 и v3. По умолчанию: 10.
securityname	string	Имя безопасности SNMPv3. Поведение свойства: - *поддерживается*, если `version` установлено в ”SNMPv3”
securitylevel	integer	Уровень безопасности SNMPv3. Возможные значения: 0 - *(по умолчанию)* - noAuthNoPriv; 1 - authNoPriv; 2 - authPriv. Поведение свойства: - *поддерживается*, если `version` установлено в ”SNMPv3”
authpassphrase	string	Парольная фраза аутентификации SNMPv3. Поведение свойства: - *поддерживается*, если `version` установлено в ”SNMPv3” и `securitylevel` установлено в ”authNoPriv” или ”authPriv”
privpassphrase	string	Парольная фраза приватности SNMPv3. Поведение свойства: - *поддерживается*, 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: - *поддерживается*, если `version` установлено в ”SNMPv3” и `securitylevel` установлено в ”authNoPriv” или ”authPriv”

Свойство	Тип	Описание
privprotocol	integer	Протокол приватности SNMPv3. Используется только интерфейсами SNMPv3. Возможные значения: 0 - *(по умолчанию)* - DES; 1 - AES128; 2 - AES192; 3 - AES256; 4 - AES192C; 5 - AES256C. Поведение свойства: - *поддерживается*, если `version` установлено в ”SNMPv3” и `securitylevel` установлено в ”authPriv”
contextname	string	Имя контекста SNMPv3. Поведение свойства: - *поддерживается*, если `version` установлено в ”SNMPv3”

hostprototype.create

Описание

`object hostprototype.create( / hostPrototypes)`

Этот метод позволяет создавать новые прототипы узлов сети.

Note:
Этот метод доступен только пользователям с типом доступа *Администратор* и *Супер-администратор*. Разрешения на вызов метода можно отозвать в настройках ролей пользователя. Дополнительную информацию см. в разделе Роли пользователей.

Параметры

`(object/array)` Прототипы узлов сети для создания.

В дополнение к стандартным свойствам прототипа узла сети, метод принимает следующие параметры.

Параметр	Тип	Описание
groupLinks	array	Связи с группами, которые будут созданы для прототипа узла сети. Поведение параметра: - *обязательный*
ruleid	ID	ID правила LLD, к которому относится прототип узла сети. Поведение параметра: - *обязательный*
groupPrototypes	array	Прототипы групп, которые будут созданы для прототипа узла сети.
macros	object/array	Пользовательские макросы, которые будут созданы для прототипа узла сети.
tags	object/array	Теги прототипа узла сети.
interfaces	object/array	Пользовательские интерфейсы прототипа узла сети.
templates	object/array	Шаблоны, которые будут привязаны к прототипу узла сети. Для шаблонов должно быть определено 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}" в правиле 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

- Связь с группой
- 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(    hostPrototypeIds)`

Этот метод позволяет удалять прототипы узлов сети.

Note:
Этот метод доступен только пользователям с типом доступа *Администратор* и *Супер-администратор*. Разрешения на вызов метода можно отозвать в настройках ролей 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

Удалите два прототипа 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)`

Этот метод позволяет получать прототипы узлов сети в соответствии с заданными параметрами.

> **Note:**
> Этот метод доступен пользователям любого типа. Права на вызов метода могут быть отозваны в настройках роли пользователя. См. Роли пользователей для получения дополнительной информации.

Параметры

`(object)` Параметры, определяющие желаемый вывод.

Метод поддерживает следующие параметры.

Параметр	Тип	Описание
hostids	ID/array	Возвращать только прототипы узлов сети с указанными ID.
discoveryids	ID/array	Возвращать только прототипы узлов сети, принадлежащие указанным правилам LLD.
inherited	boolean	Если установлено значение `true`, возвращать только элементы данных, унаследованные от шаблона.
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. 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 - *(по умолчанию)* прототип узла сети обнаружен, 1 - прототип узла сети больше не обнаруживается; `ts_delete` - `(timestamp)` время, когда прототип узла сети, который больше не обнаруживается, будет удалён; `ts_disable` - `(timestamp)` время, когда прототип узла сети, который больше не обнаруживается, будет отключён; `disable_source` - `(int)` индикатор того, был ли прототип узла сети отключён правилом LLD или вручную: 0 - *(по умолчанию)* отключён автоматически, 1 - отключён правилом LLD.
selectDiscoveryRule	query	Возвращать свойство `discoveryRule` с правилом LLD, которому принадлежит прототип узла сети.
selectDiscoveryRulePrototype	query	Возвращать свойство `discoveryRulePrototype` с родительским прототипом правила LLD, 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, находящимися в связанных шаблонах.
selectMacros	query	Возвращать свойство `macros` с макросами прототипа узла сети.
selectParentHost	query	Возвращать свойство `parentHost` с узлом сети, которому принадлежит прототип узла сети.
selectTags	query	Возвращать свойство `tags` с тегами прототипа узла сети.

Параметр	Тип	Описание
selectTemplates	query	Возвращать свойство `templates` со связанными с прототипом узла сети шаблонами. 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 одно из следующего:

- массив объектов;
- количество полученных объектов, если был использован параметр `countOutput`.

Примеры

Получение прототипов узлов сети из правила LLD

Получить 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

- Связь с группой
- Прототип группы
- Пользовательский макрос

Исходный код

CHostPrototype::get() в *ui/include/classes/api/services/CHostPrototype.php*.

hostprototype.update

Описание

`object hostprototype.update( / hostPrototypes)`

Этот метод позволяет обновлять существующие прототипы узлов сети.

> **Note:**
> Этот метод доступен только пользователям с типом доступа *Администратор* и *Супер-администратор*. Разрешения на вызов метода можно отозвать в настройках ролей пользователя. Дополнительную информацию см. в разделе Роли пользователей.

Параметры

`(object/array)` Свойства прототипов узлов сети, которые необходимо обновить.

Свойство `hostid` должно быть определено для каждого прототипа узла сети, все остальные свойства являются необязательными. Будут обновлены только переданные свойства, все остальные останутся без изменений.

В дополнение к стандартным свойствам прототипа узла сети, метод принимает следующие параметры.

Параметр	Тип	Описание
groupLinks	array	Связи с группами для замены текущих связей с группами у прототипа узла сети. Поведение параметра: - *только для чтения* для унаследованных объектов
groupPrototypes	array	Прототипы групп для замены существующих прототипов групп у прототипа узла сети. Все прототипы групп, не указанные в запросе, будут удалены. Поведение параметра: - *только для чтения* для унаследованных объектов
macros	object/array	Пользовательские макросы для замены текущих пользовательских макросов. Все макросы, не указанные в запросе, будут удалены.
tags	object/array	Теги прототипа узла сети для замены текущих тегов. Все теги, не указанные в запросе, будут удалены. Поведение параметра: - *только для чтения* для унаследованных объектов
interfaces	object/array	Пользовательские интерфейсы прототипа узла сети для замены текущих интерфейсов. Объект пользовательского интерфейса должен содержать все свои параметры. Все интерфейсы, не указанные в запросе, будут удалены. Поведение параметра: - *поддерживается*, если `custom_interfaces` объекта Host prototype object установлено в значение "use host prototypes custom interfaces" - *только для чтения* для унаследованных объектов
templates	object/array	Шаблоны для замены связанных в данный момент шаблонов. 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 его статус в "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 прототипа узла сети

Замените унаследованные интерфейсы пользовательскими интерфейсами прототипа узла сети.

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

- Связь с группой
- 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-заголовок
  - Поле HTTP-запроса
- 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	Type	Описание
itemid	ID	ID прототипа элемента данных. Поведение свойства: - *только для чтения* - *обязательно* для операций обновления
delay	string	Интервал обновления прототипа элемента данных. Принимает секунды или единицу времени с суффиксом (например, 30s, 1m, 2h, 1d) и, при необходимости, один или несколько пользовательских интервалов, разделенных точкой с запятой. Пользовательские интервалы могут представлять собой сочетание гибких интервалов и интервалов по расписанию. Принимает пользовательские макросы и макросы LLD. Если они используются, значение должно быть одним макросом. Несколько макросов или макросы, смешанные с текстом, не поддерживаются. Гибкие интервалы могут быть записаны как два макроса, разделенные 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` задан как ”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), либо если `type` задан как ”Zabbix agent (active)” (7) и `key_` не содержит ”mqtt.get”
hostid	ID	ID узла сети, которому принадлежит прототип элемента данных. Поведение свойства: - *неизменяемое* - *обязательно* для операций создания
interfaceid	ID	ID интерфейса узла сети прототипа элемента данных. Поведение свойства: - *обязательно*, если прототип элемента данных принадлежит узлу сети и `type` задан как ”Zabbix agent”, ”IPMI agent”, ”JMX agent”, ”SNMP trap” или ”SNMP agent” - *поддерживается*, если прототип элемента данных принадлежит узлу сети и `type` задан как ”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

Свойство	Type	Описание
name	string	Имя прототипа элемента данных. Поддерживает пользовательские макросы. Поведение свойства: - *обязательно* для операций создания - *только для чтения* для унаследованных объектов
type	integer	Тип прототипа элемента данных. Возможные значения: 0 - Zabbix агент; 2 - Zabbix trapper; 3 - Простая проверка; 5 - Внутренний элемент Zabbix; 7 - Zabbix агент (активный); 10 - Внешняя проверка; 11 - Монитор базы данных; 12 - IPMI агент; 13 - SSH агент; 14 - TELNET агент; 15 - Вычисляемый; 16 - JMX агент; 17 - SNMP trap; 18 - Зависимый элемент данных; 19 - HTTP агент; 20 - SNMP агент; 21 - Скрипт; 22 - Браузер. 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` задан как ”HTTP agent” - *только для чтения* для унаследованных объектов
value_type	integer	Тип информации прототипа элемента данных. Возможные значения: 0 - числовой с плавающей точкой; 1 - символьный; 2 - журнал; 3 - числовой без знака; 4 - текст; 5 - двоичный; 6 - JSON. Поведение свойства: - *обязательно* для операций создания - *только для чтения* для унаследованных объектов
allow_traps	integer	Разрешить заполнять значение аналогично элементу данных trapper. 0 - *(по умолчанию)* Не разрешать прием входящих данных; 1 - Разрешить прием входящих данных. Поведение свойства: - *поддерживается*, если `type` задан как ”HTTP agent”

Свойство	Type	Описание
authtype	integer	Метод аутентификации. Возможные значения, если type задан как ”SSH agent”: 0 - *(по умолчанию)* пароль; 1 - открытый ключ. Возможные значения, если type задан как ”HTTP agent”: 0 - *(по умолчанию)* нет; 1 - basic; 2 - NTLM; 3 - Kerberos; 4 - Digest. Поведение свойства: - *поддерживается*, если type задан как ”SSH agent” или ”HTTP agent” - *только для чтения* для унаследованных объектов (если type задан как ”HTTP agent”)
description	string	Описание прототипа элемента данных.
flags	integer	Происхождение 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 задан как ”HTTP agent” - *только для чтения* для унаследованных объектов
headers	array	Массив заголовков, которые будут отправлены при выполнении HTTP-запроса. Поведение свойства: - *поддерживается*, если type задан как ”HTTP agent” - *только для чтения* для унаследованных объектов
history	string	Единица времени, определяющая, как долго должны храниться исторические данные. Также принимает пользовательский макрос и макрос LLD. По умолчанию: 31d.
http_proxy	string	Строка подключения к HTTP(S) прокси. Поведение свойства: - *поддерживается*, если type задан как ”HTTP agent” - *только для чтения* для унаследованных объектов
ipmi_sensor	string	Сенсор IPMI. Поведение свойства: - *обязательно*, если type задан как ”IPMI agent” и key_ не задан как ”ipmi.get” - *поддерживается*, если type задан как ”IPMI agent” - *только для чтения* для унаследованных объектов

Свойство	Type	Описание
jmx_endpoint	string	Пользовательская строка подключения JMX агента. По умолчанию: service:jmx:rmi:///jndi/rmi://{HOST.CONN}:{HOST.PORT}/jmxrmi Поведение свойства: - *поддерживается*, если `type` задан как "JMX agent"
logtimefmt	string	Формат времени в записях журнала. Поведение свойства: - *поддерживается*, если `value_type` задан как "log" - *только для чтения* для унаследованных объектов
master_itemid	ID	ID мастер-элемента данных. Поведение свойства: - *обязательно*, если `type` задан как "Dependent item" - *только для чтения* для унаследованных объектов
output_format	integer	Следует ли преобразовывать ответ в JSON. Возможные значения: 0 - *(по умолчанию)* Хранить как есть; 1 - Преобразовать в JSON. Поведение свойства: - *поддерживается*, если `type` задан как "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 агент и TELNET агент; - SQL-запрос для прототипов элементов данных мониторинга базы данных; - формула для вычисляемых прототипов элементов данных; - скрипт для прототипов элементов данных скрипта и браузера. Поведение свойства: - *обязательно*, если `type` задан как "Database monitor", "SSH agent", "TELNET agent", "Calculated", "Script" или "Browser" - *только для чтения* для унаследованных объектов (если `type` задан как "Script" или "Browser")
parameters	object/array	Дополнительные параметры, если `type` задан как "Script" или "Browser". Массив объектов со свойствами `name` и `value`, где `name` должно быть уникальным. Поведение свойства: - *поддерживается*, если `type` задан как "Script" или "Browser" - *только для чтения* для унаследованных объектов
password	string	Пароль для аутентификации. Поведение свойства: - *обязательно*, если `type` задан как "JMX agent" и задан `username` - *поддерживается*, если `type` задан как "Simple check", "SSH agent", "TELNET agent", "Database monitor" или "HTTP agent" - *только для чтения* для унаследованных объектов (если `type` задан как "HTTP agent")

Свойство	Type	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` задан как "HTTP agent" - *только для чтения* для унаследованных объектов
posts	string	Данные тела HTTP(S)-запроса. Поведение свойства: - *обязательно*, если `type` задан как "HTTP agent" и `post_type` задан как "JSON data" или "XML data" - *поддерживается*, если `type` задан как "HTTP agent" и `post_type` задан как "Raw data" - *только для чтения* для унаследованных объектов
privatekey	string	Имя файла закрытого ключа. Поведение свойства: - *обязательно*, если `type` задан как "SSH agent" и `authtype` задан как "public key"
publickey	string	Имя файла открытого ключа. Поведение свойства: - *обязательно*, если `type` задан как "SSH agent" и `authtype` задан как "public key"
query_fields	array	Массив полей запроса, которые будут отправлены при выполнении HTTP-запроса. Поведение свойства: - *поддерживается*, если `type` задан как "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` задан как "HTTP agent" - *только для чтения* для унаследованных объектов
retrieve_mode	integer	Какая часть ответа должна быть сохранена. Возможные значения, если `request_method` задан как "GET", "POST" или "PUT": 0 - *(по умолчанию)* Тело; 1 - Заголовки; 2 - Будут сохранены и тело, и заголовки. Возможные значения, если `request_method` задан как "HEAD": 1 - Заголовки. Поведение свойства: - *поддерживается*, если `type` задан как "HTTP agent" - *только для чтения* для унаследованных объектов

Свойство	Type	Описание
snmp_oid	string	SNMP OID. Поведение свойства: - *обязательно*, если `type` задан как ”SNMP agent” - *только для чтения* для унаследованных объектов
ssl_cert_file	string	Путь к файлу открытого SSL-ключа. Поведение свойства: - *поддерживается*, если `type` задан как ”HTTP agent” - *только для чтения* для унаследованных объектов
ssl_key_file	string	Путь к файлу закрытого SSL-ключа. Поведение свойства: - *поддерживается*, если `type` задан как ”HTTP agent” - *только для чтения* для унаследованных объектов
ssl_key_password	string	Пароль для файла SSL-ключа. Поведение свойства: - *поддерживается*, если `type` задан как ”HTTP agent” - *только для чтения* для унаследованных объектов
status	integer	Состояние прототипа элемента данных. Возможные значения: 0 - *(по умолчанию)* включенный прототип элемента данных; 1 - отключенный прототип элемента данных; 3 - неподдерживаемый прототип элемента данных.
status_codes	string	Диапазоны требуемых кодов состояния HTTP, разделенные запятыми. Также поддерживает пользовательские макросы или макросы LLD как часть списка, разделенного запятыми. Пример: 200,200-{$M},{$M},200-400 Поведение свойства: - *поддерживается*, если `type` задан как ”HTTP agent” - *только для чтения* для унаследованных объектов
templateid	ID	ID родительского прототипа элемента данных шаблона. Поведение свойства: - *только для чтения*
timeout	string	Тайм-аут запроса опроса данных элемента данных. Принимает секунды или единицу времени с суффиксом (например, 30s, 1m). Также принимает пользовательские макросы и макросы LLD. Диапазон возможных значений: 1-600s. По умолчанию: ”” - использовать настройки прокси/глобальные настройки. Поведение свойства: - *поддерживается*, если `type` задан как ”Zabbix agent” (0), ”Simple check” (3) и `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) и `snmp_oid` начинается с ”`walk[`” или ”`get[`”, ”Script” (21), ”Browser” (22) - *только для чтения* для унаследованных объектов

Свойство	Type	Описание
trapper_hosts	string	Разрешенные узлы сети. Поведение свойства: - *поддерживается*, если `type` задан как ”Zabbix trapper”, либо если `type` задан как ”HTTP agent” и `allow_traps` задан как ”Allow to accept incoming data”
trends	string	Единица времени, определяющая, как долго должны храниться данные трендов. Также принимает пользовательский макрос и макрос LLD. По умолчанию: 365d. Поведение свойства: - *поддерживается*, если `value_type` задан как ”numeric float” или ”numeric unsigned”
units	string	Единицы измерения значения. Поведение свойства: - *поддерживается*, если `value_type` задан как ”numeric float” или ”numeric unsigned” - *только для чтения* для унаследованных объектов
username	string	Имя пользователя для аутентификации. Поведение свойства: - *обязательно*, если `type` задан как ”SSH agent” или ”TELNET agent”, либо если `type` задан как ”JMX agent” и задан `password` - *поддерживается*, если `type` задан как ”Simple check”, ”Database monitor” или ”HTTP agent” - *только для чтения* для унаследованных объектов (если `type` задан как ”HTTP agent”)
uuid	string	Универсальный уникальный идентификатор, используемый для связывания импортированных прототипов элементов данных с уже существующими. Генерируется 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` задан как ”numeric float”, ”character” или ”numeric unsigned” - *только для чтения* для унаследованных объектов
verify_host	integer	Следует ли проверять, что имя узла сети для соединения совпадает с именем в сертификате узла сети. Возможные значения: 0 - *(по умолчанию)* Не проверять; 1 - Проверять. Поведение свойства: - *поддерживается*, если `type` задан как ”HTTP agent” - *только для чтения* для унаследованных объектов

Свойство	Type	Описание
verify_peer	integer	Следует ли проверять подлинность сертификата узла сети. Возможные значения: 0 - *(по умолчанию)* Не проверять; 1 - Проверять. Поведение свойства: - *поддерживается*, если `type` задан как "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 не будут обнаруживаться, а существующие элементы данных будут помечены как потерянные.

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*

Тег прототипа элемента данных

Объект тега прототипа элемента данных имеет следующие свойства.

Свойство	Тип	Описание
tag	string	Имя тега прототипа элемента данных. Поведение свойства: - *обязательно*
value	string	Значение тега прототипа элемента данных. Поведение свойства: - *только для чтения*.
object	integer	Тип объекта, от которого был унаследован тег. Возможные значения: 0 - шаблон; 1 - узел сети. Поведение свойства: - *только для чтения*.
objectid	ID	ID объекта, от которого был унаследован тег. Поведение свойства: - *только для чтения*.

Предобработка прототипа элемента данных

Объект предобработки прототипа элемента данных имеет следующие свойства.

Свойство	Тип	Описание
type	integer	Тип варианта предобработки. Возможные значения: 1 - Пользовательский множитель; 2 - Обрезка справа; 3 - Обрезка слева; 4 - Обрезка; 5 - Регулярное выражение; 6 - Логическое значение в десятичное; 7 - Восьмеричное в десятичное; 8 - Шестнадцатеричное 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 ошибки в JSON; 17 - Проверка ошибки в XML; 18 - Проверка ошибки с использованием регулярного выражения; 19 - Отбрасывать без изменений; 20 - Отбрасывать без изменений с контрольным интервалом; 21 - JavaScript; 22 - Шаблон Prometheus; 23 - Prometheus в JSON; 24 - CSV в JSON; 25 - Замена; 26 - Проверка неподдерживаемости; 27 - XML в JSON; 28 - Значение SNMP walk; 29 - SNMP walk в JSON; 30 - Значение SNMP get. Поведение свойства: - *обязательно*
params	string	Дополнительные параметры, используемые вариантом предобработки. Несколько параметров разделяются символом новой строки (\n). Если `type` установлен в ”Проверка неподдерживаемости”, параметры следуют синтаксису `<scope>[\n<pattern>]`, где *pattern* — это регулярное выражение, а *scope* — одно из следующих значений: -1 - соответствует любой ошибке; 0 - проверить, соответствует ли сообщение об ошибке *pattern*; 1 - проверить, не соответствует ли сообщение об ошибке *pattern*. Поведение свойства: - *обязательно*, если `type` установлен в ”Пользовательский множитель” (1), ”Обрезка справа” (2), ”Обрезка слева” (3), ”Обрезка” (4), ”Регулярное выражение” (5), ”XML XPath” (11), ”JSONPath” (12), ”В диапазоне” (13), ”Соответствует регулярному выражению” (14), ”Не соответствует регулярному выражению” (15), ”Проверка ошибки в JSON” (16), ”Проверка ошибки в XML” (17), ”Проверка ошибки с использованием регулярного выражения” (18), ”Отбрасывать без изменений с контрольным интервалом” (20), ”JavaScript” (21), ”Шаблон Prometheus” (22), ”Prometheus в JSON” (23), ”CSV в JSON” (24), ”Замена” (25), ”Проверка неподдерживаемости” (26), ”Значение SNMP walk” (28), ”SNMP walk в JSON” (29) или ”Значение SNMP get” (30)

Свойство	Тип	Описание
error_handler	integer	Тип действия, используемого 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 type установлен в ”Проверка неподдерживаемости”: 1 - Отбросить значение; 2 - Установить пользовательское значение; 3 - Установить пользовательское сообщение об ошибке. Поведение свойства: - *обязательно*, если type установлен в ”Пользовательский множитель” (1), ”Регулярное выражение” (5), ”Логическое значение в десятичное” (6), ”Восьмеричное в десятичное” (7), ”Шестнадцатеричное 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 ошибки в JSON” (16), ”Проверка ошибки в XML” (17), ”Проверка ошибки с использованием регулярного выражения” (18), ”Шаблон Prometheus” (22), ”Prometheus в JSON” (23), ”CSV в JSON” (24), ”Проверка неподдерживаемости” (26), ”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: - *обязательно*, если error_handler установлен в ”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	Обрезка справа	список символов[2]			
3	Обрезка слева	список символов[2]			
4	Обрезка	список символов[2]			
5	Регулярное выражение	шаблон[3]	вывод[2]		0, 1, 2, 3
6	Логическое значение в десятичное				0, 1, 2, 3
7	Восьмеричное в десятичное				0, 1, 2, 3
8	Шестнадцатеричное в десятичное				0, 1, 2, 3

Тип предобработки	Название	Параметр 1	Параметр 2	Параметр 3	Поддерживаемые обработчики ошибок
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	min[1, 6]	max[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
16	Проверка на ошибки в JSON	путь[4]			0, 1, 2, 3
17	Проверка на ошибки в XML	путь[4]			0, 1, 2, 3
18	Проверка на ошибки с использованием регулярного выражения	шаблон[3]	вывод[2]		0, 1, 2, 3
19	Отбрасывать без изменений				
20	Отбрасывать без изменений с контрольным интервалом	секунды[5, 6]			
21	JavaScript	скрипт[2]			
22	Шаблон Prometheus	шаблон[6, 7]	`value`, `label`, `function`	вывод[8, 9]	0, 1, 2, 3
23	Prometheus в JSON	шаблон[6, 7]			0, 1, 2, 3
24	CSV в JSON	символ[2]	символ[2]	0,1	0, 1, 2, 3
25	Замена	строка поиска[2]	замена[2]		
26	Проверка на неподдерживаемости	сторе[1]	шаблон[3, 6]		1, 2, 3
27	XML в JSON				0, 1, 2, 3

Тип предобработки	Название	Параметр 1	Параметр 2	Параметр 3	Поддерживаемые обработчики ошибок
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[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, следующий 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] Вывод Prometheus, следующий синтаксису: `<label name>` (может быть пользовательским макросом или макросом LLD), если `label` выбран в качестве второго параметра.
[9] Одна из агрегирующих функций: `sum`, `min`, `max`, `avg`, `count`, если `function` выбран в качестве второго параметра.
[10] Поддерживает несколько записей "Имя поля,Префикс OID,Формат", разделенных символом новой строки.

itemprototype.create

Описание

`object itemprototype.create( / itemPrototypes)`

Этот метод позволяет создавать новые прототипы элементов данных.

> **Note:**
> Этот метод доступен только пользователям с типом доступа *Администратор* и *Супер-администратор*. Разрешения на вызов метода можно отозвать в настройках ролей пользователя. Дополнительную информацию см. в разделе Роли пользователей.

Параметры

`(object/array)` Прототип элемента данных для создания.

В дополнение к стандартным свойствам прототипа элемента данных, метод принимает следующие параметры.

Параметр	Type	Описание
ruleid	ID	ID правила LLD, к которому принадлежит элемент данных. Поведение параметра: - *обязательный*
preprocessing	array	Параметры предобработки прототипа элемента данных.
tags	array	Теги прототипа элемента данных.

Возвращаемые значения

( ) 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 данных должны быть числовыми элементами данных агента Zabbix с обновлением каждые 30 секунд.

Запрос:

```
{
    "jsonrpc": "2.0",
    "method": "itemprototype.create",
    "params": {
        "name": "                {#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 предобработкой

Создайте элемент данных, используя изменение в секунду и пользовательский множитель в качестве второго шага.

Запрос:

```
{
    "jsonrpc": "2.0",
    "method": "itemprototype.create",
    "params": {
        "name": "Incoming network traffic on {#IFNAME}",
        "key_": "net.if.in[{#IFNAME}]",
        "hostid": "10001",
        "ruleid": "27665",
        "type": 0,
        "value_type": 3,
```

```
        "delay": "60s",
        "units": "bps",
        "interfaceid": "1155",
        "preprocessing": [
            {
                "type": 10,
                "params": "",
                "error_handler": 0,
                "error_handler_params": ""
            },
            {
                "type": 1,
                "params": "8",
                "error_handler": 2,
                "error_handler_params": "10"
            }
        ]
    },
    "id": 1
}
```

Ответ:

```
{
    "jsonrpc": "2.0",
    "result": {
        "itemids": [
            "44211"
        ]
    },
    "id": 1
}
```

Создание прототипа зависимого элемента данных

Создайте прототип зависимого элемента данных для прототипа основного элемента данных с ID 44211. Допускаются зависимости только на том же узле сети (шаблоне/правиле обнаружения), поэтому основной и зависимый элемент данных 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, использующим пользовательский макрос, полями запроса и пользовательскими заголовками.

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 со скриптом

Создайте простой сбор данных с использованием прототипа элемента данных со скриптом.

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(    itemPrototypeIds)`

Этот метод позволяет удалять прототипы элементов данных.

> **Note:**
> Этот метод доступен только пользователям с типом доступа *Администратор* и *Супер-администратор*. Разрешения на вызов метода можно отозвать в настройках ролей 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 удалён основной элемент данных 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(          )`

Этот метод позволяет получать прототипы элементов данных в соответствии с заданными параметрами.

> **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)` Параметры, определяющие желаемый вывод.

Метод поддерживает следующие параметры.

Параметр	Тип	Описание
discoveryids	ID/array	Возвращать только прототипы элементов данных, принадлежащие указанным правилам LLD.
graphids	ID/array	Возвращать только прототипы элементов данных, используемые в указанных прототипах графиков.
hostids	ID/array	Возвращать только прототипы элементов данных, принадлежащие указанным узлам сети.
inherited	boolean	Если установлено значение `true`, возвращать только прототипы элементов данных, унаследованные от шаблона.
itemids	ID/array	Возвращать только прототипы элементов данных с указанными ID.
monitored	boolean	Если установлено значение `true`, возвращать только включенные прототипы элементов данных, принадлежащие отслеживаемым узлам сети.
templated	boolean	Если установлено значение `true`, возвращать только прототипы элементов данных, принадлежащие шаблонам.
templateids	ID/array	Возвращать только прототипы элементов данных, принадлежащие указанным шаблонам.
triggerids	ID/array	Возвращать только прототипы элементов данных, используемые в указанных прототипах триггеров.

Параметр	Тип	Описание
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. 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 обнаруживается; `ts_delete` - (`timestamp`) время, когда прототип элемента данных, который больше не обнаруживается, будет удален; `ts_disable` - (`timestamp`) время, когда прототип элемента данных, который больше не обнаруживается, будет отключен; `disable_source` - (`int`) индикатор 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, которому принадлежит прототип элемента данных.
selectDiscoveryRulePrototype	query	Возвращать свойство `discoveryRulePrototype` с родительским прототипом правила LLD, 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, которым принадлежит прототип элемента данных.
selectInheritedTags	query	Возвращать свойство `inheritedTags` с тегами, находящимися 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 предобработки 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 одиночное 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 элемента данных.
limitSelects	integer	Ограничивает количество записей, возвращаемых подзапросами. Применяется к следующим подзапросам: `selectGraphs` - результаты будут отсортированы по `name`; `selectTriggers` - результаты будут отсортированы по `description`.

Параметр	Тип	Описание
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": "The processor load is calculated as system CPU load divided by number of CPU c
            "inventory_link": "0",
            "lifetime": "7d",
            "evaltype": "0",
            "jmx_endpoint": "",
            "master_itemid": "0",
            "timeout": "",
            "url": "",
            "query_fields": [],
            "posts": "",
            "status_codes": "200",
            "follow_redirects": "1",
            "post_type": "0",
            "http_proxy": "",
            "headers": [],
            "retrieve_mode": "0",
            "request_method": "0",
            "output_format": "0",
            "ssl_cert_file": "",
            "ssl_key_file": "",
            "ssl_key_password": "",
            "verify_peer": "0",
            "verify_host": "0",
            "allow_traps": "0",
            "discover": "0",
            "uuid": "",
            "lifetime_type": "0",
            "enabled_lifetime_type": "2",
            "enabled_lifetime": "0",
            "parameters": []
        }
    ],
    "id": 1
}
```


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
}
```


Найти прототип элемента данных HTTP agent

Найти 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(  /  itemPrototypes)`

Этот метод позволяет обновлять существующие прототипы элементов данных.

> **Note:**
> Этот метод доступен только пользователям с типом доступа *Администратор* и *Супер-администратор*. Разрешения на вызов метода можно отозвать в настройках ролей пользователя. Дополнительную информацию см. в разделе Роли пользователей.

Параметры

`(object/array)` Свойства прототипа элемента данных, которые необходимо обновить.

Свойство `itemid` должно быть определено для каждого прототипа элемента данных, все остальные свойства являются необязательными. Будут обновлены только переданные свойства, все остальные останутся без изменений.

В дополнение к стандартным свойствам прототипа элемента данных, метод принимает следующие параметры.

Параметр	Type	Описание
preprocessing	array	Параметры предобработки прототипа элемента данных для замены текущих параметров предобработки. Поведение параметра: - *только для чтения* для унаследованных 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

Обновите прототип зависимого элемента данных, указав новый ID прототипа главного элемента данных. Допускаются только зависимости на одном и том же узле сети (шаблоне/правиле обнаружения), поэтому главный и зависимый элементы 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 agent

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 предобработки элемента данных

Обновите прототип элемента данных с правилом предобработки элемента данных «Пользовательский множитель».

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 script

Обновите прототип элемента данных script другим скриптом 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.

Свойство	Тип	Описание
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
test_string	string	Тестовая строка.

Выражения

Объект выражений имеет следующие свойства.

Свойство	Тип	Описание
expression	string	Регулярное выражение. Поведение свойства: - *обязательно*
expression_type	integer	Тип регулярного выражения. Возможные значения: 0 - Строка символов включена; 1 - Включена любая строка символов; 2 - Строка символов не включена; 3 - Результат TRUE; 4 - Результат FALSE. Поведение свойства: - *обязательно*
exp_delimiter	string	Разделитель выражения. Значение по умолчанию: `","`. Возможные значения: `","`, `"."` или `"/"`. Поведение свойства: - *поддерживается*, если `expression_type` установлен в "Включена любая строка символов"
case_sensitive	integer	Чувствительность к регистру. Значение по умолчанию: 0. Возможные значения: 0 - Без учета регистра; 1 - С учетом регистра.

regexp.create

Описание

`object regexp.create(  /  regularExpressions)`

Этот метод позволяет создавать новые глобальные регулярные выражения.

> **Note:**
> Этот метод доступен только пользователю с типом доступа *Супер-администратор*. Разрешения на вызов метода можно отозвать в настройках ролей пользователя. Дополнительную информацию см. в разделе Роли пользователей.

Параметры

`(object/array)` Регулярные выражения для создания.

В дополнение к стандартным свойствам, метод принимает следующие параметры.

Параметр	Type	Описание
expressions	array	Параметры выражений. Поведение параметра: - *обязательный*

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",
      "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": [
            "16"
        ]
    },
    "id": 1
}
```

Источник

CRegexp::create() в *ui/include/classes/api/services/CRegexp.php*.

regexp.delete

Описание

`object regexp.delete(    regexpids)`

Этот метод позволяет удалять глобальные регулярные выражения.

> **Note:**
> Этот метод доступен только пользователю с типом доступа *Супер-администратор*. Разрешения на вызов метода можно отозвать в настройках ролей пользователя. Дополнительную информацию см. в разделе Роли пользователей.

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(          )`

Метод позволяет получать глобальные регулярные выражения по заданным параметрам.

Note:
Этот метод доступен только пользователю с типом доступа *Супер-администратор*. Разрешения на вызов метода можно отозвать в настройках ролей пользователя. Дополнительную информацию см. в разделе Роли пользователей.

Параметры

`(object)` Параметры, определяющие желаемый вывод.

Метод поддерживает следующие параметры.

Параметр	Тип	Описание
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"],
        "selectExpressions": ["expression", "expression_type"],
        "regexpids": [1, 2],
        "preservekeys": true
    },
    "id": 1
}
```

Ответ:

```
{
  "jsonrpc": "2.0",
  "result": {
    "1": {
      "regexpid": "1",
      "name": "                    ",
      "expressions": [
        {
          "expression": "^(btrfs|ext2|ext3|ext4|reiser|xfs|ffs|ufs|jfs|jfs2|vxfs|hfs|apfs|refs|ntfs|fat32|
          "expression_type": "3"
        }
      ]
    },
    "2": {
      "regexpid": "2",
      "name": "                     ",
      "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( / RegularExpressions)`

Этот метод позволяет обновлять существующие глобальные регулярные выражения.

Note:
Этот метод доступен только пользователю с типом доступа *Супер-администратор*. Разрешения на вызов метода можно отозвать в настройках ролей пользователя. Дополнительную информацию см. в разделе Роли пользователей.

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.

Параметр	Тип	Описание
expressions	array	Параметры выражений.

Возвращаемые значения

`( )` Возвращает объект, который содержит ID обновленных регулярных выражений, указанных в свойстве`regexpids`.

Примеры

Обновление глобального регулярного выражения для обнаружения файловых систем.

Запрос:

```
{
    "jsonrpc": "2.0",
    "method": "regexp.update",
    "params": {
      "regexpid": "1",
      "name": "File systems for discovery",
      "test_string": "",
      "expressions": [
        {
          "expression": "^(btrfs|ext2|ext3|ext4|reiser|xfs|ffs|ufs|jfs|jfs2|vxfs|hfs|apfs|refs|zfs)$",
          "expression_type": "3",
          "exp_delimiter": ",",
          "case_sensitive": "0"
        },
        {
          "expression": "^(ntfs|fat32|fat16)$",
          "expression_type": "3",
          "exp_delimiter": ",",
          "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
- Правила роли
  - Элемент пользовательского интерфейса
  - Сервис
  - Тег сервиса
  - Модуль
  - Действие

Доступные методы:

- role.create - создать новые роли пользователей
- role.delete - удалить роли пользователей
- role.get - получить роли пользователей
- role.update - обновить роли пользователей

Объект role

Следующие объекты напрямую связаны с API `role`.

Роль

Объект роли имеет следующие свойства:

Свойство	Тип	Описание
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 - Супер-администратор. Поведение свойства: - *обязательно* для операций создания

Свойство	Тип	Описание
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	Type	Описание
ui	array	Массив объектов элемента пользовательского интерфейса.
ui.default_access	integer	Включен ли доступ к новым элементам пользовательского интерфейса. Возможные значения: 0 - Отключено; 1 - *(по умолчанию)* Включено.
services.read.mode	integer	Доступ только на чтение к сервисам. Возможные значения: 0 - Доступ только на чтение к сервисам, указанным в свойстве `services.read.list` или соответствующим свойству `services.read.tag`; 1 - *(по умолчанию)* Доступ только на чтение ко всем сервисам.
services.read.list	array	Массив объектов Service. Указанным сервисам, включая дочерние сервисы, будет предоставлен доступ только на чтение для роли пользователя. Доступ только на чтение не переопределяет доступ на чтение и запись к сервисам. Поведение свойства: - *поддерживается*, если `services.read.mode` установлено в ”0”
services.read.tag	object	Массив объектов тега сервиса. Сервисам, соответствующим тегу, включая дочерние сервисы, будет предоставлен доступ только на чтение для роли пользователя. Доступ только на чтение не переопределяет доступ на чтение и запись к сервисам. Поведение свойства: - *поддерживается*, если `services.read.mode` установлено в ”0”
services.write.mode	integer	Доступ на чтение и запись к сервисам. 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.

Свойство	Type	Описание
services.write.list	array	Массив объектов Service. Указанным сервисам, включая дочерние сервисы, будет предоставлен доступ на чтение и запись для роли пользователя. Доступ на чтение и запись переопределяет доступ только на чтение к сервисам. Поведение свойства: - *поддерживается*, если `services.write.mode` установлено в "0"
services.write.tag	object	Массив объектов тега сервиса. Сервисам, соответствующим тегу, включая дочерние сервисы, будет предоставлен доступ на чтение и запись для роли пользователя. Доступ на чтение и запись переопределяет доступ только на чтение к сервисам. Поведение свойства: - *поддерживается*, если `services.write.mode` установлено в "0"
modules	array	Массив объектов модуля.
modules.default_access	integer	Включен ли доступ к новым модулям. Возможные значения: 0 - Отключено; 1 - *(по умолчанию)* Включено.
api.access	integer	Включен ли доступ к API. Возможные значения: 0 - Отключено; 1 - *(по умолчанию)* Включено.
api.mode	integer	Режим обработки методов API, перечисленных в свойстве `api`. Возможные значения: 0 - *(по умолчанию)* Список запрещённых; 1 - Список разрешённых.
api	array	Массив методов API.
actions	array	Массив объектов операции.
actions.default_access	integer	Включен ли доступ к новым операциям. Возможные значения: 0 - Отключено; 1 - *(по умолчанию)* Включено.

Элемент пользовательского интерфейса

Объект элемента пользовательского интерфейса имеет следующие свойства:

Свойство	Тип	Описание
name	string	Имя элемента пользовательского интерфейса. Возможные значения, если type объекта Role задан как "User", "Admin" или "Super admin": monitoring.dashboard - *Панели*; 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 type объекта Role задан как "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 триггеров*; configuration.service_actions - *Оповещения → Действия → Действия сервисов*; configuration.discovery_actions - *Оповещения → Действия → Действия обнаружения*; configuration.autoregistration_actions - *Оповещения → Действия → Действия авторегистрации*; configuration.internal_actions - *Оповещения → Действия → Внутренние действия*. Возможные значения, если type объекта Role задан как "Super admin": reports.system_info - *Отчеты → Информация о системе*; reports.audit - *Отчеты → Журнал аудита*; reports.action_log - *Отчеты → Журнал действий*; configuration.event_correlation - *Сбор данных → Корреляция событий*; administration.media_types - *Оповещения → Способы оповещения*; 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.general - *Администрирование → Общие*; administration.audit_log - *Администрирование → Журнал аудита*; administration.housekeeping - *Администрирование → Очистка истории*;

Свойство	Тип	Описание
status	integer	Включен ли доступ к элементу пользовательского интерфейса. Возможные значения: 0 - Отключено; 1 - *(по умолчанию)* Включено.

Сервис

Свойство	Тип	Описание
serviceid	ID	ID сервиса. Поведение свойства: - *обязательно*

Тег сервиса

Свойство	Type	Описание
tag	string	Имя тега. Если указана пустая строка, тег сервиса не будет использоваться для сопоставления сервисов. Поведение свойства: - *обязательно*
value	string	Значение тега. Если значение не указано или указана пустая строка, для сопоставления сервисов будет использоваться только имя тега.

Модуль

Объект модуля имеет следующие свойства:

Свойство	Тип	Описание
moduleid	ID	ID модуля. Поведение свойства: - *обязательно*
status	integer	Включен ли доступ к модулю. Возможные значения: 0 - Отключено; 1 - *(по умолчанию)* Включено.

Действие

Объект действия имеет следующие свойства:

Свойство	Тип	Описание
name	string	Имя действия. Возможные значения, если `type` объекта Role установлен в "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 скриптов; `manage_api_tokens` - управление API-токенами; `change_problem_ranking` - изменение ранга проблемы с причины на симптом и наоборот; `edit_own_media` - разрешить создание/редактирование собственных способов оповещения. Возможные значения, если `type` объекта Role установлен в "Admin" или "Super admin": `edit_maintenance` - создание и редактирование периодов обслуживания; `manage_scheduled_reports` - управление запланированными отчетами, `manage_sla` - управление SLA. Возможные значения, если `type` объекта Role установлен в "User" или "Admin": `invoke_execute_now` - позволяет выполнять проверки элементов данных для пользователей, у которых есть только права на чтение узла сети. Возможные значения, если `type` объекта Role установлен в "Super admin": `edit_user_media` - разрешить создание/редактирование способов оповещения для пользователей. Поведение свойства: - *обязательно*
status	integer	Включен ли доступ на выполнение действия. Возможные значения: 0 - Отключено; 1 - *(по умолчанию)* Включено.

role.create

Описание

`object role.create( / roles)`

Этот метод позволяет создавать новые роли.

> **Note:**
> Этот метод доступен только пользователю с типом доступа *Супер-администратор*. Разрешения на вызов метода можно отозвать в настройках ролей пользователя. Дополнительную информацию см. в разделе Роли пользователей.

Параметры

`(object/array)` Роли для создания.

В дополнение к стандартным свойствам роли, метод принимает следующие параметры.

Параметр	Тип	Описание
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 типа "User" с запрещённым доступом к двум элементам пользовательского 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 пользовательского интерфейса
- Модуль
- Действие

Источник

CRole::create() в *ui/include/classes/api/services/CRole.php*.

role.delete

Описание

`object role.delete( roleids)`

Этот метов позволяет удалить роли.

> **Note:**
> Этот метод доступен только пользователю с типом доступа *Супер-администратор*. Разрешения на вызов метода можно отозвать в настройках ролей пользователя. Дополнительную информацию см. в разделе Роли пользователей.

Параметры

`( )` ID удаляемых ролей.

Возвращаемые значения

`(object)` Возвращает объект, содержащий идентификаторы удаленных ролей, указанных в свойстве `roleids`.

Примеры

Удаление нескольких ролей пользователей

Удалите две роли 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( )`

Этот метод позволяет получать роли в соответствии с заданными параметрами.

> **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)` Параметры, определяющие желаемый вывод.

Метод поддерживает следующие параметры.

Параметр	Тип	Описание
roleids	ID/array	Возвращать только роли с указанными ID.
selectRules	query	Возвращать свойство `rules` с правилами роли.
selectUsers	query	Возвращать свойство `users` с пользователями, которым назначена роль. Информацию об ограничениях в зависимости от типа пользователя смотрите в `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	
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 ”Super admin role” и её правила 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": "Super admin role",
            "type": "3",
            "readonly": "1",
            "rules": {
                "ui": [
                    {
                        "name": "monitoring.dashboard",
                        "status": "1"
                    },
```

```
                {
                    "name": "monitoring.problems",
                    "status": "1"
                },
                {
                    "name": "monitoring.hosts",
                    "status": "1"
                },
                {
                    "name": "monitoring.latest_data",
                    "status": "1"
                },
                {
                    "name": "monitoring.maps",
                    "status": "1"
                },
                {
                    "name": "services.services",
                    "status": "1"
                },
                {
                    "name": "services.sla_report",
                    "status": "1"
                },
                {
                    "name": "inventory.overview",
                    "status": "1"
                },
                {
                    "name": "inventory.hosts",
                    "status": "1"
                },
                {
                    "name": "reports.availability_report",
                    "status": "1"
                },
                {
                    "name": "reports.top_triggers",
                    "status": "1"
                },
                {
                    "name": "monitoring.discovery",
                    "status": "1"
                },
                {
                    "name": "services.sla",
                    "status": "1"
                },
                {
                    "name": "reports.scheduled_reports",
                    "status": "1"
                },
                {
                    "name": "reports.notifications",
                    "status": "1"
                },
                {
                    "name": "configuration.template_groups",
                    "status": "1"
                },
                {
                    "name": "configuration.host_groups",
```

```
            "status": "1"
        },
        {
            "name": "configuration.templates",
            "status": "1"
        },
        {
            "name": "configuration.hosts",
            "status": "1"
        },
        {
            "name": "configuration.maintenance",
            "status": "1"
        },
        {
            "name": "configuration.discovery",
            "status": "1"
        },
        {
            "name": "configuration.trigger_actions",
            "status": "1"
        },
        {
            "name": "configuration.service_actions",
            "status": "1"
        },
        {
            "name": "configuration.discovery_actions",
            "status": "1"
        },
        {
            "name": "configuration.autoregistration_actions",
            "status": "1"
        },
        {
            "name": "configuration.internal_actions",
            "status": "1"
        },
        {
            "name": "reports.system_info",
            "status": "1"
        },
        {
            "name": "reports.audit",
            "status": "1"
        },
        {
            "name": "reports.action_log",
            "status": "1"
        },
        {
            "name": "configuration.event_correlation",
            "status": "1"
        },
        {
            "name": "administration.media_types",
            "status": "1"
        },
        {
            "name": "administration.scripts",
            "status": "1"
        },
```

```
            {
                "name": "administration.user_groups",
                "status": "1"
            },
            {
                "name": "administration.user_roles",
                "status": "1"
            },
            {
                "name": "administration.users",
                "status": "1"
            },
            {
                "name": "administration.api_tokens",
                "status": "1"
            },
            {
                "name": "administration.authentication",
                "status": "1"
            },
            {
                "name": "administration.general",
                "status": "1"
            },
            {
                "name": "administration.audit_log",
                "status": "1"
            },
            {
                "name": "administration.housekeeping",
                "status": "1"
            },
            {
                "name": "administration.proxy_groups",
                "status": "1"
            },
            {
                "name": "administration.proxies",
                "status": "1"
            },
            {
                "name": "administration.macros",
                "status": "1"
            },
            {
                "name": "administration.queue",
                "status": "1"
            }
        ],
        "ui.default_access": "1",
        "services.read.mode": "1",
        "services.read.list": [],
        "services.read.tag": {
            "tag": "",
            "value": ""
        },
        "services.write.mode": "1",
        "services.write.list": [],
        "services.write.tag": {
            "tag": "",
            "value": ""
        },
```

```
            "modules": [
                {
                    "moduleid": 1,
                    "status": "1"
                },
                {
                    "moduleid": 2,
                    "status": "1"
                },
                {
                    "moduleid": 3,
                    "status": "1"
                },
                {
                    "moduleid": 4,
                    "status": "1"
                },
                {
                    "moduleid": 5,
                    "status": "1"
                },
                {
                    "moduleid": 6,
                    "status": "1"
                },
                {
                    "moduleid": 7,
                    "status": "1"
                },
                {
                    "moduleid": 8,
                    "status": "1"
                },
                {
                    "moduleid": 9,
                    "status": "1"
                },
                {
                    "moduleid": 10,
                    "status": "1"
                },
                {
                    "moduleid": 11,
                    "status": "1"
                },
                {
                    "moduleid": 12,
                    "status": "1"
                },
                {
                    "moduleid": 13,
                    "status": "1"
                },
                {
                    "moduleid": 14,
                    "status": "1"
                },
                {
                    "moduleid": 15,
                    "status": "1"
                },
                {
```

```
            "moduleid": 16,
            "status": "1"
        },
        {
            "moduleid": 17,
            "status": "1"
        },
        {
            "moduleid": 18,
            "status": "1"
        },
        {
            "moduleid": 19,
            "status": "1"
        },
        {
            "moduleid": 20,
            "status": "1"
        },
        {
            "moduleid": 21,
            "status": "1"
        },
        {
            "moduleid": 22,
            "status": "1"
        },
        {
            "moduleid": 23,
            "status": "1"
        },
        {
            "moduleid": 24,
            "status": "1"
        },
        {
            "moduleid": 25,
            "status": "1"
        },
        {
            "moduleid": 26,
            "status": "1"
        },
        {
            "moduleid": 27,
            "status": "1"
        },
        {
            "moduleid": 28,
            "status": "1"
        },
        {
            "moduleid": 29,
            "status": "1"
        },
        {
            "moduleid": 30,
            "status": "1"
        },
        {
            "moduleid": 31,
            "status": "1"
```

```
            },
            {
                "moduleid": 32,
                "status": "1"
            },
            {
                "moduleid": 33,
                "status": "1"
            }
        ],
        "modules.default_access": "1",
        "api.access": "1",
        "api.mode": "0",
        "api": [],
        "actions": [
            {
                "name": "edit_dashboards",
                "status": "1"
            },
            {
                "name": "edit_maps",
                "status": "1"
            },
            {
                "name": "acknowledge_problems",
                "status": "1"
            },
            {
                "name": "suppress_problems",
                "status": "1"
            },
            {
                "name": "close_problems",
                "status": "1"
            },
            {
                "name": "change_severity",
                "status": "1"
            },
            {
                "name": "add_problem_comments",
                "status": "1"
            },
            {
                "name": "execute_scripts",
                "status": "1"
            },
            {
                "name": "manage_api_tokens",
                "status": "1"
            },
            {
                "name": "edit_maintenance",
                "status": "1"
            },
            {
                "name": "manage_scheduled_reports",
                "status": "1"
            },
            {
                "name": "manage_sla",
                "status": "1"
```

```
                    },
                    {
                        "name": "invoke_execute_now",
                        "status": "1"
                    },
                    {
                        "name": "change_problem_ranking",
                        "status": "1"
                    },
                    {
                        "name": "edit_own_media",
                        "status": "1"
                    },
                    {
                        "name": "edit_user_media",
                        "status": "1"
                    }
                ],
                "actions.default_access": "1"
            }
        }
    ],
    "id": 1
}
```

См. также

- Правила роли
- Пользователь

Источник

CRole::get() в *ui/include/classes/api/services/CRole.php*.

role.update

Описание

`object role.update(  /  roles)`

Этот метод позволяет обновлять существующие роли.

Note:
Этот метод доступен только пользователю с типом доступа *Супер-администратор*. Разрешения на вызов метода можно отозвать в настройках ролей пользователя. Дополнительную информацию см. в разделе Роли пользователей.

Параметры

`(object/array)` Свойства роли, которые необходимо обновить.

Свойство `roleid` должно быть определено для каждой роли, все остальные свойства являются необязательными. Будут обновлены только переданные свойства, все остальные останутся без изменений.

В дополнение к стандартным свойствам роли метод принимает следующие параметры.

Параметр	Тип	Описание
rules	array	Правила доступа rules, которые необходимо обновить для роли.

Возвращаемые значения

`(object)` Возвращает объект, содержащий идентификаторы обновлённых ролей, указанных в свойстве `roleids`.

Примеры

Отключение возможности выполнять скрипты

Обновите 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

Обновите 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/бизнес-сервисами.

Ссылки на объекты:

- Сервис
- Правило статуса
- Тег сервиса
- Оповещение сервиса
- Тег проблемы

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 сервиса имеет следующие свойства.

Свойство	Тип	Описание
serviceid	ID	ID сервиса. Поведение свойства: - *только для чтения* - *обязательно* для операций обновления
algorithm	integer	Правило вычисления статуса. Применяется только при наличии дочерних сервисов. 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.

Свойство	Тип	Описание
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 значение `propagation_value` (на 1–5 уровней серьезности); 2 - уменьшить распространяемый статус на заданное значение `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 значение `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 одному из следующих значений: - серьезности наиболее критичной проблемы; - наивысшему статусу дочернего сервиса, находящегося 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. Генерируется автоматически, если не указан.
created_at	integer	Unix timestamp, когда сервис был создан.

Свойство	Тип	Описание
readonly	integer	Доступ к сервису. Возможные значения: 0 - Чтение-запись; 1 - Только чтение. Поведение свойства: - *только для чтения*

Правило статуса

Объект правила статуса имеет следующие свойства.

Свойство	Type	Описание
type	integer	Условие для установки статуса (Новый статус). Возможные значения: 0 - если как минимум (N) дочерних сервисов имеют статус (Статус) или выше; 1 - если как минимум (N%) дочерних сервисов имеют статус (Статус) или выше; 2 - если менее (N) дочерних сервисов имеют статус (Статус) или ниже; 3 - если менее (N%) дочерних сервисов имеют статус (Статус) или ниже; 4 - если вес дочерних сервисов со статусом (Статус) или выше составляет как минимум (W); 5 - если вес дочерних сервисов со статусом (Статус) или выше составляет как минимум (N%); 6 - если вес дочерних сервисов со статусом (Статус) или ниже меньше (W); 7 - если вес дочерних сервисов со статусом (Статус) или ниже меньше (N%). Где: - N (W) — это `limit_value`; - (Статус) — это `limit_status`; - (Новый статус) — это `new_status`. Поведение свойства: - *required*
limit_value	integer	Предельное значение. Возможные значения: - для N и W: 1-100000; - для N%: 1-100. Поведение свойства: - *required*

Свойство	Type	Описание
limit_status	integer	Предельный статус. Возможные значения: -1 - OK; 0 - Не классифицирован; 1 - Информация; 2 - Предупреждение; 3 - Средний; 4 - Высокий; 5 - Катастрофа. Поведение свойства: - *required*
new_status	integer	Значение нового статуса. Возможные значения: 0 - Не классифицирован; 1 - Информация; 2 - Предупреждение; 3 - Средний; 4 - Высокий; 5 - Катастрофа. Поведение свойства: - *required*

Тег сервиса

Объект тега сервиса имеет следующие свойства.

Свойство	Type	Описание
tag	string	Имя тега сервиса. Поведение свойства: - *обязательно*
value	string	Значение тега сервиса.

Аварийное сообщение сервиса

Note:
Аварийные сообщения сервиса нельзя напрямую создавать, обновлять или удалять через Zabbix API.

Объекты аварийных сообщений сервиса представляют изменение состояния сервиса. Они имеют следующие свойства.

Свойство	Тип	Описание
clock	timestamp	Время, когда произошло изменение состояния сервиса.
value	integer	Состояние сервиса. Список возможных значений см. в `свойстве status сервиса`.

Тег проблемы

Теги проблем позволяют связывать сервисы с событиями проблем. Объект тега проблемы имеет следующие свойства.

Свойство	Type	Описание
tag	string	Имя тега проблемы. Поведение свойства: - *обязательно*
operator	integer	Условие сопоставления оператор. Возможные значения: 0 - *(по умолчанию)* Равно; 2 - Содержит.
value	string	Значение тега проблемы.

service.create

Описание

`object service.create( / services)`

Этот метод позволяет создавать новые услуги.

> **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)` сервисы для создания.

В дополнение к стандартным свойствам сервиса, метод принимает следующие параметры.

Параметр	Тип	Описание
children	array	Дочерние сервисы, которые будут связаны с сервисом. Для дочерних сервисов должно быть определено только свойство `serviceid`.
parents	array	Родительские сервисы, которые будут связаны с сервисом. Для родительских сервисов должно быть определено только свойство `serviceid`.
tags	array	Теги сервиса, которые будут созданы для сервиса.
problem_tags	array	Теги проблем, которые будут созданы для сервиса.
status_rules	array	Правила статуса, которые будут созданы для сервиса.

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 переключаться в состояние проблемы, если хотя бы один дочерний сервис имеет проблему.

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(    serviceIds)`

Этот метод позволяет удалять услуги.

> **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

(   ) ID удаляемых услуг.

Возвращаемые значения

(   ) Возвращает объект, который содержит ID удаленных услуг, указанных в свойстве `serviceids`.

Примеры

Удаление нескольких сервисов

Удалите два сервиса.

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(          )`

Метод позволяет получать услуги по заданным параметры.

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)` Параметры, определяющие желаемый вывод.

Метод поддерживает следующие параметры.

Параметр	Type	Описание
serviceids	ID/array	Возвращать только сервисы с указанными ID.
parentids	ID/array	Возвращать только сервисы, связанные с указанными родительскими сервисами.
deep_parentids	flag	Возвращать все прямые и косвенные дочерние сервисы. Используется вместе с `parentids`.
childids	ID/array	Возвращать только сервисы, связанные с указанными дочерними сервисами.
evaltype	integer	Метод вычисления тегов. Возможные значения: 0 - *(по умолчанию)* And/Or; 2 - Or.
tags	object/array	Возвращать только сервисы с указанными тегами. Формат: `[{"tag": "<tag>", "value": "<value>", "operator": "<operator>"}, ...]`. Пустой массив возвращает все сервисы. Возможные значения operator: 0 - *(по умолчанию)* Contains; 1 - Equals; 2 - Does not contain; 3 - Does not equal; 4 - Exists; 5 - Does not exist.
problem_tags	object/array	Возвращать только сервисы с указанными тегами проблем. Формат: `[{"tag": "<tag>", "value": "<value>", "operator": "<operator>"}, ...]`. Пустой массив возвращает все сервисы. Возможные значения operator: 0 - *(по умолчанию)* Contains; 1 - Equals; 2 - Does not contain; 3 - Does not equal; 4 - Exists; 5 - Does not exist.
without_problem_tags	flag	Возвращать только сервисы без тегов проблем.
slaids	ID/array	Возвращать только сервисы, связанные с указанными SLA.

Параметр	Type	Описание
selectChildren	query	Возвращать свойство `children` с дочерними сервисами. Поддерживает `count`.
selectParents	query	Возвращать свойство `parents` с родительскими сервисами. Поддерживает `count`.
selectTags	query	Возвращать свойство `tags` с тегами сервиса. Поддерживает `count`.
selectProblemEvents	query	Возвращать свойство `problem_events` с массивом объектов событий проблем. Объект события проблемы имеет следующие свойства: `eventid` - (`ID`) ID события; `severity` - (`string`) Текущая важность события; `name` - (`string`) Имя восстановленного события. Поддерживает `count`.
selectProblemTags	query	Возвращать свойство `problem_tags` с тегами проблем. Поддерживает `count`.
selectStatusRules	query	Возвращать свойство `status_rules` с правилами статуса. Поддерживает `count`.
selectStatusTimeline	object/array	Возвращать свойство `status_timeline`, содержащее изменения состояния сервиса за указанные периоды. Формат `[{"period_from": "<period_from>", "period_to": "<period_to>"}, ...]` — `period_from` является начальной датой периода (включительно; целочисленная временная метка), а `period_to` — конечной датой периода (не включительно; целочисленная временная метка), который вас интересует. Возвращает массив записей, содержащих свойство `start_value` и массив `alarms` с изменениями состояния в пределах указанных периодов.
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 услуг

Получите все данные обо всех услугах 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(  /  services)`

Этот метод позволяет обновлять существующие сервисы.

> **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) свойства сервиса, которые необходимо обновить.

Свойство serviceid должно быть определено для каждого сервиса, все остальные свойства являются необязательными. Будут обновлены только переданные свойства, все остальные останутся без изменений.

В дополнение к стандартным свойствам сервиса, метод принимает следующие параметры.

Параметр	Type	Описание
children	array	Дочерние сервисы, которые заменят текущие дочерние сервисы. Для дочерних сервисов должно быть определено только свойство serviceid.
parents	array	Родительские сервисы, которые заменят текущие родительские сервисы. Для родительских сервисов должно быть определено только свойство serviceid.
tags	array	Теги сервиса, которые заменят текущие теги сервиса.
problem_tags	array	Теги проблем, которые заменят текущие теги проблем.
status_rules	array	Правила статуса, которые заменят текущие правила статуса.

Возвращаемые значения

(object) Возвращает объект, содержащий идентификаторы обновлённых сервисов в свойстве serviceids.

Примеры

Назначение родительского сервиса для сервиса

Сделайте 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 периода простоя

Добавьте период простоя 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 - выполнить 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	ID 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` установлено в ”action operation”: 0 - Script; 1 - IPMI; 2 - SSH; 3 - TELNET; 5 - Webhook. Возможные значения, если `scope` установлено в ”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` установлено в ”Script”, ”IPMI”, ”SSH”, ”TELNET” или ”Webhook”
scope	integer	Область действия скрипта. Возможные значения: 1 - action operation; 2 - manual host action; 4 - manual event action. 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` установлено в ”Script”
menu_path	string	Папки, разделённые косой чертой, которые формируют меню-подобную навигацию в веб-интерфейсе при нажатии на узел сети или событие. Поведение свойства: - *поддерживается*, если `scope` установлено в ”manual host action” или ”manual event action”
authtype	integer	Метод аутентификации, используемый для типа скрипта SSH. Возможные значения: 0 - пароль; 1 - открытый ключ. Поведение свойства: - *поддерживается*, если `type` установлено в ”SSH”
username	string	Имя пользователя, используемое для аутентификации. Поведение свойства: - *обязательно*, если `type` установлено в ”SSH” или ”TELNET”

Свойство	Тип	Описание
password	string	Пароль, используемый для SSH-скриптов с аутентификацией по паролю и для TELNET-скриптов. Поведение свойства: - *поддерживается*, если `type` установлено в ”SSH” и `authtype` установлено в ”password”, или `type` установлено в ”TELNET”
publickey	string	Имя файла открытого ключа, используемого для SSH-скриптов с аутентификацией по открытому ключу. Поведение свойства: - *обязательно*, если `type` установлено в ”SSH” и `authtype` установлено в ”public key”
privatekey	string	Имя файла закрытого ключа, используемого для SSH-скриптов с аутентификацией по открытому ключу. Поведение свойства: - *обязательно*, если `type` установлено в ”SSH” и `authtype` установлено в ”public key”
port	string	Номер порта, используемый для SSH- и TELNET-скриптов. Поведение свойства: - *поддерживается*, если `type` установлено в ”SSH” или ”TELNET”
groupid	ID	ID группы узлов сети, на которой можно запускать скрипт. Если установлено в ”0”, скрипт будет доступен для всех групп узлов сети. По умолчанию: 0.
usrgrpid	ID	ID группы пользователей, которой будет разрешено запускать скрипт. Если установлено в ”0”, скрипт будет доступен для всех групп пользователей. По умолчанию: 0.
host_access	integer	Права доступа к узлу сети, необходимые для запуска скрипта. Возможные значения: 2 - *(по умолчанию)* чтение; 3 - запись. Поведение свойства: - *поддерживается*, если `scope` установлено в ”manual host action” или ”manual event action”
confirmation	string	Текст всплывающего окна подтверждения. Всплывающее окно появится при попытке запустить скрипт из веб-интерфейса Zabbix. Поведение свойства: - *поддерживается*, если `scope` установлено в ”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: 1-60s. По умолчанию: 30s. Поведение свойства: - *обязательно*, если `type` установлено в ”Webhook”
parameters	array	Массив входных параметров вебхука. Поведение свойства: - *поддерживается*, если `type` установлено в ”Webhook”
description	string	Описание скрипта.
url	string	URL, определённый пользователем. Поведение свойства: - *обязательно*, если `type` установлено в ”URL”
new_window	integer	Открыть URL в новом окне. Возможные значения: 0 - Нет; 1 - *(по умолчанию)* Да. Поведение свойства: - *поддерживается*, если `type` установлено в ”URL”
manualinput	integer	Указывает, принимает ли скрипт ввод, предоставляемый пользователем. Возможные значения: 0 - *(по умолчанию)* Отключено; 1 - Включено; Поведение свойства: - *поддерживается*, если `scope` установлено в ”manual host action” или ”manual event action”
manualinput_prompt	string	Текст приглашения для ручного ввода. Поведение свойства: - *обязательно*, если `manualinput` установлено в ”Enabled”
manualinput_validator	string	Строковое поле, используемое для проверки введённых пользователем данных. Строка состоит либо из регулярного выражения, либо из набора значений, разделённых запятыми. Поведение свойства: - *обязательно*, если `manualinput` установлено в ”Enabled”
manualinput_validator_type	integer	Определяет тип ожидаемого пользовательского ввода. Возможные значения: 0 - *(по умолчанию)* Строка. Указывает, что manualinput_validator следует обрабатывать как регулярное выражение; 1 - Список. Указывает, что manualinput_validator следует обрабатывать как список возможных входных значений, разделённых запятыми. Поведение свойства: - *поддерживается*, если `manualinput` установлено в ”Enabled”

Свойство	Тип	Описание
manualinput_default_value	string	Значение по умолчанию для автозаполнения пользовательского ввода. Поведение свойства: - *поддерживается*, если `manualinput_validator_type` установлено в "String"

Параметры вебхука

Параметры, передаваемые скрипту вебхука при его вызове, имеют следующие свойства.

Свойство	Тип	Описание
name	string	Имя параметра. Поведение свойства: - *обязательно*
value	string	Значение параметра. Поддерживает макросы.

Отладка

Отладочная информация выполненного скрипта вебхука. Объект debug имеет следующие свойства.

Свойство	Type	Описание
logs	array	Массив записей журнала.
ms	string	Длительность выполнения скрипта в миллисекундах.

Запись журнала

Объект записи журнала имеет следующие свойства.

Свойство	Тип	Описание
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( / scripts)`

Этот метод позволяет создавать новые сценарии.

Note:
Этот метод доступен только пользователю с типом доступа *Супер-администратор*. Разрешения на вызов метода можно отозвать в настройках ролей 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 внешнему сервису.

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 открытому 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 будет требовать права на запись на узле сети и запрашивать у пользователя ввод вручную. После успешной отправки введённых данных скрипт отобразит сообщение с подтверждением в 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 открывается в том же окне и имеет текст подтверждения.

Запрос:

```
{
    "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
}
```

Ответ:

```
{
    "jsonrpc": "2.0",
    "result": {
        "scriptids": [
            "56"
        ]
    },
    "id": 1
}
```

Создание скрипта типа URL с ручным вводом

Создайте скрипт типа URL для области события, который открывается в новом окне и использует ручной ввод.

Запрос:

```
{
    "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
}
```

Ответ:

```
{
    "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(    scriptIds)`

Этот метод позволяет удалять скрипты.

Note:
Этот метод доступен только пользователю с типом доступа *Супер-администратор*. Разрешения на вызов метода можно отозвать в настройках ролей пользователя. Дополнительную информацию см. в разделе Роли пользователей.

Параметры

( ) ID удаляемых скриптов.

Возвращаемые значения

( ) Возвращает объект, который содержит ID удаленных скриптов, указанных в свойстве `scriptids`.

Примеры

Удаление нескольких скриптов

Удалите два скрипта.

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( )`

Этот метод позволяет запустить скрипт на хосте или событии. За исключением скриптов типа URL. Они не являются исполняемыми.

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)` Параметры, содержащие ID скрипта для запуска, а также либо ID узла сети, либо ID события и значение manualinput.

Параметр	Type	Описание
scriptid	ID	ID скрипта для запуска. Поведение параметра: - *обязательный*
hostid	ID	ID узла сети, на котором нужно запустить скрипт. Поведение параметра: - *обязательный*, если `eventid` не задан
eventid	ID	ID события, для которого нужно запустить скрипт. Поведение параметра: - *обязательный*, если `hostid` не задан
manualinput	string	Значение, предоставленное пользователем для запуска скрипта, подставляемое в макрос {MANUALINPUT}.

Возвращаемые значения

`(object)` Возвращает результат выполнения скрипта.

Свойство	Type	Описание
response	string	Был ли скрипт успешно выполнен. Возможное значение — `success`.
value	string	Вывод скрипта.
debug	object	Содержит объект debug, если выполняется скрипт вебхука. Для других типов скриптов содержит пустой объект.

Примеры

Запуск скрипта вебхука

Запустите скрипт вебхука, который отправляет HTTP-запрос во внешний сервис.

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
}
```

Запуск 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.

Запрос:

```
{
    "jsonrpc": "2.0",
    "method": "script.execute",
    "params": {
        "scriptid": "7",
        "hostid": "30079",
        "manualinput": "2"
    },
    "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_seq=1 tt
        "debug": []
    },
    "id": 1
}
```

Источник

CScript::execute() в *ui/include/classes/api/services/CScript.php*.

script.get

Описание

`integer/array script.get( )`

Метод позволяет получать скрипты по заданным параметрам.

> **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)` Параметры, определяющие желаемый вывод.

Метод поддерживает следующие параметры.

Параметр	Тип	Описание
groupids	ID/array	Возвращать только скрипты, которые можно запускать для указанных групп узлов сети.
hostids	ID/array	Возвращать только скрипты, которые можно запускать для указанных узлов сети.
scriptids	ID/array	Возвращать только скрипты с указанными ID.
usrgrpids	ID/array	Возвращать только скрипты, которые могут запускаться пользователями из указанных групп пользователей.
selectHostGroups	query	Возвращать свойство `hostgroups` с группами узлов сети, для которых можно запускать скрипт.
selectHosts	query	Возвращать свойство `hosts` с узлами сети, для 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

Получение всех скриптов

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(          )`

Этот метод позволяет получить все доступные скрипты для данного события или конкретный скрипт, если указан ID скрипта. Если предоставлено значение manualinput, оно заменяет макрос {MANUALINPUT} на указанное значение.

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)` Метод принимает объект или массив объектов со следующими параметрами.

Параметр	Тип	Описание
eventid	ID	ID события, для которого требуется вернуть скрипты. Должен быть уникальным. Поведение параметра: - *обязательный*
scriptid	ID	ID скрипта, который требуется вернуть.
manualinput	string	Значение предоставленного пользователем значения макроса {MANUALINPUT}.

Возвращаемые значения

`(object)` Возвращает объект, в котором идентификаторы событий используются в качестве свойств, а массивы доступных скриптов — в качестве значений. Если указан идентификатор скрипта, соответствующее значение представляет собой массив, содержащий конкретный скрипт.

Note:
Метод автоматически развернёт макросы в тексте `confirmation`, тексте `manualinput prompt` и `url`.
Если указан параметр manualinput, макрос {MANUALINPUT} будет преобразован 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": "                    ",
                "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": "   ",
                "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": "          Zabbix",
            "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": "           ",
            "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": "                    ",
            "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": "   ",
            "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": "          Zabbix",
            "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": "           ",
            "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.

Получение скрипта 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": "Are you sure you want to open page dashboard.view?",
          "type": "6",
          "execute_on": "2",
          "timeout": "30s",
          "scope": "4",
          "port": "",
          "authtype": "0",
          "username": "",
          "password": "",
          "publickey": "",
          "privatekey": "",
          "menu_path": "",
          "url": "http://localhost/ui/zabbix.php?action=dashboard.view",
          "new_window": "1",
          "manualinput": "1",
          "manualinput_prompt": "Zabbix page to open:",
          "manualinput_validator_type": "1",
          "manualinput_validator": "dashboard.view,discovery.view",
          "manualinput_default_value": "",
          "parameters": []
        }
      ]
```

```
    },
    "id": 1
}
```

Исходный код

CScript::getScriptsByEvents() в *ui/include/classes/api/services/CScript.php*.

script.getscriptsbyhosts

Описание

`object script.getscriptsbyhosts(  )`

Этот метод позволяет получить все доступные скрипты для данного хоста или конкретный скрипт, если указан ID скрипта. Если предоставлено значение manualinput, оно заменяет макрос {MANUALINPUT} на указанное значение.

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)` Метод принимает объект или массив объектов со следующими параметрами.

Параметр	Тип	Описание
hostid	ID	ID узла сети, для которого нужно вернуть скрипты. Должен быть уникальным. Поведение параметра: - *обязательный*
scriptid	ID	ID скрипта, который нужно вернуть.
manualinput	string	Значение предоставленного пользователем значения макроса {MANUALINPUT}.

Возвращаемые значения

`(object)` Возвращает объект, в котором идентификаторы узлов сети используются в качестве свойств, а массивы доступных скриптов — в качестве значений. Если указан идентификатор скрипта, соответствующее значение представляет собой массив, содержащий конкретный скрипт.

Note:
Метод автоматически развернёт макросы в тексте `confirmation`, тексте `manualinput prompt` и `url`.
Если указан параметр manualinput, макрос {MANUALINPUT} будет преобразован 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": "                    ",
                "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": "          Zabbix",
            "command": "",
            "host_access": "2",
            "usrgrpid": "0",
            "groupid": "0",
            "description": "",
            "confirmation": "      ,                  *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        :",
            "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": "                    ",
                "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": "          Zabbix",
                "command": "",
                "host_access": "2",
                "usrgrpid": "0",
                "groupid": "0",
                "description": "",
                "confirmation": "      ,                  *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        :",
                "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.

Получение скрипта 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": "Are you sure you want to open page dashboard.view?",
          "type": "6",
          "execute_on": "2",
          "timeout": "30s",
          "scope": "2",
          "port": "",
          "authtype": "0",
          "username": "",
          "password": "",
          "publickey": "",
          "privatekey": "",
          "menu_path": "",
          "url": "http://localhost/ui/zabbix.php?action=dashboard.view",
          "new_window": "1",
          "manualinput": "1",
          "manualinput_prompt": "Zabbix page to open:",
          "manualinput_validator_type": "1",
          "manualinput_validator": "dashboard.view,discovery.view",
          "manualinput_default_value": "",
          "parameters": []
        }
      ]
    },
    "id": 1
}
```

Источник

CScript::getScriptsByHosts() в *ui/include/classes/api/services/CScript.php*.

script.update

Описание

`object script.update(   /    scripts)`

Этот метод позволяет обновлять существующие скрипты.

> **Note:**
> Этот метод доступен только пользователю с типом доступа *Супер-администратор*. Разрешения на вызов метода можно отозвать в настройках ролей пользователя. Дополнительную информацию см. в разделе Роли пользователей.

Параметры

`(object/array)` Свойства скрипта, которые необходимо обновить.

Свойство `scriptid` должно быть определено для каждого скрипта, все остальные свойства являются необязательными. Будут обновлены только переданные свойства, все остальные останутся без изменений. Исключением является изменение свойства `type` с 5 (Вебхук) на другое значение: свойство `parameters` будет очищено.

Возвращаемые значения

`(object)` Возвращает объект, содержащий идентификаторы обновлённых 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".

Запрос:

```
{
    "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
}
```

Ответ:

```
{
    "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:

- Event
- Event tag
- Media type URL

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.

Свойство	Тип	Описание
eventid	ID	ID события.
source	integer	Тип события. Возможные значения: 0 - событие, созданное триггером; 1 - событие, созданное правилом обнаружения; 2 - событие, созданное активной авторегистрацией агента; 3 - внутреннее событие; 4 - событие, созданное при обновлении состояния сервиса.

Свойство	Тип	Описание
object	integer	Тип объекта, связанного с событием. Возможные значения, если `source` установлено в ”событие, созданное триггером”: 0 - триггер. Возможные значения, если `source` установлено в ”событие, созданное правилом обнаружения”: 1 - обнаруженный узел сети; 2 - обнаруженный сервис. Возможные значения, если `source` установлено в ”событие, созданное активной авторегистрацией агента”: 3 - автоматически зарегистрированный узел сети. Возможные значения, если `source` установлено в ”внутреннее событие”: 0 - триггер; 4 - элемент данных; 5 - правило LLD. Возможные значения, если `source` установлено в ”событие, созданное при обновлении состояния сервиса”: 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	Преобразованное имя события.
value	integer	Состояние связанного объекта. Возможные значения, если `source` установлено в ”событие, созданное триггером” или ”событие, созданное при обновлении состояния сервиса”: 0 - OK; 1 - проблема. Возможные значения, если `source` установлено в ”событие, созданное правилом обнаружения”: 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` установлено в ”внутреннее событие”: 0 - состояние ”normal”; 1 - состояние ”unknown” или ”not supported”. Поведение свойства: - *поддерживается*, если `source` установлено в ”событие, созданное триггером”, ”событие, созданное правилом обнаружения”, ”внутреннее событие” или ”событие, созданное при обновлении состояния сервиса”

Свойство	Тип	Описание
severity	integer	Текущая важность события. Возможные значения: 0 - не классифицировано; 1 - информация; 2 - предупреждение; 3 - средняя; 4 - высокая; 5 - катастрофа.
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 определён только когда событие закрыто правилом глобальной корреляции.
cause_eventid	ID	ID причинного события.
correlationid	ID	ID правила корреляции, которое сгенерировало закрытие проблемы. Этот параметр определён только 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 в нормальном состоянии; 1 - событие подавлено.
opdata	string	Операционные данные с раскрытыми макросами.
urls	array	Активные URL-адреса типа медиа.

Тег события

Объект тега события имеет следующие свойства.

Свойство	Type	Описание
tag	string	Имя тега события.
value	string	Значение тега события.

URL типа медиа

Объект URL типа медиа имеет следующие свойства.

Свойство	Тип	Описание
name	string	Имя URL, определённого для типа медиа.
url	string	Значение URL, определённого для типа медиа.

Результаты будут содержать записи только для активных типов медиа с включённой записью меню события. Макросы, используемые в свойствах, будут раскрыты, но если одно из свойств содержит нераскрытый макрос, оба свойства будут исключены из результатов. Поддерживаемые макросы перечислены 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 устранено, это действие будет пропущено.

- Подтвердить событие. Если событие уже подтверждено, это действие будет пропущено.
- Снять подтверждение события. Если событие не подтверждено, это действие будет пропущено.
- Добавить сообщение.
- Изменить важность события. Если у события уже такая же важность, это действие будет пропущено.
- Подавить событие. Если событие уже подавлено, это действие будет пропущено.
- Снять подавление события. Если событие не подавлено, это действие будет пропущено.
- Изменить ранг события на причину. Если событие уже имеет ранг причины, это действие будет пропущено.
- Изменить ранг события на симптом. Если событие уже имеет ранг симптома, это действие будет пропущено.

Attention:
Обновлять можно только события триггеров.
Обновлять можно только события проблем.
Для закрытия события или изменения важности события необходимы права чтения/записи для триггера.
Чтобы закрыть событие, в триггере должно быть разрешено ручное закрытие.

Note:
Этот метод доступен пользователям любого типа. Права на вызов метода могут быть отозваны в настройках роли пользователя. Подробнее см. в разделе Роли пользователей.

Параметры

`(object/array)` Параметры, содержащие идентификаторы событий и операции обновления, которые должны быть выполнены.

Параметр	Type	Описание
eventids	ID/array	Идентификаторы событий, которые необходимо подтвердить. Поведение параметра: - *обязательный*
action	integer	Действие(я) обновления 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 - снять подавление события; 128 - изменить ранг события на причину; 256 - изменить ранг события на симптом. Это поле битовой маски; допустима любая сумма возможных значений битовой маски (например, 34 для подтверждения и подавления события). Поведение параметра: - *обязательный*
cause_eventid	ID	ID события-причины. Поведение параметра: - *обязательный*, если `action` содержит бит "change event rank to symptom"
message	string	Текст сообщения. Поведение параметра: - *обязательный*, если `action` содержит бит "add message"

Параметр	Type	Описание
severity	integer	Новая важность для событий. Возможные значения: 0 - не классифицировано; 1 - информация; 2 - предупреждение; 3 - средняя; 4 - высокая; 5 - катастрофа. Поведение параметра: - *обязательный*, если `action` содержит бит "change severity"
suppress_until	integer	Unix timestamp, до которого событие должно быть подавлено. Если установлено значение "0", подавление будет бессрочным. Поведение параметра: - *обязательный*, если `action` содержит бит "suppress event"

Возвращаемые значения

`(object)` Возвращает объект, который содержит ID обновленных событий, указанных 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 важности события

Измените важность 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
}
```

Источник

CEvent::acknowledge() в *ui/include/classes/api/services/CEvent.php*.

event.get

Описание

`integer/array event.get(object parameters)`

Этот метод позволяет получать события в соответствии с заданными параметрами.

Attention:
Этот метод может возвращать события удалённой сущности, если эти события ещё не были удалены сборщиком устаревших данных.

Note:
Этот метод доступен пользователям любого типа. Права на вызов этого метода могут быть отозваны в настройках роли пользователя. Подробнее см. в разделе Роли пользователей.

Параметры

`(object)` Параметры, определяющие желаемый вывод.

Метод поддерживает следующие параметры.

Параметр	Тип	Описание
eventids	ID/array	Возвращать только события с указанными ID.
groupids	ID/array	Возвращать только события, созданные объектами, принадлежащими указанным группам узлов сети.
hostids	ID/array	Возвращать только события, созданные объектами, принадлежащими указанным узлам сети.
objectids	ID/array	Возвращать только события, созданные указанными объектами.
source	integer	Возвращать только события указанного типа. Список поддерживаемых типов событий см. на странице объекта event. По умолчанию: 0 - события триггеров.

Параметр	Тип	Описание
object	integer	Возвращать только события, созданные объектами указанного типа. Список поддерживаемых типов объектов см. на странице объекта event. По умолчанию: 0 - триггер.
acknowledged	boolean	Если установлено значение `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 установлено значение `true`, возвращать только подавленные события.
symptom	boolean	Если установлено значение `true`, возвращать только события-симптомы.
severities	integer/array	Возвращать только события с указанными уровнями важности событий. Применяется только если object — триггер.
trigger_severities	integer/array	Возвращать только события с указанными уровнями важности триггеров. Применяется только если object — триггер.
evaltype	integer	Метод вычисления тегов. Возможные значения: 0 - *(по умолчанию)* И/Или; 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, большими или равными указанному ID.
eventid_till	string	Возвращать только события с ID, меньшими или равными 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, находившиеся в состоянии проблемы начиная с `problem_time_from` независимо от их текущего состояния. Применяется только если source — событие триггера, а object — триггер. Этот параметр обязателен, если указан `problem_time_till`.
problem_time_till	timestamp	Возвращает только события, находившиеся в состоянии проблемы до `problem_time_till` независимо от их текущего состояния. Применяется только если source — событие триггера, а object — триггер. Этот параметр обязателен, если указан `problem_time_from`.
value	integer/array	Возвращать только события с указанными значениями.

Параметр	Тип	Описание
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, обновившего событие; `clock` - (`timestamp`) время обновления события; `message` - (`string`) текст сообщения; `action` - (`integer`) выполненное действие обновления, см. `event.acknowledge`; `old_severity` - (`integer`) важность события до этого действия обновления; `new_severity` - (`integer`) важность события после этого действия обновления; `suppress_until` - (`timestamp`) время, до которого событие будет подавлено; `taskid` - (`ID`) ID задачи, если для текущего события выполняется изменение ранга; `username` - (`string`) имя пользователя, обновившего событие; `name` - (`string`) имя пользователя, обновившего событие; `surname` - (`string`) фамилия пользователя, обновившего 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, создавший событие. Поддерживается только для событий, сгенерированных триггерами, элементами данных или правилами LLD.
selectRelatedObject	query	Возвращать свойство `relatedObject` с объектом, создавшим 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, подавившего событие; `suppress_until` - (`integer`) время, до которого событие подавлено.
selectTags	query	Возвращать свойство `tags` с тегами события.
filter	object	Возвращать только те результаты, которые точно соответствуют указанному фильтру. Принимает объект, где ключи — это имена свойств, а значения — либо одиночное значение, либо массив значений для сопоставления. Не поддерживает свойства типа данных `text` data type.
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`.
groupBy	string/array	Группировать результаты по указанным свойствам. Указанные свойства будут возвращены в результатах. Возможные значения: `objectid`.

Параметр	Тип	Описание
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`, но не был использован параметр `groupBy`
- массив объектов с результатами агрегации, если был использован параметр `groupBy`

Примеры

Получение событий триггера

Получите последние события 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        5   )",
            "severity": "0",
            "r_eventid": "0",
            "c_eventid": "0",
            "correlationid": "0",
            "userid": "0",
            "cause_eventid": "0",
```

```
        "acknowledges": [],
        "opdata": "               (1m 5m 15m): (1.6328 3.0522 2.2515),   -  CPU: 2",
        "suppression_data": [],
        "suppressed": "0",
        "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       5   )",
        "severity": "3",
        "r_eventid": "20",
        "c_eventid": "0",
        "correlationid": "0",
        "userid": "0",
        "cause_eventid": "0",
        "acknowledges": [
            {
                "acknowledgeid": "1",
                "userid": "1",
                "clock": "1728657938",
                "message": "        .       ,                          .",
                "action": "38",
                "old_severity": "0",
                "new_severity": "0",
                "suppress_until": "1728744338",
                "taskid": "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"
                }
            ],
            "suppressed": "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 в период 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, подтверждённых указанным пользователем

Получение событий, подтверждённых 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",
            "cause_eventid": "0",
            "acknowledges": [
                {
                    "userid": "10",
                    "action": "2"
                }
            ],
            "opdata": "",
            "suppressed": "0",
            "urls": []
        }
    ],
    "id": 1
}
```

Получение топа триггеров по количеству событий проблем

Получите 5 триггеров с уровнями важности "Предупреждение", "Средняя", "Высокая" или "Катастрофа" вместе с количеством событий проблем за указанный период времени.

Запрос:

```
{
    "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
}
```

Ответ:

```
{
    "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

- Оповещение
- Элемент данных
- Узел сети
- Правило LLD
- Сервис
- Триггер

Источник

CEvent::get() в *ui/include/classes/api/services/CEvent.php*.

Тип оповещения

Этот класс предназначен для работы с типами оповещений.

Ссылки на объекты:

- Тип оповещения
  - Параметры вебхука
  - Параметры скрипта
- Шаблон сообщения

Доступные методы:

- mediatype.create - создать новые типы оповещений
- mediatype.delete - удалить типы оповещений

- mediatype.get - получить типы оповещений
- mediatype.update - обновить типы оповещений

Объект способа оповещения

Следующие объекты непосредственно связаны с API `mediatype`.

Способ оповещения

Объект способа оповещения имеет следующие свойства.

Свойство	Тип	Описание
mediatypeid	ID	ID способа оповещения. Поведение свойства: - *только для чтения* - *обязательно* для операций обновления
name	string	Имя способа оповещения. Поведение свойства: - *обязательно* для операций создания
type	integer	Транспорт, используемый способом оповещения. Возможные значения: 0 - Email; 1 - Скрипт; 2 - SMS; 4 - Вебхук.
exec_path	string	Имя файла скрипта (например, notification.sh), расположенного в каталоге, указанном в параметре конфигурации сервера AlertScriptsPath. Поведение свойства: - *обязательно*, если `type` установлен в значение ”Скрипт”
gsm_modem	string	Имя последовательного устройства GSM-модема. Поведение свойства: - *обязательно*, если `type` установлен в значение ”SMS”
passwd	string	Пароль аутентификации. Поведение свойства: - *поддерживается*, если `smtp_authentication` установлен в значение ”Обычный пароль”
provider	integer	Провайдер электронной почты. Возможные значения: 0 - *(по умолчанию)* Generic SMTP; 1 - Gmail; 2 - Gmail relay; 3 - Office365; 4 - Office365 relay.
smtp_email	string	Адрес электронной почты, с которого будут отправляться уведомления. Поведение свойства: - *обязательно*, если `type` установлен в значение ”Email”
smtp_helo	string	SMTP HELO. Поведение свойства: - *поддерживается*, если `type` установлен в значение ”Email”

Свойство	Тип	Описание
smtp_server	string	SMTP-сервер. Поведение свойства: - *обязательно*, если `type` установлен в значение ”Email”
smtp_port	integer	Порт SMTP-сервера для подключения. По умолчанию: 25. Поведение свойства: - *поддерживается*, если `type` установлен в значение ”Email”
smtp_security	integer	Уровень безопасности SMTP-соединения. Возможные значения: 0 - *(по умолчанию)* Нет; 1 - STARTTLS; 2 - SSL/TLS. Поведение свойства: - *поддерживается*, если `type` установлен в значение ”Email”
smtp_verify_host	integer	Проверка SSL-узла сети для SMTP. Возможные значения: 0 - *(по умолчанию)* Нет; 1 - Да. Поведение свойства: - *поддерживается*, если `smtp_security` установлен в значение ”STARTTLS” или ”SSL/TLS”
smtp_verify_peer	integer	Проверка SSL-пира для SMTP. Возможные значения: 0 - *(по умолчанию)* Нет; 1 - Да. Поведение свойства: - *поддерживается*, если `smtp_security` установлен в значение ”STARTTLS” или ”SSL/TLS”
smtp_authentication	integer	Используемый метод SMTP-аутентификации. Возможные значения: 0 - *(по умолчанию)* Нет; 1 - Обычный пароль; 2 - OAuth token. Аутентификация OAuth не допускается для провайдера электронной почты *Office365 relay*. Поведение свойства: - *поддерживается*, если `type` установлен в значение ”Email”
redirection_url	string	URL веб-интерфейса Zabbix для обратного перенаправления OAuth-авторизации. По умолчанию: Значение свойства настроек API `url` с частью `zabbix.php?action=oauth.authorize` Поведение свойства: - *обязательно*, если `smtp_authentication` установлен в значение ”OAuth token”

Свойство	Тип	Описание
client_id	string	Идентификатор клиента, зарегистрированный на сервере OAuth-авторизации. Поведение свойства: - *обязательно*, если `smtp_authentication` установлен в значение "OAuth token"
client_secret	string	Секрет клиента, зарегистрированный на сервере OAuth-авторизации. Доступен только пользователю типа Super Admin. Поведение свойства: - *обязательно*, если `smtp_authentication` установлен в значение "OAuth token"
authorization_url	string	OAuth URL с параметрами для получения токенов доступа и обновления. Поведение свойства: - *обязательно*, если `smtp_authentication` установлен в значение "OAuth token"
token_url	string	OAuth URL для обмена токена авторизации на токены доступа и обновления. Этот URL также используется сервером для обновления недействительного токена доступа. Поведение свойства: - *обязательно*, если `smtp_authentication` установлен в значение "OAuth token"
tokens_status	integer	Битовая маска состояния токенов. Возможные значения: 0 - (по умолчанию) Оба токена содержат недействительное значение 1 - Токен доступа содержит действительное значение 2 - Токен обновления содержит действительное значение 3 - Оба токена содержат действительное значение. Поведение свойства: - *поддерживается*, если `smtp_authentication` установлен в значение "OAuth token"
access_token	string	Значение токена доступа OAuth. Поведение свойства: - *обязательно*, если `smtp_authentication` установлен в значение "OAuth token"
access_token_updated	timestamp	Метка времени последнего изменения `access_token`, выполненного сервером при обновлении с помощью `refresh_token` или API при изменении токенов. Поведение свойства: - *поддерживается*, если `smtp_authentication` установлен в значение "OAuth token"
access_expires_in	integer	Время в секундах, по истечении которого `access_token` устареет и потребуется выполнить запрос к `refresh_url`. Устанавливается сервером Zabbix при обновлении `access_token` или API при изменении токенов. Метка времени вычисляется добавлением значения `access_token_updated`. Поведение свойства: - *поддерживается*, если `smtp_authentication` установлен в значение "OAuth token"

Свойство	Тип	Описание
refresh_token	string	Значение токена обновления OAuth. Поведение свойства: - *обязательно*, если `smtp_authentication` установлен в значение ”OAuth token”
status	integer	Включен ли способ оповещения. Возможные значения: 0 - *(по умолчанию)* Включен; 1 - Отключен.
username	string	Имя пользователя. Поведение свойства: - *поддерживается*, если `smtp_authentication` установлен в значение ”Обычный пароль”
maxsessions	integer	Максимальное количество оповещений, которые могут обрабатываться параллельно. Возможные значения, если `type` установлен в значение ”SMS”: 1. Возможные значения, если `type` установлен в значение ”Email”, ”Скрипт” или ”Вебхук”: 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. Допускаются секунды и единица времени с суффиксом. Возможные значения: 0-1h. По умолчанию: 10s.
message_format	integer	Формат сообщения. Возможные значения: 0 - Обычный текст; 1 - *(по умолчанию)* HTML. Поведение свойства: - *поддерживается*, если `type` установлен в значение ”Email”
script	text	Тело скрипта вебхука (JavaScript). Поведение свойства: - *обязательно*, если `type` установлен в значение ”Вебхук”
timeout	string	Тайм-аут скрипта вебхука. Допускаются секунды и единица времени с суффиксом. Возможные значения: 1-60s. По умолчанию: 30s. Поведение свойства: - *поддерживается*, если `type` установлен в значение ”Вебхук”

Свойство	Тип	Описание
process_tags	integer	Обрабатывать значения свойств JSON в ответе скрипта вебхука как теги. Эти теги добавляются к любым существующим тегам проблемы. Возможные значения: 0 - *(по умолчанию)* Игнорировать ответ скрипта вебхука; 1 - Обрабатывать ответ скрипта вебхука как теги. Поведение свойства: - *поддерживается*, если `type` установлен в значение ”Вебхук”
show_event_menu	integer	Включать запись в меню события, которая ссылается на пользовательский URL. Также добавляет свойство `urls` в вывод problem.get и event.get. Возможные значения: 0 - *(по умолчанию)* Не включать запись меню события или свойство `urls`; 1 - Включать запись меню события и свойство `urls`. Поведение свойства: - *поддерживается*, если `type` установлен в значение ”Вебхук”
event_menu_url	string	URL, используемый в записи меню события и в свойстве `urls`, возвращаемом problem.get и event.get. Поведение свойства: - *поддерживается*, если `show_event_menu` установлен в значение ”Включать запись меню события и свойство `urls`”
event_menu_name	string	Имя, используемое для записи меню события и в свойстве `urls`, возвращаемом problem.get и event.get. Поведение свойства: - *поддерживается*, если `show_event_menu` установлен в значение ”Включать запись меню события и свойство `urls`”
parameters	array	Параметры вебхука или скрипта. Поведение свойства: - *поддерживается*, если `type` установлен в значение ”Вебхук” или ”Скрипт”
description	text	Описание способа оповещения.

Параметры вебхука

Параметры вебхука имеют следующие свойства.

Свойство	Тип	Описание
name	string	Имя параметра. Поведение свойства: - *обязательно*
value	string	Значение параметра, поддерживает макросы. Поддерживаемые макросы описаны на странице Поддерживаемые макросы.

Параметры скрипта

Параметры скрипта имеют следующие свойства.

Свойство	Type	Описание
sortorder	integer	Порядок, в котором значения параметров будут передаваться скрипту как аргументы командной строки, начиная с 0 для первого аргумента. Поведение свойства: - *обязательно*
value	string	Значение параметра, поддерживает макросы. Поддерживаемые макросы описаны на странице Поддерживаемые макросы.

Шаблон сообщения

Объект шаблона сообщения определяет шаблон, который будет использоваться как сообщение по умолчанию для операций действия при отправке уведомления. Он имеет следующие свойства.

Свойство	Type	Описание
eventsource	integer	Источник события. Возможные значения: 0 - Триггеры; 1 - Обнаружение; 2 - Авторегистрация; 3 - Внутренние события; 4 - Сервисы. Поведение свойства: - *обязательно*
recovery	integer	Режим операции. Возможные значения: 0 - Операции; 1 - Операции восстановления; 2 - Операции обновления. Поведение свойства: - *обязательно*
subject	string	Тема сообщения.
message	string	Текст сообщения.

mediatype.create

Описание

`object mediatype.create( / mediaTypes)`

Этот метод позволяет создавать новые способы оповещения.

Note:
Этот метод доступен только пользователю с типом доступа *Супер-администратор*. Разрешения на вызов метода можно отозвать в настройках ролей пользователя. Дополнительную информацию см. в разделе Роли пользователей.

Параметры

`(object/array)` Типы оповещений для создания.

В дополнение к стандартным свойствам типа оповещения, метод принимает следующие параметры.

Параметр	Тип	Описание
message_templates	array	Шаблоны сообщений, которые будут созданы для типа оповещения.

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(    mediaTypeIds)`

Этот метод позволяет удалять способы оповещения.

Note:
Этот метод доступен только пользователю с типом доступа *Супер-администратор*. Разрешения на вызов метода можно отозвать в настройках ролей пользователя. Дополнительную информацию см. в разделе Роли пользователей.

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)`

Этот метод позволяет получать типы оповещений 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 вывод.

Метод поддерживает следующие параметры.

Note:
При запросе связанной с пользователями информации о типах медиа пользователи типа *Admin* и *User* могут получать данные только о собственном пользователе. Пример см. в разделе Получение типов медиа как *Admin*.

Параметр	Тип	Описание
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, использующими тип медиа.
selectMessageTemplates	query	Возвращать свойство `message_templates` с массивом сообщений типа медиа. Поведение параметра: - *поддерживается* для пользователей типа *Super admin*
selectUsers	query	Возвращать свойство `users` с пользователями, использующими тип медиа. Ограничения в зависимости от типа пользователя см. в `user.get`.
sortfield	string/array	Сортировать результат по указанным свойствам. Возможные значения: `mediatypeid`.

Параметр	Тип	Описание
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.
output	query	Возможные свойства объекта типа медиа для пользователей типа *Admin* и *User*: `mediatypeid`, `name`, `type`, `status`, `maxattempts`. Свойства объекта типа медиа, которые должны быть возвращены. 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 типа данных `string` и `text` data type. Возможные свойства объекта типа медиа для пользователей типа *Admin* и *User*: `name`, `description`.
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 был использован параметр `countOutput`.

Примеры

Получение способов оповещения

Получить все настроенные способы оповещения. Следующий пример возвращает два способа оповещения:

- способ оповещения по электронной почте;
- способ оповещения по 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(  /  mediaTypes)`

Этот метод позволяет обновлять существующие способы оповещения

> **Note:**
> Этот метод доступен только пользователю с типом доступа *Супер-администратор*. Разрешения на вызов метода можно отозвать в настройках ролей пользователя. Дополнительную информацию см. в разделе Роли пользователей.

Параметры

`(object/array)` Свойства типа оповещения, которые необходимо обновить.

Свойство mediatypeid должно быть определено для каждого типа оповещения, все остальные свойства являются необязательными. Будут обновлены только переданные свойства, все остальные останутся без изменений.

В дополнение к стандартным свойствам типа оповещения, метод принимает следующие параметры.

Параметр	Type	Описание
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 оповещения

Включите тип оповещения, то есть установите его статус в "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*.

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 тренда

Следующие объекты напрямую связаны с API trend.

Note:
Объекты тренда различаются в зависимости от типа информации элемента данных. Они создаются сервером Zabbix и не могут быть изменены через API.

Тренд с плавающей точкой

Объект тренда с плавающей точкой имеет следующие свойства.

Свойство	Тип	Описание
clock	timestamp	Метка времени часа, для которого было вычислено значение. Например, метка времени "04:00:00" означает значения, вычисленные для периода "04:00:00-04:59:59".
itemid	ID	ID связанного элемента данных.
num	integer	Количество значений, доступных за этот 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.

Свойство	Тип	Описание
clock	timestamp	Метка времени часа, для которого было вычислено значение. Например, метка времени "04:00:00" означает значения, вычисленные для периода "04:00:00-04:59:59".
itemid	ID	ID связанного элемента данных.
num	integer	Количество значений, доступных за этот 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)`

Этот метод позволяет получать данные трендов в соответствии с заданными параметрами.

> **Note:**
> Этот метод доступен пользователям любого типа. Права на вызов этого метода могут быть отозваны в настройках роли пользователя. Дополнительную информацию смотрите в разделе Роли пользователей.

Параметры

`(object)` Параметры, определяющие желаемый вывод.

Метод поддерживает следующие параметры.

Параметр	Тип	Описание
itemids	ID/array	Возвращать только тренды с указанными ID элементов данных.
time_from	timestamp	Возвращать только значения, которые были собраны после или в указанное время.
time_till	timestamp	Возвращать только значения, которые были собраны до или в указанное время.
countOutput	boolean	Подсчитать количество полученных объектов.
limit	integer	Ограничить количество полученных объектов.
output	query	Задать свойства объекта Trend, которые должны быть возвращены.

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:

- Триггер
- Тег триггера

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.

Свойство	Тип	Описание
triggerid	ID	ID триггера. Поведение свойства: - *только для чтения* - *обязательно* для операций обновления
description	string	Имя триггера. Поведение свойства: - *обязательно* для операций создания
expression	string	Выражение триггера. Элементы данных, у которых `value_type` установлен в 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	Операционные данные.
comments	string	Дополнительное описание триггера.
error	string	Текст ошибки, если при обновлении состояния триггера возникли какие-либо проблемы. Поведение свойства: - *только для чтения*
flags	integer	Происхождение триггера. Возможные значения: 0 - *(по умолчанию)* обычный триггер; 4 - триггер, преобразованный из прототипа. Поведение свойства: - *только для чтения*
lastchange	timestamp	Время последнего изменения состояния триггера. Поведение свойства: - *только для чтения*
priority	integer	Важность триггера. Возможные значения: 0 - *(по умолчанию)* не классифицирован; 1 - информация; 2 - предупреждение; 3 - средняя; 4 - высокая; 5 - катастрофическая.
state	integer	Состояние выражения триггера. Возможные значения: 0 - *(по умолчанию)* состояние триггера актуально; 1 - текущее состояние триггера неизвестно. Поведение свойства: - *только для чтения*

Свойство	Тип	Описание
status	integer	Включен или отключен триггер. Возможные значения: 0 - *(по умолчанию)* включен; 1 - отключен.
templateid	ID	ID триггера родительского шаблона. Поведение свойства: - *только для чтения*
type	integer	Может ли триггер генерировать несколько событий о проблеме. Возможные значения: 0 - *(по умолчанию)* 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 - *(по умолчанию)* OK; 1 - проблема. Поведение свойства: - *только для чтения*
recovery_mode	integer	Режим генерации события OK. Возможные значения: 0 - *(по умолчанию)* Выражение; 1 - Выражение восстановления; 2 - Нет.
recovery_expression	string	Выражение восстановления триггера. Элементы данных, у которых `value_type` установлен в 5 (binary) или 6 (JSON), не могут использоваться в выражениях восстановления триггера.
correlation_mode	integer	Закрытие событий 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.
uuid	string	Универсальный уникальный идентификатор, используемый для связывания импортированных триггеров с уже существующими. Генерируется автоматически, если не указан. Поведение свойства: - *поддерживается*, если триггер принадлежит шаблону

Тег триггера

Объект тега триггера имеет следующие свойства.

Свойство	Type	Описание
tag	string	Имя тега триггера. Поведение свойства: - *обязательно*
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( / triggers)`

Этот метод позволяет создавать новые триггеры.

Note:
Этот метод доступен только пользователям с типом доступа *Администратор* и *Супер-администратор*. Разрешения на вызов метода можно отозвать в настройках ролей пользователя. Дополнительную информацию см. в разделе Роли пользователей.

Параметры

`(object/array)` Триггеры для создания.

В дополнение к стандартным свойствам триггера метод принимает следующие параметры.

Параметр	Тип	Описание
dependencies	array	Триггеры, от которых зависит триггер. Для 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(    triggerIds)`

Этот метод позволяет удалять триггеры.

Note:
Этот метод доступен только пользователям с типом доступа *Администратор* и *Супер-администратор*. Разрешения на вызов метода можно отозвать в настройках ролей пользователя. Дополнительную информацию см. в разделе Роли пользователей.

Параметры

( ) ID удаляемых триггеров.

Возвращаемые значения

( ) Возвращает объект, который содержит ID удаленных триггеров, указанных в свойстве `triggerids`.

Примеры

Удаление нескольких триггеров

Удалите два триггера.

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)`

Этот метод позволяет получать триггеры в соответствии с заданными параметрами.

Note:
Этот метод доступен пользователям любого типа. Права на вызов метода могут быть отозваны в настройках роли пользователя. Дополнительную информацию смотрите в разделе Роли пользователей.

Параметры

`(object)` Параметры, определяющие желаемый вывод.

Метод поддерживает следующие параметры.

Параметр	Тип	Описание
triggerids	ID/array	Возвращать только триггеры с указанными ID.

Параметр	Тип	Описание
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.
hostids	ID/array	Возвращать только триггеры, принадлежащие указанным узлам сети.
itemids	ID/array	Возвращать только триггеры, содержащие указанные элементы данных.
functions	string/array	Возвращать только триггеры, использующие указанные функции. Список поддерживаемых функций см. на странице supported function.
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, у которых последнее событие не подтверждено.
skipDependent	flag	Пропускать триггеры в состоянии проблемы, которые зависят 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.
min_severity	integer	Возвращать только триггеры с важностью не ниже указанной.
evaltype	integer	Метод вычисления тегов. Возможные значения: 0 - *(по умолчанию)* And/Or; 2 - Or.

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 operator: 0 - *(по умолчанию)* Contains; 1 - Equals; 2 - Does not contain; 3 - Does not equal; 4 - Exists; 5 - Does not exist.
expandComment	flag	Разворачивать макросы в описании триггера.
expandDescription	flag	Разворачивать макросы в имени триггера.
expandExpression	flag	Разворачивать функции и макросы в выражении триггера.
selectHostGroups	query	Возвращать группы узлов сети, к которым принадлежит триггер, в свойстве `hostgroups`.
selectHosts	query	Возвращать узлы сети, к 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)` ID функции; `itemid` - `(ID)` ID элемента данных, используемого в функции; `function` - `(string)` имя функции; `parameter` - `(string)` параметр, передаваемый 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, из которого он был обнаружен. Имеет следующие свойства: `parent_triggerid` - `(ID)` ID прототипа триггера, из которого был создан триггер; `status` - `(int)` статус обнаружения триггера: 0 - *(по умолчанию)* триггер обнаружен, 1 - триггер больше не обнаруживается; `ts_delete` - `(timestamp)` время, когда триггер, который больше не обнаруживается, будет удалён; `ts_disable` - `(timestamp)` время, когда триггер, который больше не обнаруживается, будет отключён; `disable_source` - `(int)` признак того, был ли триггер отключён правилом LLD или вручную: 0 - *(по умолчанию)* отключён автоматически, 1 - отключён правилом LLD.
selectDiscoveryRule	query	Возвращать правило низкоуровневого обнаружения, создавшее триггер, в свойстве `discoveryRule`.
selectLastEvent	query	Возвращать последнее значимое событие триггера в свойстве `lastEvent`.
selectTags	query	Возвращать теги триггера в свойстве `tags`.

Параметр	Тип	Описание
inheritedTags	boolean	Возвращать триггеры, имеющие указанные `tags` также 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.
selectInheritedTags	query	Возвращать свойство `inheritedTags` с тегами, унаследованными от шаблонов, связанных шаблонов и узлов сети, а также от элементов данных, на которые ссылаются выражения триггера или восстановления.
selectTemplateGroups	query	Возвращать группы шаблонов, к которым принадлежит триггер, в свойстве `templategroups`.
filter	object	Возвращать только те результаты, которые точно соответствуют указанному фильтру. Принимает объект, где ключи — это имена свойств, а значения — либо одиночное значение, либо массив значений для сопоставления. Не поддерживает свойства типа данных `text` data type. Поддерживает дополнительные свойства: `host` - техническое имя узла сети, которому принадлежит триггер; `hostid` - ID узла сети, которому принадлежит триггер.
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: `triggerid`, `description`, `status`, `priority`, `lastchange`, `hostname`.
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	
selectTriggerDiscovery	query	Возвращать объект обнаружения триггера в свойстве `triggerDiscovery`. Объекты обнаружения триггера связывают триггер с прототипом триггера, из которого он был создан. Этот запрос **устарел**, используйте вместо него `selectDiscoveryData`.

Возвращаемые значения

`(integer/array)` Возвращает одно из следующего:

- массив объектов;
- количество полученных объектов, если использовался параметр `countOutput`.

Примеры

Получение данных по ID триггера

Получить 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} has been restarted (uptime < 10m)",
            "url": "",
            "status": "0",
            "value": "0",
            "priority": "2",
            "lastchange": "0",
            "comments": "The host uptime is less than 10 minutes",
            "error": "",
            "templateid": "10016",
            "type": "0",
            "state": "0",
            "flags": "0",
            "recovery_mode": "0",
            "recovery_expression": "",
            "correlation_mode": "0",
            "correlation_tag": "",
            "manual_close": "0",
            "opdata": "",
            "event_name": "",
            "uuid": "",
            "url_name": "",
            "functions": [
                {
                    "functionid": "13513",
                    "itemid": "24350",
                    "triggerid": "14062",
                    "parameter": "$",
                    "function": "last"
                }
            ]
        }
    ],
    "id": 1
}
```

Получение триггеров в состоянии проблемы

Получите ID, имя и важность всех триггеров в состоянии проблемы и отсортируйте их по важности в порядке убывания.

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             100%",
            "priority": "4"
        },
        {
            "triggerid": "13824",
            "description": "            Zabbix             75%",
            "priority": "3"
        }
    ],
    "id": 1
}
```

Получение определённого триггера с тегами

Получение определённого 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(  /  triggers)`

Этот метод позволяет обновлять существующие триггеры.

Note:
Этот метод доступен только пользователям с типом доступа *Администратор* и *Супер-администратор*. Разрешения на вызов метода можно отозвать в настройках ролей пользователя. Дополнительную информацию см. в разделе Роли пользователей.

Параметры

`(object/array)` Свойства триггера, которые необходимо обновить.

Свойство `triggerid` должно быть определено для каждого триггера, все остальные свойства являются необязательными. Будут обновлены только переданные свойства, все остальные останутся без изменений.

В дополнение к стандартным свойствам триггера метод принимает следующие параметры.

Параметр	Тип	Описание
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 обновлённых триггеров в свойстве `triggerids`.

Примеры

Включение триггера

Включите триггер, то есть установите его статус в ”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

Заменяет 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 являются частью кластера высокой доступности или автономного экземпляра сервера.

Ссылки на объекты:

- Узел высокой доступности

Доступные методы:

- hanode.get - получить узлы

Объект узла высокой доступности

Следующий объект относится к работе кластера высокой доступности серверов Zabbix.

Узел сети высокой доступности

> **Note:**
> Узлы сети создаются сервером Zabbix и не могут быть изменены через API.

Объект узла сети высокой доступности имеет следующие свойства.

Свойство	Тип	Описание
ha_nodeid	ID	ID узла сети.
name	string	Имя, назначенное узлу сети с использованием параметра конфигурации HANodeName в zabbix_server.conf. Пустое для сервера, работающего в автономном режиме.
address	string	IP-адрес или DNS-имя, откуда подключается узел сети.
port	integer	Порт, на котором работает узел сети.
lastaccess	integer	Время heartbeat, то есть время последнего обновления от узла сети. Метка времени UTC.

Свойство	Тип	Описание
status	integer	Состояние узла сети. Возможные значения: 0 - резервный; 1 - остановлен вручную; 2 - недоступен; 3 - активный.

hanode.get

Описание

`integer/array hanode.get( )`

Метод позволяет получить список узлов кластера высокой доступности по заданным параметрам.

> **Note:**
> Этот метод доступен только пользователю с типом доступа *Супер-администратор*. Разрешения на вызов метода можно отозвать в настройках ролей пользователя. Дополнительную информацию см. в разделе Роли пользователей.

Параметры

`(object)` Параметры, определяющие желаемый вывод.

Метод поддерживает следующие параметры.

Параметр	Тип	Описание
ha_nodeids	ID/array	Возвращать только узлы с указанными ID узлов.
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: `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	
output	query	
preservekeys	boolean	
sortorder	string/array	

Возвращаемые значения

`( / )` Возвращает либо:

- массив объектов;
- количество извлеченных объектов, если использовался параметр `countOutput`.

Примеры

Получение списка узлов, упорядоченных по состоянию

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
}
```

Получение списка определённых 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
}
```

Получение списка остановленных узлов

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
}
```

Получение количества резервных узлов

Запрос:

```
{
    "jsonrpc": "2.0",
    "method": "hanode.get",
```

```
    "params": {
        "countOutput": true,
        "filter": {
            "status": 0
        }
    },
    "id": 1
}
```

Ответ:

```
{
    "jsonrpc": "2.0",
    "result": "3",
    "id": 1
}
```

Проверка состояния узлов по определённым 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:

- Узел сети
- Инвентарные данные узла сети
- Тег узла сети

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 узла сети.

Узел сети

Объект узла сети имеет следующие свойства.

Свойство	Type	Описание
hostid	ID	ID 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	Происхождение узла сети. Возможные значения: 0 - обычный узел сети; 4 - узел сети, преобразованный из прототипа. Поведение свойства: - *только для чтения*
inventory_mode	integer	Режим заполнения инвентарных данных узла сети. Возможные значения: -1 - *(по умолчанию)* отключено; 0 - вручную; 1 - автоматически.
ipmi_authtype	integer	Алгоритм аутентификации IPMI. Возможные значения: -1 - *(по умолчанию)* по умолчанию; 0 - отсутствует; 1 - MD2; 2 - MD5 4 - straight; 5 - OEM; 6 - RMCP+.
ipmi_password	string	Пароль IPMI.
ipmi_privilege	integer	Уровень привилегий IPMI. Возможные значения: 1 - callback; 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*

Свойство	Type	Описание
maintenance_status	integer	Статус действующего обслуживания. Возможные значения: 0 - *(по умолчанию)* обслуживание отсутствует; 1 - обслуживание активно. 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	ID обслуживания, которое в данный момент действует для узла 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.
proxyid	ID	ID прокси, который используется для мониторинга узла сети. Поведение свойства: - *обязательно*, если `monitored_by` установлено в ”Proxy”
proxy_groupid	ID	ID группы прокси, которая используется для мониторинга узла сети. Поведение свойства: - *обязательно*, если `monitored_by` установлено в ”Proxy group”
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 битовой маски; допустима любая сумма возможных значений битовой маски (например, 6 для PSK и сертификата).
tls_issuer	string	Издатель сертификата.
tls_subject	string	Субъект сертификата.

Свойство	Type	Описание
tls_psk_identity	string	Идентификатор PSK; должен быть связан только с одним PSK (среди autoregistration, hosts и proxies). Не включайте конфиденциальную информацию в идентификатор PSK, так как он передается по сети в незашифрованном виде, чтобы сообщить получателю, какой PSK использовать. Поведение свойства: - *только для записи* - *обязательно*, если `tls_connect` установлено в "PSK" или `tls_accept` содержит бит "PSK"
tls_psk	string	Предварительно общий ключ (PSK); должен содержать не менее 32 шестнадцатеричных цифр. Поведение свойства: - *только для записи* - *обязательно*, если `tls_connect` установлено в "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*
assigned_proxyid	ID	ID прокси, назначенного сервером Zabbix, если узел сети мониторится группой прокси. Поведение свойства: - *только для чтения*

Инвентарные данные узла сети

Объект инвентарных данных узла сети имеет следующие свойства.

Note:
Каждое свойство имеет собственный уникальный идентификационный номер, который используется для связывания полей инвентарных данных узла сети с элементами данных.

ID	Свойство	Type	Описание	Максимальная длина
4	alias	string	Псевдоним.	128 символов
11	asset_tag	string	Инвентарный номер.	64 символа
28	chassis	string	Шасси.	64 символа
23	contact	string	Контактное лицо.	65535 символов для SQL-баз данных
32	contract_number	string	Номер контракта.	64 символа
47	date_hw_decomm	string	Дата вывода оборудования из эксплуатации.	64 символа
46	date_hw_expiry	string	Дата окончания обслуживания оборудования.	64 символа
45	date_hw_install	string	Дата установки оборудования.	64 символа
44	date_hw_purchase	string	Дата покупки оборудования.	64 символа
34	deployment_status	string	Статус развертывания.	64 символа
14	hardware	string	Оборудование.	255 символов
15	hardware_full	string	Подробные сведения об оборудовании.	65535 символов для SQL-баз данных
39	host_netmask	string	Маска подсети узла сети.	39 символов
38	host_networks	string	Сети узла сети.	65535 символов для SQL-баз данных
40	host_router	string	Маршрутизатор узла сети.	39 символов

ID	Свойство	Type	Описание	Максимальная длина
30	hw_arch	string	Архитектура оборудования.	32 символа
33	installer_name	string	Имя установщика.	64 символа
24	location	string	Местоположение.	65535 символов для SQL-баз данных
25	location_lat	string	Широта местоположения.	16 символов
26	location_lon	string	Долгота местоположения.	16 символов
12	macaddress_a	string	MAC-адрес A.	64 символа
13	macaddress_b	string	MAC-адрес B.	64 символа
29	model	string	Модель.	64 символа
3	name	string	Имя.	128 символов
27	notes	string	Примечания.	65535 символов для SQL-баз данных
41	oob_ip	string	IP-адрес OOB.	39 символов
42	oob_netmask	string	Маска подсети OOB узла сети.	39 символов
43	oob_router	string	Маршрутизатор OOB.	39 символов
5	os	string	Имя ОС.	128 символов
6	os_full	string	Полное имя ОС.	255 символов
7	os_short	string	Краткое имя ОС.	128 символов
61	poc_1_cell	string	Мобильный номер основного контактного лица.	64 символа
58	poc_1_email	string	Основной email.	128 символов
57	poc_1_name	string	Имя основного контактного лица.	128 символов
63	poc_1_notes	string	Примечания по основному контактному лицу.	65535 символов для SQL-баз данных
59	poc_1_phone_a	string	Телефон A основного контактного лица.	64 символа
60	poc_1_phone_b	string	Телефон B основного контактного лица.	64 символа
62	poc_1_screen	string	Экранное имя основного контактного лица.	64 символа
68	poc_2_cell	string	Мобильный номер дополнительного контактного лица.	64 символа
65	poc_2_email	string	Дополнительный email.	128 символов
64	poc_2_name	string	Имя дополнительного контактного лица.	128 символов
70	poc_2_notes	string	Примечания по дополнительному контактному лицу.	65535 символов для SQL-баз данных
66	poc_2_phone_a	string	Телефон A дополнительного контактного лица.	64 символа
67	poc_2_phone_b	string	Телефон B дополнительного контактного лица.	64 символа
69	poc_2_screen	string	Экранное имя дополнительного контактного лица.	64 символа
8	serialno_a	string	Серийный номер A.	64 символа
9	serialno_b	string	Серийный номер B.	64 символа
48	site_address_a	string	Адрес площадки A.	128 символов
49	site_address_b	string	Адрес площадки B.	128 символов
50	site_address_c	string	Адрес площадки C.	128 символов
51	site_city	string	Город площадки.	128 символов
53	site_country	string	Страна площадки.	64 символа
56	site_notes	string	Примечания по площадке.	65535 символов для SQL-баз данных
55	site_rack	string	Расположение стойки на площадке.	128 символов
52	site_state	string	Регион/область площадки.	64 символа
54	site_zip	string	ZIP/почтовый индекс площадки.	64 символа
16	software	string	Программное обеспечение.	255 символов
18	software_app_a	string	Программное приложение A.	64 символа
19	software_app_b	string	Программное приложение B.	64 символа
20	software_app_c	string	Программное приложение C.	64 символа
21	software_app_d	string	Программное приложение D.	64 символа
22	software_app_e	string	Программное приложение E.	64 символа
17	software_full	string	Сведения о программном обеспечении.	65535 символов для SQL-баз данных
10	tag	string	Тег.	64 символа

ID	Свойство	Type	Описание	Максимальная длина
1	type	string	Тип.	64 символа
2	type_full	string	Сведения о типе.	64 символа
35	url_a	string	URL A.	2048 символов
36	url_b	string	URL B.	2048 символов
37	url_c	string	URL C.	2048 символов
31	vendor	string	Поставщик.	64 символа

Тег узла сети

Объект тега узла сети имеет следующие свойства.

Свойство	Type	Описание
tag	string	Имя тега узла сети. Поведение свойства: - *обязательно*
value	string	Значение тега узла сети. Поведение свойства: - *только для чтения*.
automatic	integer	Тип тега узла сети. Возможные значения: 0 - *(по умолчанию)* вручную (тег создан пользователем); 1 - автоматически (тег создан обнаружением низкого уровня) Поведение свойства: - *только для чтения*
object	integer	Тип объекта, от которого был унаследован тег. Возможные значения: 0 - шаблон. Поведение свойства: - *только для чтения*.
objectid	ID	ID объекта, от которого был унаследован тег. Поведение свойства: - *только для чтения*.

host.create

Описание

`object host.create( / hosts)`

Этот метод позволяет создавать новые хосты.

> **Note:**
> Этот метод доступен только пользователям с типом доступа *Администратор* и *Супер-администратор*. Разрешения на вызов метода можно отозвать в настройках ролей 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.

Параметр	Тип	Описание
groups	object/array	Группы узлов сети, в которые нужно добавить узел сети. Для групп узлов сети должно быть определено только свойство `groupid`. Поведение параметра: - *обязательный*
interfaces	object/array	Интерфейсы, которые будут созданы для узла сети.
tags	object/array	Теги узла сети.
templates	object/array	Шаблоны, которые будут связаны с узлом сети. Для шаблонов должно быть определено только свойство `templateid`.
macros	object/array	Пользовательские макросы, которые будут созданы для узла сети.
inventory	object	Свойства инвентарных данных 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 инвентарных данных узла сети.

Запрос:

```
{
    "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": "          ,          "
            }
        ],
        "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 PSK-шифрованием

Создайте узел сети с именем "PSK host" с настроенным PSK-шифрованием. 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, отслеживаемого прокси

Создайте узел сети, который будет отслеживаться 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, отслеживаемого группой прокси

Создайте узел сети, который будет отслеживаться группой прокси с ID ”1”.

Запрос:

```
{
    "jsonrpc": "2.0",
    "method": "host.create",
    "params": {
        "host": "Host monitored by proxy group",
        "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(    hosts)`

Этот метод позволяет удалять узлы сети.

> **Note:**
> Этот метод доступен только пользователям с типом доступа *Администратор* и *Супер-администратор*. Разрешения на вызов метода можно отозвать в настройках ролей 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(          )`

Метод позволяет получить хосты по заданным параметрам.

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)` Параметры, определяющие желаемый вывод.

Метод поддерживает следующие параметры.

Параметр	Тип	Описание
groupids	ID/array	Возвращать только узлы сети, принадлежащие указанным группам.
dserviceids	ID/array	Возвращать только узлы сети, связанные с указанными обнаруженными сервисами.
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, использующие 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 отслеживаемые узлы сети.
proxyids	ID/array	Возвращать только узлы сети, которые отслеживаются указанными прокси.
proxy_groupids	ID/array	Возвращать только узлы сети, которые отслеживаются указанными группами прокси.
templated_hosts	flag	Возвращать как узлы сети, так 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.
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, включённые 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 включённые веб-проверки.
with_monitored_items	flag	Возвращать только узлы сети, у которых есть включённые элементы данных. Переопределяет параметр `with_simple_graph_items`.
with_monitored_triggers	flag	Возвращать только узлы сети, у которых есть включённые триггеры. Все элементы данных, используемые в триггере, также должны быть включены.

Параметр	Тип	Описание
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 значение `true`, возвращать только узлы сети с подавленными проблемами.
evaltype	integer	Метод вычисления тегов. Возможные значения: 0 - *(по умолчанию)* И/Или; 2 - Или.
severities	integer/array	Возвращать узлы сети, у которых есть только проблемы указанных уровней серьёзности. Применяется только если объект проблемы — триггер.
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 operator: 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 шаблонах. Возможные значения: `true` - связанные шаблоны также должны иметь указанные `tags`; `false` - *(по умолчанию)* теги связанных шаблонов игнорируются.
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. 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 обнаруживается; `ts_delete` - `(timestamp)` время, когда узел сети, который больше не обнаруживается, будет удалён; `ts_disable` - `(timestamp)` время, когда узел сети, который больше не обнаруживается, будет отключён; `disable_source` - `(int)` индикатор того, был ли узел сети отключён правилом LLD или вручную: 0 - *(по умолчанию)* отключён автоматически, 1 - отключён правилом LLD.
selectDiscoveryRule	query	Возвращать свойство `discoveryRule` с правилом обнаружения низкого уровня, которое создало узел сети (из прототипа узла сети при 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`.

Параметр	Тип	Описание
selectHostGroups	query	Возвращать свойство `hostgroups` с данными групп узлов сети, к которым принадлежит узел сети.
selectHttpTests	query	Возвращать свойство `httpTests` с веб-сценариями узла сети. Поддерживает `count`.
selectInterfaces	query	Возвращать свойство `interfaces` с интерфейсами узла сети. Поддерживает `count`.
selectInventory	query	Возвращать свойство `inventory` с инвентарными данными 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, с которыми связан узел сети. Помимо полей объекта Template, содержит `link_type` - `(integer)` способ, которым шаблон связан с узлом сети. Возможные значения: 0 - *(по умолчанию)* связан вручную; 1 - автоматически связан через 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.
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 заданному 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: 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`.

Параметр	Тип	Описание
search	object	Возвращать результаты, соответствующие заданному шаблону (без учёта 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 типов данных `string` и `text` data type. Поддерживает дополнительные свойства: свойства интерфейса узла сети.
searchInventory	object	Возвращать узлы сети, у которых инвентарные данные соответствуют заданному шаблону (без учёта 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 типов данных `string` и `text` data type.
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	
selectDiscoveries	query	Возвращать свойство `discoveries` с правилами обнаружения низкого уровня узла сети. Поддерживает `count`. Этот запрос **устарел**, используйте вместо него `selectDiscoveryRules`.
selectHostDiscovery	query	Возвращать свойство `hostDiscovery` с данными объекта обнаружения узла сети. Этот запрос **устарел**, используйте вместо него `selectDiscoveryData`.

Возвращаемые значения

`(integer/array)` Возвращает либо:

- массив объектов;
- количество извлеченных объектов, если параметр `countOutput` был использован.

Примеры

Получение данных по имени

Получите 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": "The Zabbix monitoring server.",
            "tls_connect": "1",
            "tls_accept": "1",
            "tls_issuer": "",
            "tls_subject": "",
            "proxy_groupid": "0",
            "monitored_by": "0",
            "inventory_mode": "1",
            "active_available": "1",
            "assigned_proxyid": "0"
        },
        {
            "hostid": "10167",
            "proxyid": "0",
            "host": "Linux server",
            "status": "0",
            "ipmi_authtype": "-1",
            "ipmi_privilege": "2",
            "ipmi_username": "",
            "ipmi_password": "",
            "maintenanceid": "0",
            "maintenance_status": "0",
            "maintenance_type": "0",
            "maintenance_from": "0",
            "name": "Linux server",
            "flags": "0",
            "description": "",
            "tls_connect": "1",
            "tls_accept": "1",
            "tls_issuer": "",
            "tls_subject": "",
```

```
            "proxy_groupid": "0",
            "monitored_by": "0",
            "inventory_mode": "1",
            "active_available": "1",
            "assigned_proxyid": "0"
        }
    ],
    "id": 1
}
```

Получение групп узлов сети

Получите группы узлов сети, в которые входит 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 агент*).

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 инвентаризации узла сети

Получение узлов сети, содержащих "Linux" в поле инвентаризации узла сети "OS".

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 есть не только на уровне узла сети, но и в их 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, которые связаны 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 элементов 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(          )`

Этот метод позволяет одновременно добавлять несколько связанных объектов во все заданные хосты.

Note:
Этот метод доступен только пользователям с типом доступа *Администратор* и *Супер-администратор*. Разрешения на вызов метода можно отозвать в настройках ролей пользователя. Дополнительную информацию см. в разделе Роли пользователей.

Параметры

`(object)` Параметры, содержащие идентификаторы узлов сети для обновления и объекты для добавления ко всем узлам сети.

Метод принимает следующие параметры.

Параметр	Тип	Описание
hosts	object/array	Узлы сети, которые необходимо обновить. Для узлов сети должно быть определено только свойство `hostid`. Поведение параметров: - *обязательный*
groups	object/array	Группы узлов сети, которые необходимо добавить к указанным узлам сети. Для групп узлов сети должно быть определено только свойство `groupid`.
interfaces	object/array	Интерфейсы узлов сети, которые будут созданы для указанных узлов сети.
macros	object/array	Пользовательские макросы, которые будут созданы для указанных узлов сети.
templates	object/array	Шаблоны, которые необходимо привязать к указанным узлам сети. Для 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": "           "
            }
        ]
    },
    "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(          )`

Этот метод позволяет удалять связанные объекты с нескольких хостов.

Note:
Этот метод доступен только пользователям с типом доступа *Администратор* и *Супер-администратор*. Разрешения на вызов метода можно отозвать в настройках ролей пользователя. Дополнительную информацию см. в разделе Роли пользователей.

Параметры

`(object)` Параметры, содержащие идентификаторы узлов сети для обновления и объекты, которые должны быть удалены.

Параметр	Тип	Описание
hostids	ID/array	Идентификаторы узлов сети, которые должны быть обновлены. Поведение параметра: - *обязательный*
groupids	ID/array	Идентификаторы групп узлов сети, из которых следует удалить указанные узлы сети.
interfaces	object/array	Интерфейсы узла сети, которые следует удалить с указанных узлов сети. В объекте интерфейса узла сети должны быть определены только свойства `ip`, `dns` и `port`.
macros	string/array	Пользовательские макросы, которые следует удалить с указанных узлов сети.
templateids	ID/array	Идентификаторы шаблонов, которые следует отвязать от указанных узлов сети.
templateids_clear	ID/array	Идентификаторы шаблонов, которые следует отвязать от указанных узлов сети и очистить их.

Возвращаемые значения

`(object)` Возвращает объект, содержащий идентификаторы обновленных хостов. под свойством `hostids`.

Примеры

Отсоединение шаблонов

Отсоедините шаблон от двух узлов сети и удалите все сущности, унаследованные из 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(  /  hosts)`

Этот метод позволяет обновлять существующие узлы сети.

> **Note:**
> Этот метод доступен только пользователям с типом доступа *Администратор* и *Супер-администратор*. Разрешения на вызов метода можно отозвать в настройках ролей пользователя. Дополнительную информацию см. в разделе Роли пользователей.

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 обновление технического имени узла сети также обновит видимое имя узла сети (если оно не указано отдельно) значением технического имени узла сети.

В дополнение к стандартным свойствам узла сети, метод принимает следующие параметры.

Параметр	Тип	Описание
groups	object/array	Группы узлов сети для замены текущих групп узлов сети, к которым принадлежит узел сети. Все группы узлов сети, не перечисленные в запросе, будут отвязаны. Для групп узлов сети должно быть определено только свойство `groupid`.
interfaces	object/array	Интерфейсы узла сети для замены текущих интерфейсов узла сети. Все интерфейсы, не перечисленные в запросе, будут удалены.
tags	object/array	Теги узла сети для замены текущих тегов узла сети. Все теги, не перечисленные в запросе, будут удалены.
inventory	object	Свойства инвентарных данных узла сети.
macros	object/array	Пользовательские макросы для замены текущих пользовательских макросов. Все макросы, не перечисленные в запросе, будут удалены.
templates	object/array	Шаблоны для замены текущих привязанных шаблонов. Все шаблоны, не перечисленные в запросе, будут только отвязаны. Для шаблонов должно быть определено только свойство `templateid`.
templates_clear	object/array	Шаблоны для отвязки и очистки от узла сети. Для шаблонов должно быть определено 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
}
```

Отсоединение шаблонов

Отсоединить и очистить 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": "           "
            }
        ]
    },
    "id": 1
}
```

Ответ:

```
{
    "jsonrpc": "2.0",
    "result": {
        "hostids": [
            "10126"
        ]
    },
    "id": 1
}
```

Обновление инвентарных данных узла сети

Измените режим инвентаризации и добавьте местоположение

Запрос:

```
{
    "jsonrpc": "2.0",
    "method": "host.update",
    "params": {
        "hostid": "10387",
        "inventory_mode": 0,
        "inventory": {
            "location": "    ,    "
        }
    },
    "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 макросов обнаруженного узла сети

Преобразуйте макрос, созданный правилом обнаружения как ”automatic”, в ”manual” и измените его значение 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, а также 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
- Инвентарные данные узла сети
- Тег узла сети
- Прокси
- Группа прокси

Источник

CHost::update() в *ui/include/classes/api/services/CHost.php*.

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 в шаблоны
- 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.

Свойство	Тип	Описание
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	Видимое имя шаблона. По умолчанию: значение свойства `host`.
readme	text	Инструкции по настройке, относящиеся к шаблону, которые отображаются в мастере узла сети. Поддерживается 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 свойства `vendor_name` и `vendor_version` должны быть либо заданы, либо оставлены пустыми. Для операций обновления `vendor_version` можно оставить пустым, если оно имеет значение в базе данных.
vendor_version	string	Версия поставщика шаблона. Для операций создания оба свойства `vendor_name` и `vendor_version` должны быть либо заданы, либо оставлены пустыми. Для операций обновления `vendor_name` можно оставить пустым, если оно имеет значение в базе данных.
wizard_ready	integer	Доступен ли шаблон для выбора в мастере узла сети. Возможные значения: 0 - *(по умолчанию)* Недоступен. 1 - Доступен.

Тег шаблона

Объект тега шаблона имеет следующие свойства.

Свойство	Тип	Описание
tag	string	Имя тега шаблона. Поведение свойства: - *обязательно*
value	string	Значение тега шаблона. Поведение свойства: - *только для чтения*.
object	integer	Тип объекта, от которого был унаследован тег. Возможные значения: 0 - шаблон. Поведение свойства: - *только для чтения*.

Свойство	Тип	Описание
objectid	ID	ID объекта, от которого был унаследован тег. Поведение свойства: - *только для чтения*.

template.create

Описание

`object template.create( / templates)`

Этот метод позволяет создавать новые шаблоны.

Note:
Этот метод доступен только пользователям с типом доступа *Администратор* и *Супер-администратор*. Разрешения на вызов метода можно отозвать в настройках ролей пользователя. Дополнительную информацию см. в разделе Роли пользователей.

Параметры

`(object/array)` Шаблоны для создания.

В дополнение к стандартным свойствам шаблона, метод принимает следующие параметры.

Параметр	Тип	Описание
groups	object/array	Группы шаблонов, в которые следует добавить шаблон. Для групп шаблонов должно быть определено только свойство `groupid`. Поведение параметра: - *обязательный*
tags	object/array	Теги шаблона.
templates	object/array	Шаблоны, которые должны быть связаны с шаблоном. Для шаблонов должно быть определено только свойство `templateid`.
macros	object/array	Пользовательские макросы, которые должны быть созданы 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 привяжите к этому шаблону два шаблона.

Запрос:

```
{
    "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
}
```

Ответ:

```
{
    "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.

Удаление шаблона также удаляет все его сущности (элементы данных, триггеры, графики и т. д.) со всех узлов сети или шаблонов, с которыми он связан. Чтобы сохранить эти сущности, сначала отсоедините 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*. Права на вызов этого метода могут быть отозваны в настройках роли пользователя. Дополнительную информацию смотрите 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

Удалите два шаблона.

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. Права на вызов этого метода могут быть отозваны в настройках роли пользователя. См. Роли пользователей для получения дополнительной информации.

Параметры

`(object)` Параметры, определяющие желаемый вывод.

Метод поддерживает следующие параметры.

Параметр	Тип	Описание
templateids	ID/array	Возвращать только шаблоны с указанными ID шаблонов.
groupids	ID/array	Возвращать только шаблоны, принадлежащие указанным группам шаблонов.
parentTemplateids	ID/array	Возвращать только шаблоны, с которыми связан указанный шаблон.
hostids	ID/array	Возвращать только шаблоны, связанные с указанными узлами сети/шаблонами.
graphids	ID/array	Возвращать только шаблоны, содержащие указанные графики.
itemids	ID/array	Возвращать только шаблоны, содержащие указанные элементы данных.
triggerids	ID/array	Возвращать только шаблоны, содержащие 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 вычисления тегов. Возможные значения: 0 - *(по умолчанию)* И/Или; 2 - Или.

Параметр	Тип	Описание
tags	object/array	Возвращать только шаблоны с указанными тегами. Формат: `[{"tag": "<tag>", "value": "<value>", "operator": "<operator>"}, ...]`. Пустой массив возвращает все шаблоны. Возможные значения operator: 0 - *(по умолчанию)* Содержит; 1 - Равно; 2 - Не содержит; 3 - Не равно; 4 - Существует; 5 - Не существует.
inheritedTags	boolean	Возвращать шаблоны, у которых указанные `tags` также присутствуют 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, находящимися 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, связанные с шаблоном, в свойстве `hosts`. Поддерживает `count`.
selectTemplateGroups	query	Возвращать группы шаблонов, к которым принадлежит шаблон, в свойстве `templategroups`.
selectTemplates	query	Возвращать шаблоны, с которыми связан указанный шаблон, в свойстве `templates`. Поддерживает `count`.
selectParentTemplates	query	Возвращать шаблоны, связанные с указанным шаблоном, 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 панели из шаблона в свойстве `dashboards`. Поддерживает `count`.
selectValueMaps	query	Возвращать свойство `valuemaps` с картами значений шаблона.

Параметр	Тип	Описание
limitSelects	integer	Ограничивает количество записей, возвращаемых подзапросами. Применяется к следующим подзапросам: `selectTemplates` - результаты будут отсортированы по `name`; `selectHosts` - отсортированы по `host`; `selectParentTemplates` - отсортированы по `host`; `selectItems` - отсортированы по `name`; `selectDiscoveryRules` - отсортированы по `name`; `selectTriggers` - отсортированы по `description`; `selectGraphs` - отсортированы по `name`; `selectDashboards` - отсортированы 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 low-level discovery из шаблона в свойстве `discoveries`. Поддерживает `count`. Этот запрос **устарел**, используйте вместо него `selectDiscoveryRules`.

Возвращаемые значения

`(integer/array)` Возвращает одно из следующего:

- массив объектов;
- количество полученных объектов, если был 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

Получите группы шаблонов, в которые входит шаблон "Linux by Zabbix агент".

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 агент*).

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(  )`

Этот метод позволяет одновременно добавить несколько связанных объектов к указанным шаблонам.

Note:
Этот метод доступен только пользователям с типом доступа *Администратор* и *Супер-администратор*. Разрешения на вызов метода можно отозвать в настройках ролей пользователя. Дополнительную информацию см. в разделе Роли пользователей.

Параметры

`(object)` Параметры, содержащие идентификаторы шаблонов для обновления и объекты для добавления в шаблоны.

Метод принимает следующие параметры.

Параметр	Type	Описание
templates	object/array	Шаблоны, которые необходимо обновить. Для шаблонов должно быть определено только свойство `templateid`. Поведение параметра: - *обязательный*
groups	object/array	Группы шаблонов, в которые необходимо добавить указанные шаблоны. Для групп шаблонов должно быть определено только свойство `groupid`.
macros	object/array	Пользовательские макросы, которые необходимо создать для указанных шаблонов.
templates_link	object/array	Шаблоны, которые необходимо связать с указанными шаблонами. Для шаблонов должно быть определено только свойство `templateid`.

Возвращаемые значения

`(  )` Возвращает объект, который содержит ID обновленных шаблонов, указанных в свойстве `templateids`.

Примеры

Привязка группы к шаблонам

Добавьте 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
}
```

Привязка двух шаблонов к шаблону

Привяжите шаблоны ”10106” и ”10104” к шаблону ”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 templategroup.massremove(  )`

Этот метод позволяет удалить связанные объекты из нескольких шаблонов.

Note:
Этот метод доступен только пользователям с типом доступа *Администратор* и *Супер-администратор*. Разрешения на вызов метода можно отозвать в настройках ролей пользователя. Дополнительную информацию см. в разделе Роли пользователей.

Параметры

`(object)` Параметры, содержащие идентификаторы шаблонов для обновления и объекты, которые должны быть удалены.

Параметр	Type	Описание
templateids	ID/array	Идентификаторы шаблонов, которые должны быть обновлены. Поведение параметра: - *обязательный*
groupids	ID/array	Идентификаторы групп шаблонов, из которых следует удалить указанные шаблоны.
macros	string/array	Идентификаторы пользовательских макросов, которые следует удалить из указанных шаблонов.
templateids_clear	ID/array	Идентификаторы шаблонов, которые следует отвязать и очистить от указанных шаблонов (вышестоящих).
templateids_link	ID/array	Идентификаторы шаблонов, которые следует отвязать от указанных шаблонов (вышестоящих).

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
}
```

Отсоединение шаблонов от узла сети

Отсоедините 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( / templates)`

Этот метод позволяет обновлять существующие шаблоны.

Note:
Этот метод доступен только пользователям с типом доступа *Администратор* и *Супер-администратор*. Разрешения на вызов метода можно отозвать в настройках ролей пользователя. Дополнительную информацию см. в разделе Роли пользователей.

Параметры

`(object/array)` Свойства шаблона, которые необходимо обновить.

Свойство templateid должно быть определено для каждого шаблона, все остальные свойства являются необязательными. Будут обновлены только указанные свойства, все остальные останутся без изменений.

В дополнение к стандартным свойствам шаблона, метод принимает следующие параметры.

Параметр	Тип	Описание
groups	object/array	Группы шаблонов, которые заменят текущие группы шаблонов, к которым принадлежат шаблоны. Для групп шаблонов должно быть определено только свойство groupid.
tags	object/array	Теги шаблона, которые заменят текущие теги шаблона.
macros	object/array	Пользовательские макросы, которые заменят текущие пользовательские макросы в указанных шаблонах.
templates	object/array	Шаблоны, которые заменят связанные в данный момент шаблоны. Шаблоны, которые не переданы, будут только отвязаны. Для шаблонов должно быть определено только свойство templateid.
templates_clear	object/array	Шаблоны, которые необходимо отвязать и очистить от указанных шаблонов. Для шаблонов должно быть определено 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-заголовок
  - Поле HTTP-запроса
- 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 данных нельзя напрямую создавать, обновлять или удалять через Zabbix API.

Объект элемента данных имеет следующие свойства.

Свойство	Type	Описание
itemid	ID	ID элемента данных. Поведение свойства: - *только для чтения* - *обязательно* для операций обновления

Свойство	Type	Описание
delay	string	Интервал обновления элемента данных. Принимает секунды или единицу времени с суффиксом (например, 30s, 1m, 2h, 1d) и, при необходимости, один или несколько пользовательских интервалов, разделенных точкой с запятой. Пользовательские интервалы могут представлять собой смесь гибких интервалов и интервалов по расписанию. Поддерживает пользовательские макросы. Если они используются, значение должно быть одним макросом. Несколько макросов или макросы, смешанные с текстом, не поддерживаются. Гибкие интервалы могут быть записаны как два макроса, разделенные 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` установлен в ”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), или если `type` установлен в ”Zabbix agent (active)” (7) и `key_` не содержит ”mqtt.get”
hostid	ID	ID узла сети или шаблона, которому принадлежит элемент данных. Поведение свойства: - *постоянное* - *обязательно* для операций создания
interfaceid	ID	ID интерфейса узла сети для элемента данных. Поведение свойства: - *обязательно*, если элемент данных принадлежит узлу сети и `type` установлен в ”Zabbix agent”, ”IPMI agent”, ”JMX agent”, ”SNMP trap” или ”SNMP agent” - *поддерживается*, если элемент данных принадлежит узлу сети и `type` установлен в ”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 раскрытыми пользовательскими макросами. Поведение свойства: - *только для чтения*

Свойство	Type	Описание
type	integer	Тип элемента данных. Возможные значения: 0 - Zabbix agent; 2 - Zabbix trapper; 3 - Simple check; 5 - Zabbix internal; 7 - Zabbix agent (active); 9 - Веб-элемент данных; 10 - External check; 11 - Database monitor; 12 - IPMI agent; 13 - SSH agent; 14 - TELNET agent; 15 - Calculated; 16 - JMX agent; 17 - SNMP trap; 18 - Зависимый элемент данных; 19 - HTTP agent; 20 - SNMP agent; 21 - Script; 22 - Browser. 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 установлен в ”HTTP agent” - *только для чтения* для унаследованных или обнаруженных объектов
value_type	integer	Тип информации элемента данных. Возможные значения: 0 - числовой с плавающей точкой; 1 - символьный; 2 - журнал; 3 - числовой без знака; 4 - текст; 5 - двоичный; 6 - JSON. Поведение свойства: - *обязательно* для операций создания - *только для чтения* для унаследованных или обнаруженных объектов
allow_traps	integer	Разрешить заполнять значение аналогично элементу данных trapper. 0 - *(по умолчанию)* Не разрешать принимать входящие данные; 1 - Разрешить принимать входящие данные. Поведение свойства: - *поддерживается*, если type установлен в ”HTTP agent” - *только для чтения* для обнаруженных объектов

Свойство	Type	Описание
authtype	integer	Метод аутентификации. Возможные значения, если type установлен в ”SSH agent”: 0 - *(по умолчанию)* пароль; 1 - открытый ключ. Возможные значения, если type установлен в ”HTTP agent”: 0 - *(по умолчанию)* нет; 1 - basic; 2 - NTLM; 3 - Kerberos; 4 - Digest. Поведение свойства: - *поддерживается*, если type установлен в ”SSH agent” или ”HTTP agent” - *только для чтения* для унаследованных объектов (если type установлен в ”HTTP agent”) или обнаруженных объектов
description	string	Описание элемента данных. Поведение свойства: - *только для чтения* для обнаруженных объектов
error	string	Текст ошибки, если возникают проблемы при обновлении значения элемента данных. Поведение свойства: - *только для чтения*
flags	integer	Происхождение 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 установлен в ”HTTP agent” - *только для чтения* для унаследованных или обнаруженных объектов
headers	array	Массив заголовков, которые будут отправлены при выполнении HTTP-запроса. Поведение свойства: - *поддерживается*, если type установлен в ”HTTP agent” - *только для чтения* для унаследованных или обнаруженных объектов
history	string	Единица времени, определяющая, как долго должны храниться исторические данные. Также поддерживает пользовательский макрос. По умолчанию: 31d. Поведение свойства: - *только для чтения* для обнаруженных объектов

Свойство	Type	Описание
http_proxy	string	Строка подключения к HTTP(S) прокси. Поведение свойства: - *поддерживается*, если `type` установлен в "HTTP agent" - *только для чтения* для унаследованных или обнаруженных объектов
inventory_link	integer	ID поля инвентаризации узла сети, которое заполняется этим элементом данных. Список поддерживаемых полей инвентаризации узла сети и их ID см. на странице инвентаризации узла сети. По умолчанию: 0. Поведение свойства: - *поддерживается*, если `value_type` установлен в "numeric float", "character", "numeric unsigned" или "text" - *только для чтения* для обнаруженных объектов
ipmi_sensor	string	Сенсор IPMI. Поведение свойства: - *обязательно*, если `type` установлен в "IPMI agent" и `key_` не установлен в "ipmi.get" - *поддерживается*, если `type` установлен в "IPMI agent" - *только для чтения* для унаследованных или обнаруженных объектов
jmx_endpoint	string	Пользовательская строка подключения JMX agent. Значение по умолчанию: service:jmx:rmi:///jndi/rmi://{HOST.CONN}:{HOST.PORT}/jmxrmi Поведение свойства: - *поддерживается*, если `type` установлен в "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. Вы можете увеличить этот период времени, изменив значение параметра *Max history display period* в разделе меню *Администрирование → Общие*. Поведение свойства: - *только для чтения*
lastns	integer	Наносекунды времени последнего обновления значения элемента данных. По умолчанию отображаются только значения за последние 24 часа. Вы можете увеличить этот период времени, изменив значение параметра *Max history display period* в разделе меню *Администрирование → Общие*. Поведение свойства: - *только для чтения*
lastvalue	string	Последнее значение элемента данных. По умолчанию отображаются только значения за последние 24 часа. Вы можете увеличить этот период времени, изменив значение параметра *Max history display period* в разделе меню *Администрирование → Общие*. Поведение свойства: - *только для чтения*

Свойство	Type	Описание
logtimefmt	string	Формат времени в записях журнала. Поведение свойства: - *поддерживается*, если `value_type` установлен в ”log” - *только для чтения* для унаследованных или обнаруженных объектов
master_itemid	ID	ID мастер-элемента данных. Поведение свойства: - *обязательно*, если `type` установлен в ”Dependent item” - *только для чтения* для унаследованных или обнаруженных объектов
output_format	integer	Следует ли преобразовывать ответ в JSON. 0 - *(по умолчанию)* Хранить как есть; 1 - Преобразовать в JSON. Поведение свойства: - *поддерживается*, если `type` установлен в ”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` установлен в ”Database monitor”, ”SSH agent”, ”TELNET agent”, ”Calculated”, ”Script” или ”Browser” - *только для чтения* для унаследованных объектов (если `type` установлен в ”Script” или ”Browser”) или обнаруженных объектов
parameters	object/array	Дополнительные параметры, если `type` установлен в ”Script” или ”Browser”. Массив объектов со свойствами `name` и `value`, где `name` должно быть уникальным. Поведение свойства: - *поддерживается*, если `type` установлен в ”Script” или ”Browser” - *только для чтения* для унаследованных или обнаруженных объектов
password	string	Пароль для аутентификации. Поведение свойства: - *обязательно*, если `type` установлен в ”JMX agent” и задан `username` - *поддерживается*, если `type` установлен в ”Simple check”, ”SSH agent”, ”TELNET agent”, ”Database monitor” или ”HTTP agent” - *только для чтения* для унаследованных объектов (если `type` установлен в ”HTTP agent”) или обнаруженных объектов
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` установлен в ”HTTP agent” - *только для чтения* для унаследованных или обнаруженных объектов

Свойство	Type	Описание
posts	string	Данные тела HTTP(S)-запроса. Поведение свойства: - *обязательно*, если `type` установлен в ”HTTP agent” и `post_type` установлен в ”JSON data” или ”XML data” - *поддерживается*, если `type` установлен в ”HTTP agent” и `post_type` установлен в ”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. Вы можете увеличить этот период времени, изменив значение параметра *Max history display period* в разделе меню *Администрирование → Общие*. Поведение свойства: - *только для чтения*
privatekey	string	Имя файла закрытого ключа. Поведение свойства: - *обязательно*, если `type` установлен в ”SSH agent” и `authtype` установлен в ”public key” - *только для чтения* для обнаруженных объектов
publickey	string	Имя файла открытого ключа. Поведение свойства: - *обязательно*, если `type` установлен в ”SSH agent” и `authtype` установлен в ”public key” - *только для чтения* для обнаруженных объектов
query_fields	array	Массив полей запроса, которые будут отправлены при выполнении HTTP-запроса. Поведение свойства: - *поддерживается*, если `type` установлен в ”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` установлен в ”HTTP agent” - *только для чтения* для унаследованных или обнаруженных объектов

Свойство	Type	Описание
retrieve_mode	integer	Какая часть ответа должна быть сохранена. Возможные значения, если `request_method` установлен в ”GET”, ”POST” или ”PUT”: 0 - *(по умолчанию)* Тело; 1 - Заголовки; 2 - Будут сохранены и тело, и заголовки. Возможные значения, если `request_method` установлен в ”HEAD”: 1 - Заголовки. Поведение свойства: - *поддерживается*, если `type` установлен в ”HTTP agent” - *только для чтения* для унаследованных или обнаруженных объектов
snmp_oid	string	SNMP OID. Поведение свойства: - *обязательно*, если `type` установлен в ”SNMP agent” - *только для чтения* для унаследованных или обнаруженных объектов
ssl_cert_file	string	Путь к файлу открытого SSL-ключа. Поведение свойства: - *поддерживается*, если `type` установлен в ”HTTP agent” - *только для чтения* для унаследованных или обнаруженных объектов
ssl_key_file	string	Путь к файлу закрытого SSL-ключа. Поведение свойства: - *поддерживается*, если `type` установлен в ”HTTP agent” - *только для чтения* для унаследованных или обнаруженных объектов
ssl_key_password	string	Пароль для файла SSL-ключа. Поведение свойства: - *поддерживается*, если `type` установлен в ”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	Статус элемента данных. Возможные значения: 0 - *(по умолчанию)* элемент данных включен; 1 - элемент данных отключен.

Свойство	Type	Описание
status_codes	string	Диапазоны требуемых кодов состояния HTTP, разделенные запятыми. Также поддерживает пользовательские макросы как часть списка, разделенного запятыми. Пример: 200,200-{$M},{$M},200-400 Поведение свойства: - *поддерживается*, если `type` установлен в ”HTTP agent” - *только для чтения* для унаследованных или обнаруженных объектов
templateid	ID	ID родительского элемента данных шаблона. *Подсказка*: Используйте свойство `hostid`, чтобы указать шаблон, которому принадлежит элемент данных. Поведение свойства: - *только для чтения*
timeout	string	Тайм-аут запроса опроса данных элемента данных. Принимает секунды или единицу времени с суффиксом (например, 30s, 1m). Также поддерживает пользовательские макросы. Допустимый диапазон значений: 1-600s. По умолчанию: ”” - использовать настройки прокси/глобальные настройки. Поведение свойства: - *поддерживается*, если `type` установлен в ”Zabbix agent” (0), ”Simple check” (3) и `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) и `snmp_oid` начинается с ”`walk[`” или ”`get[`”, ”Script” (21), ”Browser” (22) - *только для чтения* для унаследованных и обнаруженных объектов
trapper_hosts	string	Разрешенные узлы сети. Поведение свойства: - *только для чтения* для обнаруженных объектов - *поддерживается*, если `type` установлен в ”Zabbix trapper”, или если `type` установлен в ”HTTP agent” и `allow_traps` установлен в ”Allow to accept incoming data”
trends	string	Единица времени, определяющая, как долго должны храниться данные трендов. Также поддерживает пользовательский макрос. По умолчанию: 365d. Поведение свойства: - *поддерживается*, если `value_type` установлен в ”numeric float” или ”numeric unsigned” - *только для чтения* для обнаруженных объектов
units	string	Единицы измерения значения. Поведение свойства: - *поддерживается*, если `value_type` установлен в ”numeric float” или ”numeric unsigned” - *только для чтения* для унаследованных или обнаруженных объектов

Свойство	Type	Описание
username	string	Имя пользователя для аутентификации. Поведение свойства: - *обязательно*, если `type` установлен в ”SSH agent”, ”TELNET agent”, или если `type` установлен в ”JMX agent” и задан `password` - *поддерживается*, если `type` установлен в ”Simple check”, ”Database monitor” или ”HTTP agent” - *только для чтения* для унаследованных объектов (если `type` установлен в ”HTTP agent”) или обнаруженных объектов
uuid	string	Универсальный уникальный идентификатор, используемый для связывания импортированного элемента данных с уже существующими. Если не указан, генерируется автоматически. Поведение свойства: - *поддерживается*, если элемент данных принадлежит шаблону
valuemapid	ID	ID связанной карты значений. Поведение свойства: - *поддерживается*, если `value_type` установлен в ”numeric float”, ”character” или ”numeric unsigned” - *только для чтения* для унаследованных или обнаруженных объектов
verify_host	integer	Следует ли проверять, что имя узла сети для соединения совпадает с именем в сертификате узла сети. Возможные значения: 0 - *(по умолчанию)* Не проверять; 1 - Проверять. Поведение свойства: - *поддерживается*, если `type` установлен в ”HTTP agent” - *только для чтения* для унаследованных или обнаруженных объектов
verify_peer	integer	Следует ли проверять подлинность сертификата узла сети. Возможные значения: 0 - *(по умолчанию)* Не проверять; 1 - Проверять. Поведение свойства: - *поддерживается*, если `type` установлен в ”HTTP agent” - *только для чтения* для унаследованных или обнаруженных объектов

HTTP-заголовок

Объект заголовка имеет следующие свойства:

Свойство	Type	Описание
name	string	Имя HTTP-заголовка. Поведение свойства: - *обязательно*

Свойство	Type	Описание
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*

Тег элемента данных

Объект тега элемента данных имеет следующие свойства.

Свойство	Type	Описание
tag	string	Имя тега элемента данных. Поведение свойства: - *обязательно*
value	string	Значение тега элемента данных. Поведение свойства: - *только для чтения*.
object	integer	Тип объекта, от которого был унаследован тег. Возможные значения: 0 - шаблон; 1 - узел сети. Поведение свойства: - *только для чтения*.
objectid	ID	ID объекта, от которого был унаследован тег. Поведение свойства: - *только для чтения*.

Предобработка элемента данных

Объект предобработки элемента данных имеет следующие свойства.

Свойство	Тип	Описание
type	integer	Тип варианта предобработки. Возможные значения: 1 - Пользовательский множитель; 2 - Обрезка справа; 3 - Обрезка слева; 4 - Обрезка; 5 - Регулярное выражение; 6 - Логическое значение в десятичное; 7 - Восьмеричное в десятичное; 8 - Шестнадцатеричное 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 ошибки в JSON; 17 - Проверка ошибки в XML; 18 - Проверка ошибки с использованием регулярного выражения; 19 - Отбрасывать без изменений; 20 - Отбрасывать без изменений с контрольным интервалом; 21 - JavaScript; 22 - Шаблон Prometheus; 23 - Prometheus в JSON; 24 - CSV в JSON; 25 - Замена; 26 - Проверка неподдерживаемости; 27 - XML в JSON; 28 - Значение SNMP walk; 29 - SNMP walk в JSON; 30 - Значение SNMP get. Поведение свойства: - *обязательно*
params	string	Дополнительные параметры, используемые вариантом предобработки. Несколько параметров разделяются символом новой строки (\n). Если `type` установлен в ”Проверка неподдерживаемости”, параметры следуют синтаксису `<scope>[\n<pattern>]`, где *pattern* — это регулярное выражение, а *scope* — одно из следующих значений: -1 - соответствие любой ошибке; 0 - проверить, соответствует ли сообщение об ошибке *pattern*; 1 - проверить, не соответствует ли сообщение об ошибке *pattern*. Поведение свойства: - *обязательно*, если `type` установлен в ”Пользовательский множитель” (1), ”Обрезка справа” (2), ”Обрезка слева” (3), ”Обрезка” (4), ”Регулярное выражение” (5), ”XML XPath” (11), ”JSONPath” (12), ”В диапазоне” (13), ”Соответствует регулярному выражению” (14), ”Не соответствует регулярному выражению” (15), ”Проверка ошибки в JSON” (16), ”Проверка ошибки в XML” (17), ”Проверка ошибки с использованием регулярного выражения” (18), ”Отбрасывать без изменений с контрольным интервалом” (20), ”JavaScript” (21), ”Шаблон Prometheus” (22), ”Prometheus в JSON” (23), ”CSV в JSON” (24), ”Замена” (25), Check unsupported (26), ”Значение SNMP walk” (28), ”SNMP walk в JSON” (29) или ”Значение SNMP get” (30)

Свойство	Тип	Описание
error_handler	integer	Тип действия, используемого в случае сбоя шага предобработки. Возможные значения: 0 - Сообщение об ошибке задаётся сервером Zabbix; 1 - Отбросить значение; 2 - Установить пользовательское значение; 3 - Установить пользовательское сообщение об ошибке. Возможные значения, если type установлен в ”Проверка неподдерживаемости”: 1 - Отбросить значение; 2 - Установить пользовательское значение; 3 - Установить пользовательское сообщение об ошибке. Поведение свойства: - *обязательно*, если type установлен в ”Пользовательский множитель” (1), ”Регулярное выражение” (5), ”Логическое значение в десятичное” (6), ”Восьмеричное в десятичное” (7), ”Шестнадцатеричное 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 ошибки в JSON” (16), ”Проверка ошибки в XML” (17), ”Проверка ошибки с использованием регулярного выражения” (18), ”Шаблон Prometheus” (22), ”Prometheus в JSON” (23), ”CSV в JSON” (24), ”Проверка неподдерживаемости” (26), ”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: - *обязательно*, если error_handler установлен в ”Установить пользовательское значение” или ”Установить пользовательское сообщение об ошибке”

Следующие параметры и обработчики ошибок поддерживаются для каждого типа предобработки.

Тип предобработки	Имя	Параметр 1	Параметр 2	Параметр 3	Поддерживаемые обработчики ошибок
1	Пользовательский множитель	значение[1, 6]			0, 1, 2, 3
2	Обрезка справа	список символов[2]			
3	Обрезка слева	список символов[2]			
4	Обрезка	список символов[2]			
5	Регулярное выражение	шаблон[3]	вывод[2]		0, 1, 2, 3
6	Логическое значение в десятичное				0, 1, 2, 3
7	Восьмеричное в десятичное				0, 1, 2, 3
8	Шестнадцатеричное в десятичное				0, 1, 2, 3

Тип предобработки	Имя	Параметр 1	Параметр 2	Параметр 3	Поддерживаемые обработчики ошибок
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	min[1, 6]	max[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
16	Проверка на ошибки в JSON	путь[4]			0, 1, 2, 3
17	Проверка на ошибки в XML	путь[4]			0, 1, 2, 3
18	Проверка на ошибки с использованием регулярного выражения	шаблон[3]	вывод[2]		0, 1, 2, 3
19	Отбрасывать без изменений				
20	Отбрасывать без изменений с контрольным интервалом	секунды[5, 6]			
21	JavaScript	скрипт[2]			
22	Шаблон Prometheus	шаблон[6, 7]	`value`, `label`, `function`	вывод[8, 9]	0, 1, 2, 3
23	Prometheus в JSON	шаблон[6, 7]			0, 1, 2, 3
24	CSV в JSON	символ[2]	символ[2]	0,1	0, 1, 2, 3
25	Замена	строка поиска[2]	замена[2]		
26	Проверка неподдерживаемости	область[1]	шаблон[3, 6]		1, 2, 3
27	XML в JSON				0, 1, 2, 3

Тип предобработки	Имя	Параметр 1	Параметр 2	Параметр 3	Поддерживаемые обработчики ошибок
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[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 float)
[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, следующий 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, следующий синтаксису: `<label name>` (может быть пользовательским макросом), если `label` выбран в качестве второго параметра.
[9] Одна из агрегирующих функций: `sum`, `min`, `max`, `avg`, `count`, если `function` выбран в качестве второго параметра.
[10] Поддерживает несколько записей "Имя поля,Префикс OID,Формат", разделённых символом новой строки.

item.create

Описание

`object item.create(object/array items)`

Этот метод позволяет создавать новые элементы данных.

Note:
Веб-элементы данных нельзя создать через Zabbix API.

Note:
Этот метод доступен только для типов пользователей *Admin* и *Super admin*. Права на вызов метода могут быть отозваны в настройках роли пользователя. Дополнительную информацию смотрите в разделе Роли пользователей.

Параметры

`(object/array)` Элементы данных для создания.

В дополнение к стандартным свойствам элемента данных, метод принимает следующие параметры.

Параметр	Тип	Описание
preprocessing	array	Параметры предобработки 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 для мониторинга свободного места на диске на узле сети с ID "30074".

Запрос:

```
{
    "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
}
```

Ответ:

```
{
    "jsonrpc": "2.0",
    "result": {
        "itemids": [
            "24758"
        ]
    },
    "id": 1
}
```

Создание элемента данных инвентаризации узла сети

Создайте элемент данных агента Zabbix, чтобы заполнить поле инвентаризации узла сети "OS".

Запрос:

```
{
    "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
}
```

Ответ:

```
{
    "jsonrpc": "2.0",
    "result": {
        "itemids": [
            "24759"
        ]
    },
    "id": 1
}
```

Создание элемента данных с предобработкой

Создайте элемент данных, используя пользовательский множитель.

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 основного элемента данных с ID 24759. Допускаются зависимости только на одном и том же узле сети, поэтому основной 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 предобработкой ответа JSON.

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 script

Создайте простой сбор данных с использованием элемента данных 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(    itemIds)`

Этот метод позволяет удалять элементы данных.

> **Note:**
> Веб элементы данных нельзя удалить через Zabbix API.

> **Note:**
> Этот метод доступен только пользователям с типом доступа *Администратор* и *Супер-администратор*. Разрешения на вызов метода можно отозвать в настройках ролей 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

Удалите два элемента данных.
Зависимые элементы данных и прототипы элементов данных удаляются автоматически, если удален основной 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)`

Этот метод позволяет получать элементы данных 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 смотрите в разделе Роли пользователей.

Параметры

`(object)` Параметры, определяющие желаемый вывод.

Метод поддерживает следующие параметры.

Параметр	Тип	Описание
itemids	ID/array	Возвращать только элементы данных с указанными ID.
groupids	ID/array	Возвращать только элементы данных, принадлежащие узлам сети из указанных групп.
templateids	ID/array	Возвращать только элементы данных, принадлежащие указанным шаблонам.
hostids	ID/array	Возвращать только элементы данных, принадлежащие указанным узлам сети.
proxyids	ID/array	Возвращать только элементы данных, которые мониторятся указанными прокси.
interfaceids	ID/array	Возвращать только элементы данных, использующие указанные интерфейсы узла сети.
graphids	ID/array	Возвращать только элементы данных, используемые в указанных графиках.
triggerids	ID/array	Возвращать только элементы данных, используемые в указанных триггерах.
webitems	flag	Включать веб-элементы данных 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. Возможные значения: `true` - шаблон/узел сети/связанные шаблоны также должны иметь указанные теги; `false` - *(по умолчанию)* теги из шаблона/узла сети/связанных шаблонов игнорируются.
templated	boolean	Если установлено в `true`, возвращать только элементы данных, принадлежащие шаблонам.
monitored	boolean	Если установлено в `true`, возвращать только включённые элементы данных, принадлежащие мониторируемым узлам сети.
group	string	Возвращать только элементы данных, принадлежащие группе с указанным именем.
host	string	Возвращать только элементы данных, принадлежащие узлу сети с указанным именем.
evaltype	integer	Метод вычисления тегов. Возможные значения: 0 - *(по умолчанию)* And/Or; 2 - Or.
tags	array	Возвращать только элементы данных с указанными тегами. Формат: `[{"tag": "<tag>", "value": "<value>", "operator": "<operator>"}, ...]`. Пустой массив возвращает все элементы данных. Возможные значения operator: 0 - *(по умолчанию)* Содержит; 1 - Равно; 2 - Не содержит; 3 - Не равно; 4 - Существует; 5 - Не существует.

Параметр	Тип	Описание
with_triggers	boolean	Если установлено в `true`, возвращать только элементы данных, используемые в триггерах.
selectHosts	query	Возвращать свойство `hosts` с массивом узлов сети, которым принадлежит элемент данных.
selectInterfaces	query	Возвращать свойство `interfaces` с массивом интерфейсов узла сети, используемых элементом данных.
selectTriggers	query	Возвращать свойство `triggers` с триггерами, в которых используется элемент данных. Поддерживает `count`.
selectGraphs	query	Возвращать свойство `graphs` с графиками, содержащими элемент данных. Поддерживает `count`.
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 - *(по умолчанию)* элемент данных обнаружен, 1 - элемент данных больше не обнаруживается; `ts_delete` - `(timestamp)` время, когда элемент данных, который больше не обнаруживается, будет удалён; `ts_disable` - `(timestamp)` время, когда элемент данных, который больше не обнаруживается, будет отключён; `disable_source` - `(int)` индикатор того, был ли элемент данных отключён правилом LLD или вручную: 0 - *(по умолчанию)* отключён автоматически, 1 - отключён правилом LLD.
selectDiscoveryRule	query	Возвращать свойство `discoveryRule` с правилом LLD, создавшим элемент данных.
selectInheritedTags	query	Возвращать свойство `inheritedTags` с тегами, находящимися в шаблоне/узле сети/связанных шаблонах.
selectPreprocessing	query	Возвращать свойство `preprocessing` с параметрами предобработки элемента данных.
selectTags	query	Возвращать теги элемента данных в свойстве `tags`.
selectValueMap	query	Возвращать свойство `valuemap` с картой значений элемента данных.
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 элемент данных.
limitSelects	integer	Ограничивает количество записей, возвращаемых подзапросами. Применяется к следующим подзапросам: `selectGraphs` - результаты будут отсортированы по `name`; `selectTriggers` - результаты будут отсортированы по `description`.

Параметр	Тип	Описание
sortfield	string/array	Сортировать результат по указанным свойствам. Возможные значения: `itemid`, `name`, `key_`, `delay`, `history`, `trends`, `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	
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**, используйте вместо него `selectDiscoveryData`.

Возвращаемые значения

`(integer/array)` Возвращает одно из следующего:

- массив объектов;
- количество полученных объектов, если был использован параметр `countOutput`.

Примеры

Поиск элементов данных по ключу

Получить все элементы данных, используемые в триггерах для указанного ID узла сети, у которых ключ элемента данных содержит 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

Получить все зависимые элементы данных с узла сети с ID ”10116”, у которых ключ содержит 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 предобработки

Получите все элементы данных и их правила предобработки 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(    /     items)`

Этот метод позволяет обновлять существующие элементы данных.

Note:
Веб элементы данных нельзя обновлять через Zabbix API.

Note:
Этот метод доступен только пользователям с типом доступа *Администратор* и *Супер-администратор*. Разрешения на вызов метода можно отозвать в настройках ролей 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.

В дополнение к стандартным свойствам элемента данных, метод принимает следующие параметры.

Параметр	Type	Описание
preprocessing	array	Параметры предобработки элемента данных, которые заменяют текущие параметры предобработки. Поведение параметра: - *только для чтения* для унаследованных или обнаруженных объектов
tags	array	Теги элемента данных. Поведение параметра: - *только для чтения* 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 его статус в ”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 мастер-элемента данных. Разрешены только зависимости на одном и том же узле сети, поэтому мастер-элемент данных 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-агент

Включите 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 предобработкой

Обновите элемент данных с правилом предобработки элемента данных «В диапазоне».

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 script

Обновите элемент данных типа script, 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

Graph item object

The following objects are directly related to the `graphitem` API.

Элемент графика

Note:
Элементы графика можно изменять только через API `graph`.

Объект элемента графика имеет следующие свойства.

Свойство	Тип	Описание
gitemid	ID	ID элемента графика. Поведение свойства: - *только для чтения*
color	string	Цвет отрисовки элемента графика в виде шестнадцатеричного кода цвета.
itemid	ID	Поведение свойства: - *обязательно* для операций создания ID элемента данных.
calc_fnc	integer	Поведение свойства: - *обязательно* для операций создания 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 круговыми и разнесёнными круговыми графиками.
drawtype	integer	Стиль отрисовки элемента графика. Возможные значения: 0 - *(по умолчанию)* линия; 1 - заполненная область; 2 - жирная линия; 3 - точка; 4 - пунктирная линия; 5 - градиентная линия.
graphid	ID	ID графика, к которому принадлежит элемент графика.
sortorder	integer	Позиция элемента данных на графике. По умолчанию: начинается с ”0” и увеличивается на единицу для каждой записи.
type	integer	Тип элемента графика. Возможные значения: 0 - *(по умолчанию)* простой; 2 - сумма графика, используется только круговыми и разнесёнными круговыми графиками.
yaxisside	integer	Сторона графика, на которой будет отрисована шкала Y элемента графика. Возможные значения: 0 - *(по умолчанию)* левая сторона; 1 - правая сторона.

graphitem.get

Описание

```
integer/array graphitem.get(          )
```

Этот метод позволяет получать элементы графика в соответствии с заданными параметрами.

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) Параметры, определяющие желаемый вывод.

Метод поддерживает следующие параметры.

Параметр	Тип	Описание
graphids	ID/array	Возвращать только элементы графиков, принадлежащие указанным графикам.
itemids	ID/array	Возвращать только элементы графиков с указанными ID элементов данных.
type	integer	Возвращать только элементы графиков указанного типа. Список поддерживаемых типов элементов графиков см. 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

Получите 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 Типы данных

API Zabbix поддерживает следующие типы данных в качестве входных данных:

Type	Description
ID	Уникальный идентификатор, используемый для ссылки на сущность.
boolean	Логическое значение (`true` или `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 данные, которые 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 подзапросами.

Attention:
API Zabbix всегда возвращает значения только в виде строк или массивов.

Поведение свойств

Некоторые свойства объекта помечены короткими метками, описывающими их поведение. Используются следующие метки:

- ***только для чтения*** - значение свойства устанавливается автоматически и не может быть задано или изменено пользователем, даже в некоторых особых условиях (например, *только для чтения* для унаследованных или обнаруженных объектов);
- ***только для записи*** - значение свойства может быть задано, но после этого к нему нельзя получить доступ;
- ***постоянное*** - значение свойства можно задать при создании объекта, но нельзя изменить позже;
- ***поддерживается*** - значение свойства не обязательно задавать, но его можно задавать в некоторых особых условиях (например, *поддерживается*, если `type` установлено в ”Simple check”, ”External check”, ”SSH agent”, ”TELNET agent” или ”HTTP agent”); однако обратите внимание, что свойства с меткой *поддерживается* все равно могут быть установлены в значения по умолчанию независимо от условий;
- ***обязательное*** - значение свойства обязательно должно быть задано для всех операций (кроме операций получения) или в некоторых особых условиях (например, *обязательное* для операций создания; *обязательное*, если `operationtype` установлено в ”global script” и `opcommand_hst` не задано).

Note:
Для операций обновления свойство считается ”заданным”, если оно задается в ходе операции обновления.

Свойства, не помеченные метками, являются необязательными.

Поведение параметров

Некоторые параметры операции помечены короткими метками, описывающими их поведение для операции. Используются следующие метки:

- ***только для чтения*** - значение параметра устанавливается автоматически и не может быть задано или изменено пользователем, даже в некоторых особых условиях (например, *только для чтения* для унаследованных объектов или обнаруженных объектов);
- ***только для записи*** - значение параметра можно задать, но после этого к нему нельзя получить доступ;
- ***поддерживается*** - значение параметра не требуется задавать, но его можно задавать в некоторых особых условиях (например, *поддерживается*, если `operating_mode` объекта прокси установлен в ”passive proxy”); однако обратите внимание, что параметры со статусом *поддерживается* всё равно могут быть установлены в значения по умолчанию независимо от условий;
- ***обязательный*** - значение параметра 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 методов ”get” Следующие параметры поддерживаются всеми методами `get`:

Параметр	Тип	Описание
countOutput	boolean	Возвращать количество записей в результате вместо фактических данных.

Параметр	Тип	Описание
editable	boolean	Если установлено значение `true`, возвращать только 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 точно соответствуют заданному фильтру. Принимает объект, где ключами являются имена свойств (например, свойства объекта Host в `host.get`, свойства объекта Item в `item.get` и т. д.), а значениями — либо одно значение, либо массив значений для сопоставления. Не поддерживает свойства типа данных `text` data type. 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 должны быть возвращены. Обратите внимание, что ID 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 ID в качестве ключей в результирующем массиве.
search	object	Возвращать результаты, соответствующие заданному шаблону (без учета регистра). Принимает объект, где ключами являются имена свойств (например, свойства объекта Host в `host.get`, свойства объекта Item в `item.get` и т. д.), а значениями — строки для поиска. Если дополнительные параметры не заданы, будет выполнен поиск `LIKE "%...%"`. Поддерживает только свойства типов данных `string` и `text` data type. 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, соответствующие любому из критериев, заданных в параметре `filter` или `search`, а не всем сразу. По умолчанию: `false`.
searchWildcardsEnabled	boolean	Если установлено значение `true`, разрешает использование "*" в качестве подстановочного символа в параметре `search`. По умолчанию: `false`.
sortfield	string/array	Сортировать результат по заданным свойствам. Список свойств, которые можно использовать для сортировки, см. в описании соответствующего метода API get. Макросы перед сортировкой не раскрываются. Если значение не указано, данные будут возвращены без сортировки.
sortorder	string/array	Порядок сортировки. Если передан массив, каждое значение будет сопоставлено соответствующему свойству, заданному в параметре `sortfield`. Возможные значения: `ASC` - *(по умолчанию)* по возрастанию; `DESC` - по убыванию.
startSearch	boolean	Параметр `search` будет сравнивать начало полей, то есть вместо этого будет выполняться 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

Критические изменения история

ZBXNEXT-10069 В метод history.get добавлен новый параметр maxValueSize. Если вы запрашиваете данные для элементов данных с двоичным значением (history=5), установите maxValueSize в null; в противном случае history.get вернет значение элемента данных, усеченное до 64 КиБ.

узел сети

ZBXNEXT-10387 host.massupdate: Метод удалён. Используйте host.update вместо него.

hostgroup

ZBXNEXT-10387 hostgroup.massupdate: Метод удалён. Используйте вместо него hostgroup.update, hostgroup.massadd или hostgroup.massremove.

hostinterface

ZBXNEXT-10387 hostinterface.replacehostinterfaces: Метод удалён. Вместо него используйте hostinterface.update, hostinterface.massadd или hostinterface.massremove.

шаблон

ZBXNEXT-10387 template.massupdate: Метод удалён. Пожалуйста, используйте вместо него template.update, template.massadd или template_massremove.

templategroup

ZBXNEXT-10387 templategroup.massupdate: Метод удалён. Используйте вместо него templategroup.update, templategroup.massadd или templategroup.massremove.

Другие изменения и исправления ошибок коннектор

ZBXNEXT-10069 В объект connector добавлено новое значение item_value_type (64 - JSON).

панель мониторинга

ZBXNEXT-4769 Добавлено новое поле виджета панели мониторинга show_hostnames в виджет graph.
ZBXNEXT-9546 Добавлены новые поля виджета панели мониторинга ds.0.item_tags_evaltype, ds.0.item_tags.0.tag, ds.0.item_tags.0.operator и ds.0.item_tags.0.value в виджеты svggraph и piechart.
ZBXNEXT-10158 Добавлены новые поля виджета панели мониторинга columns.0.aggregate_columns, columns.0.combined_colu и columns.0.column_aggregate_function в виджет topitems.
ZBXNEXT-9424, ZBXNEXT-4769 Добавлен новый тип виджета панели мониторинга scatterplot.
ZBXNEXT-8160 Добавлены новые поля виджета панели мониторинга clustering_mode и clustering_zoom_level в виджет geomap.
ZBXNEXT-943 Добавлено новое поле виджета панели мониторинга ds.0.invert_values в виджет graph.

templatedashboard

ZBXNEXT-9424 Добавлен новый тип виджета шаблона панели мониторинга scatterplot.

история

ZBXNEXT-10069 В метод history.get добавлено новое значение history (6 - JSON).
ZBXNEXT-10069 Добавлен новый объект JSON history.

hostprototype

ZBXNEXT-9546 В метод hostprototype.get добавлен новый параметр selectInheritedTags.

httptest

ZBXNEXT-9546 Добавлены новые параметры inheritedTags и selectInheritedTags в метод httptest.get.

элемент данных

ZBXNEXT-10069 Добавлено новое значение value_type (6 - JSON) в объект item.
ZBXNEXT-9546 В метод item.get добавлены новые параметры inheritedTags и selectInheritedTags.

itemprototype

ZBXNEXT-10069 В объект itemprototype добавлено новое значение value_type (6 - JSON).
ZBXNEXT-9546 В метод itemprototype.get добавлен новый параметр selectInheritedTags.

шаблон

В ZBXNEXT-9546 в метод template.get добавлены новые параметры inheritedTags и selectInheritedTags.

триггер

В метод trigger.get добавлены новые параметры `inheritedTags` и `selectInheritedTags` (ZBXNEXT-9546).

triggerprototype

В ZBXNEXT-9546 в метод triggerprototype.get был добавлен новый параметр `selectInheritedTags`.

triggertag

ZBXNEXT-5467 Добавлено свойство `automatic` в объект тега триггера.

userdirectory

В объект каталога пользователей добавлены свойства `idp_certificate`, `sp_private_key`, `sp_certificate`, `idp_certificate_hash`, `sp_private_key_hash` и `sp_certificate_hash` (ZBXNEXT-9834).

Приложение 3. Изменения в 8.0

8.0.1 Эта версия ещё не выпущена.

19 Расширения

Обзор Хотя Zabbix предлагает множество функций, 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, используя встроенные параметры расширения (траппер элементы данных, пользовательские параметры и т. д.) или путем 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 Траппер элементы данных

Траппер элементы данных — это элементы, которые принимают входящие данные, а не запрашивают их. Траппер элементы данных полезны для отправки определенных данных на сервер Zabbix или прокси Zabbix, например, периодические данные о доступности и производительности в случае длительных пользовательских сценариев. Вы можете сделать это с помощью утилиты командной строки Zabbix sender или путем реализации коммуникационного протокола на основе JSON (аналогично тому, что используется в Zabbix sender). Отправка данных к серверу Zabbix также возможна при помощи метода API `history.push`.

Внешние проверки

Внешняя проверка — это элемент для выполнения проверок, запуском скрипта, например сценария оболочки или двоичного файла.

Внешние проверки выполняются сервером Zabbix или прокси (когда узел сети контролируется прокси), и не требуют наличия агента, работающего на отслеживаемом узле сети.

Пользовательские параметры

Пользовательские параметры — это заданная пользователем команда (связанная задаваемым пользователем ключом), который при выполнении может получить необходимые вам данные с узла сети, на котором работает агент Zabbix. Пользовательские параметры полезны для настройки элементов агента или агента 2, которые не определены в Zabbix.

Элементы данных Zabbix агент `system.run[]`

Zabbix элемент данных агента `system.run[]` — это элемент данных для пользовательской команды (связанной с предопределённым ключом `system.run[]`, например, `system.run[myscript.sh]`), которая может быть выполнена на узле сети, где запущен Zabbix агент.

Примечание: элементы данных `system.run[]` по умолчанию отключены и, если используются, должны быть включены (разрешены) и определены в конфигурационном файле Zabbix агент или agent 2 (параметр конфигурации `AllowKey`).

Attention:
Пользовательские команды в таких элементах, как внешние проверки, пользовательские параметры и элементы Zabbix агента `system.run[]` выполняются от пользователя ОС, который используется для запуска компонентов Zabbix. Для выполнения этих команд этот пользователь должен иметь необходимые разрешения.

Элементы данных HTTP-агента

Элемент данных HTTP-агента — это элемент данных для выполнения запросов данных через HTTP/HTTPS. Элементы данных HTTP-агента полезны для отправки запросов к конечным точкам HTTP для получения данных из таких сервисов, как *Elasticsearch* и *OpenWeatherMap*, для проверки статуса Zabbix API или статуса веб-сервера Apache или Nginx и т. д. Элементы данных HTTP (с включенной опцией разрешения получения трапов) так же могут функционировать как траппер элементы данных.

Скриптовые элементы данных

Скриптовый элемент данных — это элемент 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, с последующей передачей преобразованного значения второму HTTP-вызову.

Note:
Элементы HTTP-агента и скриптовые элементы данных поддерживаются Zabbix-сервером и прокси-сервером, и не требуют наличия агента, работающего на отслеживаемом узле сети.

Продвинутые расширения Загружаемые модули

Загружаемые модули, написанные на языке C, представляют собой универсальный и ориентированный на производительность вариант расширения функциональности компонентов Zabbix (сервер, прокси, агент) на платформах UNIX. Загружаемый модуль — это, по сути, общая библиотека, используемая демоном Zabbix и загружаемая при запуске. Библиотека должна содержать определенные функции, чтобы процесс Zabbix мог обнаружить, что файл действительно является модулем, который он может загружать и с которым он может работать.

Загружаемые модули имеют набор преимуществ, включая возможность добавлять новые метрики или реализовывать любую другую логику (для примера, экспорта данных истории Zabbix), отличную 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 базовой основы кода Zabbix.

Загружаемые 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, расширения, такие как пользовательские параметры, элементы агента `system.run[]` Zabbix и внешние проверки, будут влиять на производительность. Загружаемые модули предлагают способ расширения функциональности Zabbix без ущерба для производительности.

Плагины

Плагины расширяют возможности мониторинга Zabbix агент 2. Написанные на языке программирования Go, они предоставляют альтернативу загружаемым модулям (написанным на C).

Плагин — это пакет Go, который определяет структуру и реализует один или несколько интерфейсов плагина (*Exporter*, *Collector*, *Configurator*, *Runner*, *Watcher*). Поддерживаются два типа плагинов Zabbix агент 2:

- Встроенные плагины (поддерживаются начиная с Zabbix 4.4.0)
- Загружаемые плагины (поддерживаются начиная с Zabbix 6.0.0)

См. список плагинов, предоставляемых Zabbix.

Инструкции и руководства по созданию собственных плагинов см. в разделе Центр разработчика.

Настройка оповещений Вебхуки

Вебхук это способ оповещения Zabbix, который позволяет расширить возможности оповещения Zabbix к внешнему программному обеспечению, такому как системы службы поддержки, чаты или мессенджеры. Подобно элементам сценария, вебхуки полезны для выполнения HTTP-вызовов с использованием специального кода JavaScript, например, для отправки уведомлений на различные платформы, такие как Microsoft Teams, Discord и Jira. Также возможно вернуть некоторые данные (например, о созданных заявках в службу поддержки), которые затем отображаются в Zabbix.

Доступны существующие вебхуки в Zabbix Git repository. Для разработки собственного вебхука см. [Руководство по разработке веб-перехватчиков] (https://www.zabbix.com/documentation/guidelines/en/webhooks).

Пользовательские скрипты оповещений

Пользовательский скрипт оповещения это способ оповещения Zabbix, который предоставляет возможность создать альтернативный способ (скрипт) оповещений Zabbix. Пользовательские скрипты оповещений полезны, если вас не устраивают существующие способы оповещения в Zabbix.

Кастомизация веб-интерфейса Пользовательские темы

Можно изменить внешний вид веб-интерфейса Zabbix, используя пользовательские темы. См. инструкции по созданию и применению собственных тем.

Внешние модули

Внешние модули предоставляют возможность расширить функциональность веб-интерфейса Zabbix путем добавления сторонних модулей или разработки 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:

- Действие операции
- Действие хоста вручную
- Действие по событию вручную

Глобальные сценарии полезны во многих случаях. Например, если настроены операции действий или действия хоста вручную, вы можете использовать глобальные сценарии для автоматического или ручного выполнения удаленных 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 по событиям вручную: вы можете использовать глобальные сценарии для управления заявками о проблемах во внешних системах.

Глобальные скрипты могут выполняться Zabbix-сервером, Zabbix-прокси или агентом.

> **Attention:**
> Пользовательские команды выполняются от пользователя ОС, который используется для запуска компонентов Zabbix. Для выполнения этих команд этот пользователь должен иметь необходимые разрешения.

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 историческим и трендовым данным 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 определенную задачу. Доступные методы, а также обзор функций, предоставляемых Zabbix API, см. метод Zabbix API.

1 Загружаемые модули

Обзор

Загружаемые модули предлагают ориентированный на производительность способ расширения функциональности Zabbix.

Вы можете расширять функциональность Zabbix многими способами, например с помощью пользовательских параметров, внешних проверок и элементов данных агента Zabbix system.run. Все они работают очень хорошо, но имеют один существенный недостаток, а именно fork(). Zabbix должен создавать новый процесс каждый раз, когда обрабатывает пользовательскую метрику, что плохо сказывается на производительности. Обычно это не является большой проблемой,

однако может стать серьезной при мониторинге встроенных систем, при большом количестве отслеживаемых параметров или при использовании тяжелых скриптов со сложной логикой либо длительным временем запуска.

Поддержка загружаемых модулей предоставляет способы расширения агента Zabbix, сервера и прокси без ущерба для производительности.

Загружаемый модуль по сути представляет собой разделяемую библиотеку, используемую демоном Zabbix и загружаемую при запуске. Библиотека должна содержать определенные функции, чтобы процесс Zabbix мог определить, что файл действительно является модулем, который можно загрузить и использовать.

Загружаемые модули имеют ряд преимуществ. Очень важны высокая производительность и возможность реализовать любую логику, но, возможно, самым важным преимуществом является возможность разрабатывать, использовать и распространять модули Zabbix. Это способствует беспроблемному сопровождению и помогает проще внедрять новую функциональность независимо от основной кодовой базы Zabbix.

Лицензирование модулей и их распространение в бинарной форме регулируются лицензией AGPL-3.0 (модули динамически связываются с Zabbix во время выполнения и используют заголовочные файлы Zabbix; весь код Zabbix лицензируется по AGPL-3.0 начиная с Zabbix 7.0). Бинарная совместимость со стороны Zabbix не гарантируется.

Стабильность API модулей гарантируется в течение одного цикла релиза Zabbix LTS (Long Term Support). Стабильность API Zabbix не гарантируется (технически из модуля можно вызывать внутренние функции Zabbix, но нет гарантии, что такие модули будут работать).

API модулей

Чтобы общая библиотека рассматривалась как модуль Zabbix, она должна реализовывать и экспортировать несколько функций. В настоящее время в API модулей Zabbix есть шесть функций, только одна из которых является обязательной, а остальные пять — необязательными.

Обязательный интерфейс

Единственной обязательной функцией является **zbx_module_api_version()**:

```
int zbx_module_api_version(void);
```

Эта функция должна возвращать версию API, реализованную данным 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 эту константу. Старая константа ZBX_MODULE_API_VERSION_ONE, использовавшаяся для этой цели, теперь определена равной ZBX_MODULE_API_VERSION для сохранения совместимости исходного кода, однако её 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 выполнять необходимую инициализацию модуля (если требуется). В случае успеха она должна возвращать ZBX_MODULE_OK. В противном случае она должна возвращать 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, подробности см. в разделе ниже. Список завершается структурой ZBX_METRIC, у которой поле ”key” имеет значение 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 конфигурационном файле Zabbix, которым должны подчиняться 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 callback-функции, которые сервер Zabbix будет использовать для экспорта истории различных типов данных. Callback-функции предоставляются как поля структуры ZBX_HISTORY_WRITE_CBS; поля могут иметь значение NULL, если модуль не заинтересован в истории определенного типа.

```
int zbx_module_uninit(void);
```

Эта функция должна выполнять необходимую деинициализацию (если требуется), например освобождение выделенных ресурсов, закрытие файловых дескрипторов и т. д.

Все функции вызываются один раз при запуске Zabbix во время загрузки модуля, за исключением zbx_module_uninit(), которая вызывается один раз при завершении работы Zabbix во время выгрузки модуля.

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 ли элемент данных параметры), **function** — функция C, реализующая элемент данных (например, "zbx_module_dummy_random"), а **test_param** — список параметров, который будет использоваться при запуске Zabbix агент 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. Подробную информацию о том, как получать данные из AGENT_REQUEST и как задавать данные в AGENT_RESULT, смотрите в примере модуля "dummy" ниже.

Предоставление callback-функций экспорта истории

Attention:
Экспорт истории через модуль больше не поддерживается Zabbix прокси.

Модуль может определять функции для экспорта данных истории по типам: числовой (float), числовой (unsigned), символьный, текстовый и журнал:

```
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 экспортируемых данных истории, "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 Zabbix сервера в конце процедуры синхронизации истории после того, как данные будут записаны в базу данных Zabbix и сохранены 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 Zabbix серверу продолжать работу.

Сборка модулей

В настоящее время модули предполагается собирать внутри дерева исходного кода Zabbix, поскольку API модулей зависит от некоторых структур данных, определённых в заголовочных файлах Zabbix.

Наиболее важным заголовочным файлом для загружаемых модулей является **include/module.h**, в котором определены

эти структуры данных. Другими необходимыми системными заголовочными файлами, обеспечивающими корректную работу **include/module.h**, являются **stdlib.h** и **stdint.h**.

С учётом этой информации всё 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 include. Подробности см. в примере модуля "dummy" ниже.

Ещё одним полезным заголовочным файлом является **include/zbxcommon.h**, в котором определена функция **zabbix_log()**, которую можно использовать для журналирования и отладки.

Параметры конфигурации

Zabbix агент, сервер и прокси поддерживают два параметра для работы с модулями:

- LoadModulePath – полный путь к расположению загружаемых модулей
- LoadModule – модуль(и), загружаемые при запуске. Модули должны находиться в каталоге, указанном в LoadModulePath, либо путь должен предшествовать имени модуля. Если указанный перед именем путь является абсолютным (начинается с '/'), то LoadModulePath игнорируется. Допускается указывать несколько параметров LoadModule.

Например, чтобы расширить Zabbix агент, 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, тогда как dummy.so будет загружен из /usr/local/lib/zabbix. Агент не запустится, если модуль отсутствует, при некорректных правах доступа или если разделяемая библиотека не является модулем Zabbix.

Настройка веб-интерфейса

Загружаемые модули поддерживаются Zabbix агентом, сервером и прокси. Поэтому тип элемента данных в веб-интерфейсе Zabbix зависит от того, где 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 настройки веб-интерфейса. Если модуль успешно загружен сервером и предоставляет функцию **zbx_module_history_write_cbs()**, которая возвращает как минимум одну callback-функцию, отличную от NULL, то экспорт истории будет включен автоматически.

Тестовый модуль

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 и может использоваться как шаблон для ваших собственных модулей.

После запуска ./configure в корневом каталоге 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:                                                 *
 *                                                                            *
 * Return value: ZBX_MODULE_API_VERSION -     module.h,                 *
 *                         ;                      ,     *
 *               Zabbix                                    *
 *                                                                 *
 *                                                                            *
 ******************************************************************************/
int zbx_module_api_version(void)
{
    return ZBX_MODULE_API_VERSION;
}

/******************************************************************************
 *                                                                            *
 * Function: zbx_module_item_timeout                                          *
 *                                                                            *
 * Purpose:               -                          *
 *                                                                            *
 * Parameters: timeout -   -        , 0 -   -                  *
 *                                                                            *
 ******************************************************************************/
void    zbx_module_item_timeout(int timeout)
{
    item_timeout = timeout;
}

/******************************************************************************
 *                                                                            *
```

```
 * Function: zbx_module_item_list                                             *
 *                                                                            *
 * Purpose:                          ,                    *                   *
 *                                                                        *   *
 *                                                                            *
 * Return value:                                                  *           *
 *                                                                            *
 ******************************************************************************/
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_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:                                              *                    *
 *                                                                            *
 * Parameters: request -      ,                              *                *
 *                                                                        *   *
 *              request->key -                                    *           *
 *              request->nparam -                                      *     *
 *              request->params[N-1] -                                *       *
 *                                                                        *   *
 *              request->types[N-1] -                           :  *         *
 *                  REQUEST_PARAMETER_TYPE_UNDEFINED (           )    *       *
 *                  REQUEST_PARAMETER_TYPE_ARRAY (   )                    *  *
 *                  REQUEST_PARAMETER_TYPE_STRING (               ) *        *
 *                                                                            *
 *             result -      ,                                  *             *
 *                                                                            *
 * Return value: SYSINFO_RET_FAIL -                 ,          *             *
 *                                             zabbix              *          *
 *                                                                    *       *
 *               SYSINFO_RET_OK -                                        *    *
 *                                                                            *
```

```
 * Comment: get_rparam(request, N-1)                                *
 *                  N-      ,       0 (         ).        *
 *               ,           ,           request->nparam. *
 *                     get_rparam_type(request, N-1)              *
 *                                 .                          *
 *                                                                                *
 **********************************************************************************/
static int  dummy_random(AGENT_REQUEST *request, AGENT_RESULT *result)
{
    char    *param1, *param2;
    int from, to;

    if (2 != request->nparam)
    {
        /*                              */
        SET_MSG_RESULT(result, strdup("Invalid number of parameters."));
        return SYSINFO_RET_FAIL;
    }

    param1 = get_rparam(request, 0);
    param2 = get_rparam(request, 1);

    /*                                          */
    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:                                                  *
 *                                                        *
 *                                                                  *
 *                                                                                *
 * Return value: ZBX_MODULE_OK -                                          *
 *               ZBX_MODULE_FAIL -                          *
 *                                                                                *
 * Comment:                      ZBX_MODULE_FAIL                 *
 *                                                                                *
 **********************************************************************************/
int zbx_module_init(void)
{
    /*            dummy.random */
    srand(time(NULL));

    return ZBX_MODULE_OK;
}

/**********************************************************************************
 *                                                                                *
 * Function: zbx_module_uninit                                                    *
```

```
 *                                                                            *
 * Purpose:                                                 *
 *                                                ,      *
 *                                                                  *
 *                                                                            *
 * Return value: ZBX_MODULE_OK -                                          *
 *               ZBX_MODULE_FAIL -                                *
 *                                                                            *
 ******************************************************************************/
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-                              float,  *
 *          integer, string, text   log                          *
 *                                                                            *
 * Parameters: history     -                                  *
 *             history_num -                   history          *
 *                                                                            *
 ******************************************************************************/
static void dummy_history_float_cb(const ZBX_HISTORY_FLOAT *history, int history_num)
{
    int i;

    for (i = 0; i < history_num; i++)
    {
        /*       -     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++)
    {
        /*       -     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++)
    {
        /*       -     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++)
    {
        /*       -     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++)
    {
        /*       -     history[i].itemid, history[i].clock, history[i].ns, history[i].value, ... */
    }
}

/******************************************************************************
 *                                                                            *
 * Function: zbx_module_history_write_cbs                                     *
 *                                                                            *
 * Purpose:                    ,      Zabbix            *
 *                                                        *
 *                                                                            *
 * Return value:                  callback-     (            *
 *               NULL,                                    *
 *                  )                                                 *
 *                                                                            *
 ******************************************************************************/
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 param1-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 чтение связанных с модулем параметров конфигурации из *zabbix_agentd.conf*. В настоящее время это не поддерживается. Если вашему модулю необходимо использовать какие-либо параметры конфигурации, вам, вероятно, следует реализовать разбор файла конфигурации, специфичного для модуля.

3 Модули веб-интерфейса

Обзор

Можно улучшить функциональность веб-интерфейса Zabbix, добавив сторонние модули или разработав собственные

модули без необходимости изменения исходного кода Zabbix.

Обратите внимание, что код модуля будет работать с теми же привилегиями, что и исходный код Zabbix. Это означает, что:

- сторонние модули могут быть вредоносными. Вы должны доверять модулям, которые вы устанавливаете;
- ошибки в коде стороннего модуля могут привести к сбою веб-интерфейса. Если это произойдёт, просто удалите код модуля из веб-интерфейса. Как только вы перезагрузите веб-интерфейс Zabbix, вы увидите уведомление о том, что некоторые модули отсутствуют. Перейдите к администрированию модулей (в разделе *Администрирование* → *Общие* → *Модули*) и нажмите *Сканировать директорию*, чтобы удалить несуществующие модули из базы данных.

Установка

Пожалуйста, всегда читайте руководство по установке для конкретного модуля. Рекомендуется устанавливать новые модули по одному, чтобы легко выявлять сбои.

Непосредственно перед установкой модуля:

- Убедитесь, что вы загрузили модуль из доверенного источника. Установка вредоносного кода может привести к таким последствиям, как потеря данных
- Разные версии одного и того же модуля (с одинаковым ID) могут быть установлены параллельно, но одновременно может быть включена только одна версия

Шаги по установке модуля:

- Распакуйте модуль в отдельную папку внутри папки `modules` в веб-интерфейсе Zabbix
- Убедитесь, что папка модуля содержит как минимум файл manifest.json
- Перейдите в раздел Управление модулями 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 модуль не появился в списке, убедитесь, что файл 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 вложенных каталогов, а также права на чтение (`r--`) для всех файлов внутри.
Веб-интерфейс перестал работать	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 раздел Управление модулями и снова нажмите *Сканировать каталог*, чтобы удалить из базы данных несуществующие модули.
Появляется сообщение об ошибке об одинаковом namespace, ID или actions	Новый модуль попытался зарегистрировать namespace, ID или actions, 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 см. в Центре разработчиков.

20 Приложения

Пожалуйста, используйте боковую панель для доступа к содержимому раздела «Приложения».

1 Установка и настройка

Пожалуйста, используйте боковую панель для доступа к содержимому этого раздела.

1 Создание базы данных

Обзор

База данных Zabbix должна быть создана во время установки сервера или прокси Zabbix.

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, создавая роли/пользователей базы данных с минимальными привилегиями, см. рекомендации по созданию базы данных для каждой поддерживаемой базы данных:

> - MySQL/MariaDB
> - PostgreSQL/TimescaleDB
>
> Для настройки защищённых TLS-соединений см. Защищённое подключение к базе данных.

UTF-8 — единственная кодировка, поддерживаемая Zabbix. Известно, что она работает без каких-либо уязвимостей безопасности. Пользователи должны учитывать, что при использовании некоторых других кодировок существуют известные проблемы безопасности. О переходе на UTF-8 см. Исправление набора символов и сопоставления базы данных Zabbix. См. также Ограничения фильтрации с сопоставлениями utf8mb4.

> **Note:**
> Если установка выполняется из репозитория Zabbix Git, перед переходом к следующим шагам необходимо выполнить следующую команду:

 `make dbschema`

MySQL/MariaDB

Для корректной работы сервера/прокси Zabbix с базой данных MySQL поддерживаются кодировки utf8 (также известная как utf8mb3) и utf8mb4 (с сортировками utf8_bin и utf8mb4_bin соответственно). Для новых установок рекомендуется использовать utf8mb4.

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 бинарное журналирование, отсутствуют привилегии суперпользователя и `log_bin_trust_function_creators = 1` не задан в конфигурационном файле MySQL.

Если вы устанавливаете из **пакетов** Zabbix, перейдите к инструкциям для вашей платформы.

Если вы устанавливаете Zabbix из исходного кода:

- Создайте и настройте базу данных и пользователя.

```
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 кодировку utf8mb4 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

У вас должен быть пользователь базы данных с правами на создание объектов базы данных.

Если вы устанавливаете из **пакетов** Zabbix, перейдите к инструкциям для вашей платформы.

Если вы устанавливаете Zabbix из исходного кода:

- Создайте пользователя базы данных.

Следующая команда оболочки создаст пользователя zabbix. При появлении запроса укажите пароль и повторите его (обратите внимание, что сначала у вас могут запросить пароль sudo):

```
sudo -u postgres createuser --pwprompt zabbix
```

- Создайте базу данных.

Следующая команда оболочки создаст базу данных zabbix (последний параметр), где ранее созданный пользователь будет владельцем (-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 исходного кода Zabbix). Для базы данных прокси Zabbix следует импортировать только schema.sql (без images.sql и data.sql).

```
cd database/postgresql
cat schema.sql | sudo -u zabbix psql zabbix
####           ,                          Zabbix
cat images.sql | sudo -u zabbix psql zabbix
cat data.sql | sudo -u zabbix psql zabbix
```

Attention:
Приведенные выше команды даны в качестве примера и будут работать в большинстве установок GNU/Linux. В зависимости от конфигурации вашей системы/базы данных вы можете использовать другие команды, например:

 psql -U <username>

 Если у вас возникли какие-либо трудности при настройке базы данных, обратитесь к вашему администратору базы данных.

TimescaleDB

Инструкции по созданию и настройке TimescaleDB приведены в отдельном разделе.

SQLite

Использование SQLite поддерживается только для **Zabbix прокси**!

База данных будет автоматически создана, если она не существует.

Вернуться к разделу установки.

2 Исправление набора символов и сопоставления для базы данных Zabbix

MySQL/MariaDB

Исторически MySQL и производные от него СУБД использовали 'utf8' как псевдоним для utf8mb3 — собственной 3-байтовой реализации стандартного UTF8 в MySQL, который является 4-байтовым. Начиная с MySQL 8.0.28 и MariaDB 10.6.1, набор символов 'utf8mb3' считается устаревшим, и в какой-то момент его поддержка будет прекращена, тогда как 'utf8' станет ссылкой на 'utf8mb4'. Начиная с Zabbix 6.0, поддерживается 'utf8mb4'. Во избежание проблем в будущем настоятельно рекомендуется использовать 'utf8mb4'. Еще одним преимуществом перехода на 'utf8mb4' является поддержка дополнительных символов Unicode.

Warning:
Поскольку версии до Zabbix 6.0 не поддерживают utf8mb4, перед выполнением преобразования в utf8mb4 сначала обязательно обновите сервер Zabbix и схему БД до версии 6.0.x или новее.

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 видно, здесь набор символов не ’utf8mb4’, а сопоставление не ’utf8mb4_bin’, поэтому их необходимо 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>';
If MariaDB →    set innodb_strict_mode = OFF;
                CALL zbx_convert_utf8();
If MariaDB →    set innodb_strict_mode = ON;
                drop procedure zbx_convert_utf8;
```

Обратите внимание, что ’utf8mb4’, как ожидается, будет занимать немного больше места на диске.

7. Если ошибок нет, вы можете создать резервную копию базы данных с исправленной базой данных.

8. Запустите Zabbix.

3 Обновление базы данных до первичных ключей

Обзор

В этом разделе приведены инструкции по ручному обновлению таблиц в существующих установках для использования первичных ключей.

Переход на первичные ключи оптимизирует индексацию данных и доступ к ним, что может ускорить выполнение запросов и сэкономить место. Это также улучшает управление данными и синхронизацию в кластерных конфигурациях, помогая с масштабированием и обеспечивая надежность системы даже в случае отказа некоторых серверов.

Attention:
Инструкции, приведенные на этой странице, предназначены для опытных пользователей и могут потребовать адаптации под вашу конкретную конфигурацию. Обновление до первичных ключей может занять много времени и потребовать значительных ресурсов. Убедитесь, что доступно достаточно свободного места на диске; в зависимости от размера вашей базы данных и хранимых данных, процесс может потребовать до 2,5 раз больше места, чем в настоящее время занимают таблицы истории.

Первичные ключи используются для всех таблиц в новых установках, начиная с Zabbix 6.0.

Автоматическое обновление базы данных до первичных ключей не предусмотрено; однако существующие установки можно обновить вручную **после** обновления сервера Zabbix до версии 6.0 или новее.

Attention:
Начиная с Zabbix 7.0, обновление таблиц до первичных ключей также обновляет таблицы для использования типов данных double precision.

 Если у вас Zabbix 7.0 (или более поздняя версия), таблицы уже используют double precision. Однако инструкциям на этой странице все равно можно следовать, чтобы обновить таблицы 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 смотрите в разделе Переход на числовые значения расширенного диапазона документации Zabbix 7.0.

Инструкции доступны для:

- MySQL
- PostgreSQL
- TimescaleDB

Важные примечания

Чтобы выполнить обновление базы данных:

1. Остановите сервер Zabbix.

Настоятельно рекомендуется остановить сервер Zabbix на время обновления. Однако при крайней необходимости 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 преимуществ от этого обновления.

Если база данных использует секционирование, 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.

MySQL

Экспорт и импорт необходимо выполнять в tmux/screen, чтобы гарантировать, что сессия не будет прервана.

См. также: Важные примечания

MySQL 8.4+ с mysqlsh

Этот метод можно использовать с работающим сервером Zabbix, однако на время обновления сервер рекомендуется остановить. 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. Если используется подключение 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 (CSVPATH можно изменить при необходимости):

```
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 получаете сообщение "JavaScript is not supported", значит, ваша установка MySQL Shell не поддерживает JS. В этом случае установите официальный пакет MySQL Shell от Oracle (или соберите его из исходного кода), чтобы был включен режим JavaScript.

- Следуйте инструкциям после миграции, чтобы удалить старые таблицы.

MariaDB/MySQL 8.4+ без mysqlsh

Этот метод 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 как любой пользователь с привилегиями FILE.
- Если миграция выполняется 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

Перед миграцией данных необходимо отключить `max_execution_time` (в MySQL) или `max_statement_time` (в MariaDB), 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;
```

Чтобы удалить старые таблицы, следуйте инструкциям после миграции.

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* содержит путь к каталогу, экспорт/импорт будет выполняться для файлов в этом каталоге. В этом случае соответствующим образом измените пути к файлам в запросах или установите для *secure_file_priv* 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, установите его в путь, содержащий экспортированные данные таблиц (в примере выше это '/var/lib/mysql-files/').

Для получения дополнительной информации см. документацию MySQL или документацию MariaDB.

Перед экспортом данных необходимо отключить `max_execution_time` (в MySQL) или `max_statement_time` (в MariaDB), 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
```

Чтобы удалить старые таблицы, следуйте инструкциям после миграции.

PostgreSQL

Экспорт и импорт необходимо выполнять в tmux/screen, чтобы избежать разрыва сессии. 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 history, импортируйте её 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;
```

Советы по повышению производительности INSERT см.: 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

- Экспортируйте текущую историю, импортируйте её во временную таблицу, затем вставьте данные 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 после миграции, чтобы удалить старые таблицы.

PostgreSQL + TimescaleDB

Экспорт и импорт необходимо выполнять в tmux/screen, чтобы избежать разрыва сессии. Во время обновления сервер Zabbix должен быть остановлен.

См. также: Важные примечания

- Переименуйте таблицы с помощью `history_upgrade_prepare.sql`.
  - Если сжатие включено (в установке по умолчанию), запустите скрипт из `/usr/share/zabbix/sql-scripts/postgresql`

    ```
    cat /usr/share/zabbix/sql-scripts/postgresql/timescaledb/option-patches/with-compression/history_
    ```

  - Если сжатие отключено, запустите скрипт из `/usr/share/zabbix/sql-scripts/postgresql/timescaledb/option-p`

    ```
    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 установке по умолчанию), запустите скрипты из `/usr/share/zabbix/sql-scripts/postgres`

    ```
    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, запустите 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: Советы по повышению производительности INSERT.

- Следуйте инструкциям после миграции, чтобы удалить старые таблицы.

После миграции

Для всех баз данных после завершения миграции выполните следующие действия:

- Убедитесь, 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;
```

См. также

- Подготовка таблицы auditlog к секционированию

4 Подготовка таблицы auditlog к секционированию

Обзор

Некоторые базы данных (например, MySQL) требуют, чтобы столбец секционирования был частью уникального ограничения таблицы. Поэтому, чтобы секционировать таблицу `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, приведённые на этой странице, предназначены для опытных пользователей. Обратите внимание, что эти инструкции, возможно, потребуется скорректировать в соответствии с вашей конкретной конфигурацией. Изменение первичного ключа также может быть несовместимо с будущими патчами обновления, поэтому в дальнейшем может потребоваться выполнять обновления вручную.

 Изменение первичного ключа может быть ресурсоёмкой операцией, которая занимает много времени в зависимости от размера таблицы auditlog. На время выполнения изменения рекомендуется остановить сервер Zabbix и перевести веб-интерфейс Zabbix в режим обслуживания. Однако при крайней необходимости существует способ изменить первичный ключ без простоя (см. ниже).

Секционирование таблицы auditlog может улучшить, например, очистку истории в крупных установках. Хотя очистка истории Zabbix в настоящее время не может использовать преимущества секционированных таблиц (за исключением TimescaleDB), вы можете отключить очистку истории Zabbix и удалять секции с помощью скриптов.

Начиная с Zabbix 7.0, таблица auditlog для TimescaleDB была преобразована в гипертаблицу, что позволяет процессу очистки истории удалять данные по чанкам. Чтобы обновить существующую таблицу auditlog до гипертаблицы, см. Обновление схемы TimescaleDB.

MySQL

Важные замечания по перестроению индексов

MySQL автоматически перестраивает индексы для первичного ключа во время операции ALTER TABLE. Однако настоятельно рекомендуется также вручную перестроить индексы с помощью инструкции OPTIMIZE TABLE, чтобы обеспечить оптимальную производительность базы данных.

Перестроение индексов может временно потребовать дополнительное дисковое пространство в объеме, равном размеру самой таблицы. Чтобы получить текущий размер данных и индексов, можно выполнить следующие инструкции:

```
ANALYZE TABLE auditlog;
SHOW TABLE STATUS LIKE 'auditlog';
```

Если доступное дисковое пространство ограничено, следуйте инструкциям Изменение первичного ключа без простоя. Также доступны другие варианты:

- Увеличение параметра MySQL sort_buffer_size может помочь уменьшить использование дискового пространства при ручном перестроении индексов. Однако изменение этой переменной может повлиять на общее использование памяти базой данных.
- Рассмотрите возможность освобождения места путем удаления потенциально ненужных данных.
- Рассмотрите возможность уменьшения параметра *Период хранения данных* housekeeper перед выполнением housekeeper.

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, см. Важные примечания по перестроению индексов).

```
OPTIMIZE TABLE auditlog;
```

Изменение первичного ключа без простоя

Ручной метод изменения первичного ключа описан здесь. В качестве альтернативы можно использовать набор инструментов pt-online-schema-change от Percona. Этот набор инструментов автоматически выполняет следующие действия, а также минимизирует объем пространства, используемого для изменения таблицы auditlog.

1. Создать новую таблицу с новым первичным ключом и создать 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. Поменять таблицы местами.

```
RENAME TABLE auditlog TO auditlog_old, auditlog_new TO auditlog;
```

3. Скопировать данные из старой таблицы в новую таблицу.

```
INSERT INTO auditlog SELECT * FROM auditlog_old;
```

Это можно сделать по частям (несколькими инструкциями `INSERT INTO` с условиями `WHERE clock` по мере необходимости), чтобы избежать чрезмерного использования ресурсов.

4. Удалить старую таблицу.

```
DROP TABLE auditlog_old;
```

PostgreSQL

Важные примечания по перестроению индексов

PostgreSQL автоматически перестраивает индексы для первичного ключа во время операции `ALTER TABLE`. Однако настоятельно рекомендуется также вручную перестроить индексы с помощью инструкции `REINDEX TABLE CONCURRENTLY`, чтобы обеспечить оптимальную производительность базы данных.

Для перестроения индексов может временно потребоваться до трёхкратного объёма дискового пространства, используемого индексами в настоящий момент. Чтобы получить текущий размер индексов, вы можете выполнить следующий запрос:

```
SELECT pg_size_pretty(pg_indexes_size('auditlog'));
```

Если доступное дисковое пространство вызывает опасения, следуйте инструкциям Изменение первичного ключа без простоя. Также доступны и другие варианты:

- Увеличение параметра PostgreSQL `maintenance_work_mem` может помочь уменьшить использование дискового пространства при ручном перестроении индексов. Однако изменение этой переменной может повлиять на общее использование памяти базой данных.
- Если у вас есть другой диск или табличное пространство с большим объёмом доступного места, можно рассмотреть изменение расположения временного хранилища для перестроения индекса. Вы можете задать параметр PostgreSQL `temp_tablespaces`, чтобы указать другое табличное пространство для временных объектов.
- Рассмотрите возможность освобождения места путём удаления потенциально ненужных данных.
- Рассмотрите возможность уменьшения параметра *Период хранения данных* housekeeper перед выполнением housekeeper.

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, см. 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 таблицу.

```
INSERT INTO auditlog SELECT * FROM auditlog_old;
```

Это можно сделать по частям (несколькими инструкциями `INSERT INTO` с условиями `WHERE clock` по мере необходимости), чтобы избежать чрезмерного использования ресурсов.

4. Удалите старую таблицу.

```
DROP TABLE auditlog_old;
```

См. также

- Обновление базы данных до первичных ключей

5 Безопасное подключение к базе данных

Обзор

В этом разделе приведены шаги по настройке Zabbix и примеры конфигурации для защищённых TLS-соединений между:

База данных	Компоненты Zabbix
MySQL	веб-интерфейс Zabbix, сервер Zabbix, прокси Zabbix
PostgreSQL	веб-интерфейс Zabbix, сервер Zabbix, прокси Zabbix

Чтобы настроить шифрование соединений в СУБД, подробную информацию см. в официальной документации поставщиков:

- MySQL: исходный сервер базы данных и сервер реплики.
- MySQL: групповая репликация и т. д. серверов баз данных.
- PostgreSQL параметры шифрования.

Все примеры основаны на GA-выпусках MySQL CE (8.0) и PgSQL (13), доступных через официальные репозитории, с использованием CentOS 8.

Требования

Для настройки шифрования требуется следующее:

- Операционная система, поддерживаемая разработчиком, с OpenSSL >=1.1.X или альтернативой.

> **Note:**
> Рекомендуется избегать ОС со статусом end-of-life, особенно в случае новых установок

- СУБД (RDBMS), установленная и поддерживаемая из официального репозитория, предоставляемого разработчиком. Операционные системы часто поставляются с устаревшими версиями программного обеспечения баз 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 включает использование TLS-соединения с базой данных от сервера/прокси Zabbix и веб-интерфейса к базе данных:

- `required` — подключение с использованием TLS в качестве транспортного режима без проверки подлинности
- `verify_ca` — подключение с использованием TLS и проверка сертификата
- `verify_full` — подключение с использованием TLS, проверка сертификата, а также проверка того, что идентификатор базы данных (CN), указанный в DBHost, соответствует ее сертификату

Конфигурация Zabbix

Веб-интерфейс к базе 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, когда отмечено поле *TLS encryption*, чтобы включить шифрование с использованием сертификатов.

Note:
Для MySQL флажок *Database TLS encryption* отключён, если в поле *Database host* указано localhost, поскольку подключение, использующее файл сокета (в Unix) или общую память (в Windows), не может быть зашифровано.
Для PostgreSQL флажок *TLS encryption* отключён, если значение поля *Database host* начинается со слеша или поле пустое.

Следующие параметры становятся доступны в режиме TLS-шифрования с сертификатами (если установлены оба флажка):

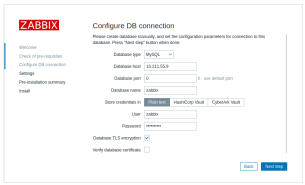
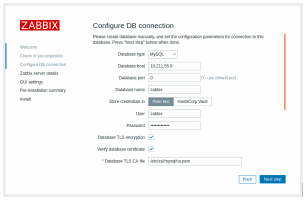
Parameter	Description
Database TLS CA file	Укажите полный путь к действительному файлу центра сертификации (CA) TLS.
Database TLS key file	Укажите полный путь к действительному файлу ключа TLS.
Database TLS certificate file	Укажите полный путь к действительному файлу сертификата TLS.
Database host verification	Установите этот флажок, чтобы активировать проверку узла сети. Отключено для MYSQL, поскольку библиотека PHP MySQL не позволяет пропустить этап проверки сертификата удалённой стороны.
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 приведёт к ошибке авторизации.
Если файлы сертификатов доступны для записи, веб-интерфейс генерирует предупреждение в отчёте System information «TLS certificate files must be read-only.» (отображается только в том случае, если пользователь PHP является владельцем сертификата).

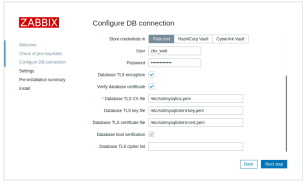
Сертификаты, защищённые паролями, не поддерживаются.

Варианты использования

Веб-интерфейс Zabbix использует графический интерфейс для определения возможных параметров: `required`, `verify_ca`, `verify_full`. Укажите необходимые параметры на шаге мастера установки *Configure DB connections*. Эти параметры сопоставляются с файлом конфигурации (zabbix.conf.php) следующим образом:

Параметры GUI	Файл конфигурации	Описание	Результат
	... // Используется для TLS-соединения. $DB['ENCRYPTION'] = true; $DB['KEY_FILE'] = ''; $DB['CERT_FILE'] = ''; $DB['CA_FILE'] = ''; $DB['VERIFY_HOST'] = false; $DB['CIPHER_LIST'] = ''; ...	Установите флажок *Database TLS encryption* Оставьте *Verify database certificate* неотмеченным	Включает режим `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 *Database TLS encryption* и *Verify database certificate* 2. Укажите путь к *Database TLS CA file*	Включает режим `verify_ca`.

Параметры GUI	Файл конфигурации	Описание	Результат

	...
	// Используется для TLS-соединения со строго заданным списком шифров.
	$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:
	
	...
	// Используется для TLS-соединения без заданного списка шифров — выбирается сервером MySQL
	$DB['ENCRYPTION'] = true;
	$DB['KEY_FILE'] = '<key_file_path>';
	$DB['CERT_FILE'] = '<key_file_path>';
	$DB['CA_FILE'] = '<key_file_path>';
	$DB['VERIFY_HOST'] = true;
	$DB['CIPHER_LIST'] = '';
	...
		1. Установите флажки *Database TLS encryption* и *Verify database certificate*
		2. Укажите путь к *Database TLS key file*
		3. Укажите путь к *Database TLS CA file*
		4. Укажите путь к *Database TLS certificate file*
		5. Укажите *Database TLS cipher list* (необязательно)
			Включает режим `verify_full` для MySQL.

	... $DB['ENCRYPTION'] = true; $DB['KEY_FILE'] = '<key_file_path>'; $DB['CERT_FILE'] = '<key_file_path>'; $DB['CA_FILE'] = '<key_file_path>'; $DB['VERIFY_HOST'] = true; $DB['CIPHER_LIST'] = ' '; ...	1. Установите флажки *Database TLS encryption* и *Verify database certificate* 2. Укажите путь к *Database TLS key file* 3. Укажите путь к *Database TLS CA file* 4. Укажите путь к *Database TLS certificate file* 5. Установите флажок *Database host verification*	Включает режим `verify_full` для PostgreSQL.

См. также: Примеры настройки шифрования для MySQL, Примеры настройки шифрования для PostgreSQL.

Конфигурация сервера/прокси Zabbix

Безопасные подключения к базе данных можно настроить с помощью соответствующих параметров в файле конфигурации Zabbix сервера и/или прокси.

Конфигурация	Результат
Нет	Подключение к базе данных без шифрования.
1. Установите `DBTLSConnect=required`	Сервер/прокси устанавливает TLS-подключение к базе данных. Нешифрованное подключение не допускается.
1. Установите `DBTLSConnect=verify_ca` 2. Установите `DBTLSCAFile` — укажите файл центра сертификации TLS	Сервер/прокси устанавливает TLS-подключение к базе данных после проверки сертификата базы данных.
1. Установите `DBTLSConnect=verify_full` 2. Установите `DBTLSCAFile` — укажите файл центра сертификации TLS	Сервер/прокси устанавливает TLS-подключение к базе данных после проверки сертификата базы данных и подлинности узла сети базы данных.
1. Установите `DBTLSCAFile` — укажите файл центра сертификации TLS 2. Установите `DBTLSCertFile` — укажите файл сертификата открытого ключа клиента 3. Установите `DBTLSKeyFile` — укажите файл закрытого ключа клиента	Сервер/прокси предоставляет сертификат клиента при подключении к базе данных.
1. Установите `DBTLSCipher` — список шифров шифрования, которые клиент разрешает для подключений с использованием протоколов TLS до TLS 1.2 включительно или `DBTLSCipher13` — список шифров шифрования, которые клиент разрешает для подключений с использованием протокола TLS 1.3	(MySQL) TLS-подключение устанавливается с использованием шифра из предоставленного списка. (PostgreSQL) Указание этого параметра будет считаться ошибкой.

1 Настройка шифрования MySQL

Обзор

В этом разделе приведено несколько примеров настройки шифрования для CentOS 8.2 и MySQL 8.4.0, которые можно использовать как краткое руководство по шифрованию подключения к базе данных.

Attention:
Если узел сети MySQL задан как localhost, параметры шифрования будут недоступны. В этом случае соединение между веб-интерфейсом Zabbix и базой данных использует файл сокета (в Unix) или общую память (в Windows) и не может быть зашифровано.

Note:
Список комбинаций шифрования не ограничивается приведёнными на этой странице. Доступно гораздо больше комбинаций.

Предварительные требования

Установите базу данных MySQL из официального репозитория.

Подробные сведения об использовании репозитория MySQL см. в документации MySQL.

Сервер MySQL готов принимать защищённые соединения с использованием самоподписанного сертификата.

Чтобы увидеть, какие пользователи используют зашифрованное соединение, выполните следующий запрос (Performance Schema должна быть включена):

```
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 использованию «из коробки» с режимом шифрования `required`. Сертификат на стороне сервера будет создан после первоначальной настройки и запуска.

Создайте пользователей и роли для основных компонентов:

```
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 подлинности, однако пользователь настроен на использование только зашифрованных соединений. Дополнительные сведения о настройке пользователей см. в документации MySQL.

Выполните команду для проверки соединения (для тестирования защищённых соединений нельзя использовать соединение через сокет):

```
mysql -u zbx_srv -p -h 10.211.55.9 --ssl-mode=REQUIRED
```

Проверьте текущее состояние 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

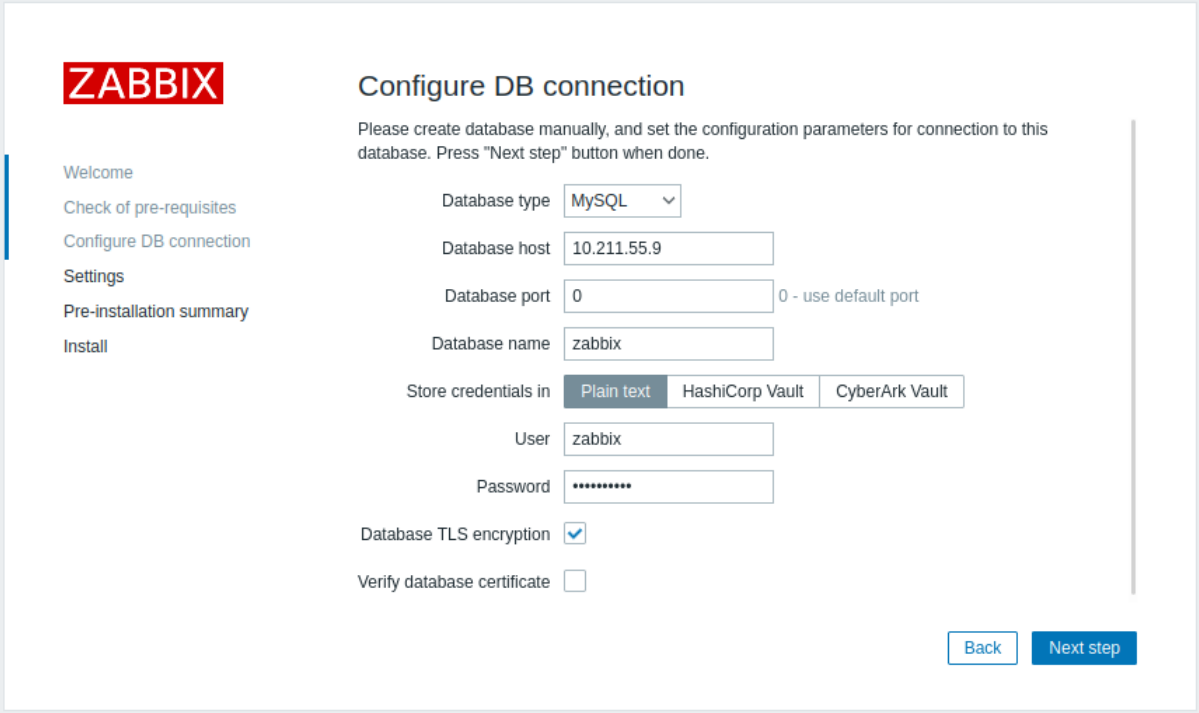

mysql> SHOW SESSION STATUS LIKE 'Ssl_cipher_list'\G;
*************************** 1. row ***************************
Variable_name: Ssl_cipher_list
Value: TLS_AES_256_GCM_SHA384:TLS_CHACHA20_POLY1305_SHA256:TLS_AES_128_GCM_SHA256:TLS_AES_128_CCM_SHA256:E
1 row in set (0.00 sec)

ERROR:
No query specified
```

Веб-интерфейс

Чтобы включить шифрование только транспорта для соединений между веб-интерфейсом Zabbix и базой данных:

- Установите флажок *Database TLS encryption*
- Оставьте *Verify database certificate* неотмеченным



Сервер

Чтобы включить шифрование только на транспортном уровне для соединений между сервером и базой данных, настройте */etc/zabbix/zabbix_server.conf*:

```
...
DBHost=10.211.55.9
DBName=zabbix
DBUser=zbx_srv
DBPassword=<strong_password>
DBTLSConnect=required
...
```

Шифрование с проверкой центра сертификации

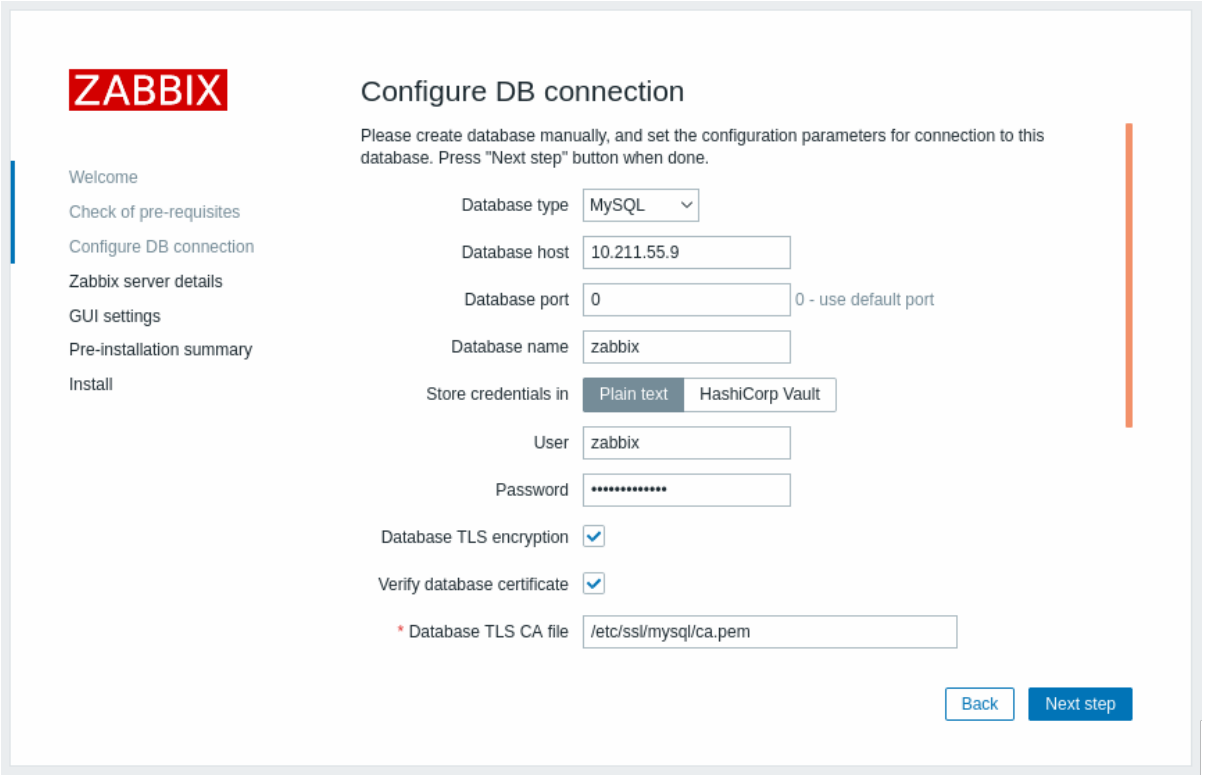
Скопируйте необходимый центр сертификации MySQL на сервер веб-интерфейса Zabbix и назначьте соответствующие права доступа, чтобы веб-сервер мог читать этот файл.

Note:
Этот режим не работает в RHEL 7 из-за более старых библиотек MySQL.

Веб-интерфейс

Чтобы включить шифрование с проверкой сертификата для соединений между веб-интерфейсом Zabbix и базой данных:

- Установите флажки *Database TLS encryption* и *Verify database certificate*
- Укажите путь к файлу Database TLS CA



Либо это можно задать в */etc/zabbix/web/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 пользователь может воспользоваться инструментом командной строки, чтобы проверить, возможно ли подключение для требуемого 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, настройте */etc/zabbix/zabbix_server.conf*:

```
...
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 для параметров конфигурации сервера MySQL CE (*/etc/my.cnf.d/server-tls.cnf*) следующие значения:

```
[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 и клиента MySQL CE (веб-интерфейс Zabbix) необходимо создать вручную в соответствии с документацией MySQL CE: Создание SSL- и RSA-сертификатов и ключей с помощью MySQL или Создание SSL-сертификатов и ключей с помощью openssl

Attention:
Сертификат сервера MySQL должен содержать поле Common Name, установленное в FQDN-имя, 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 в систему 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
```

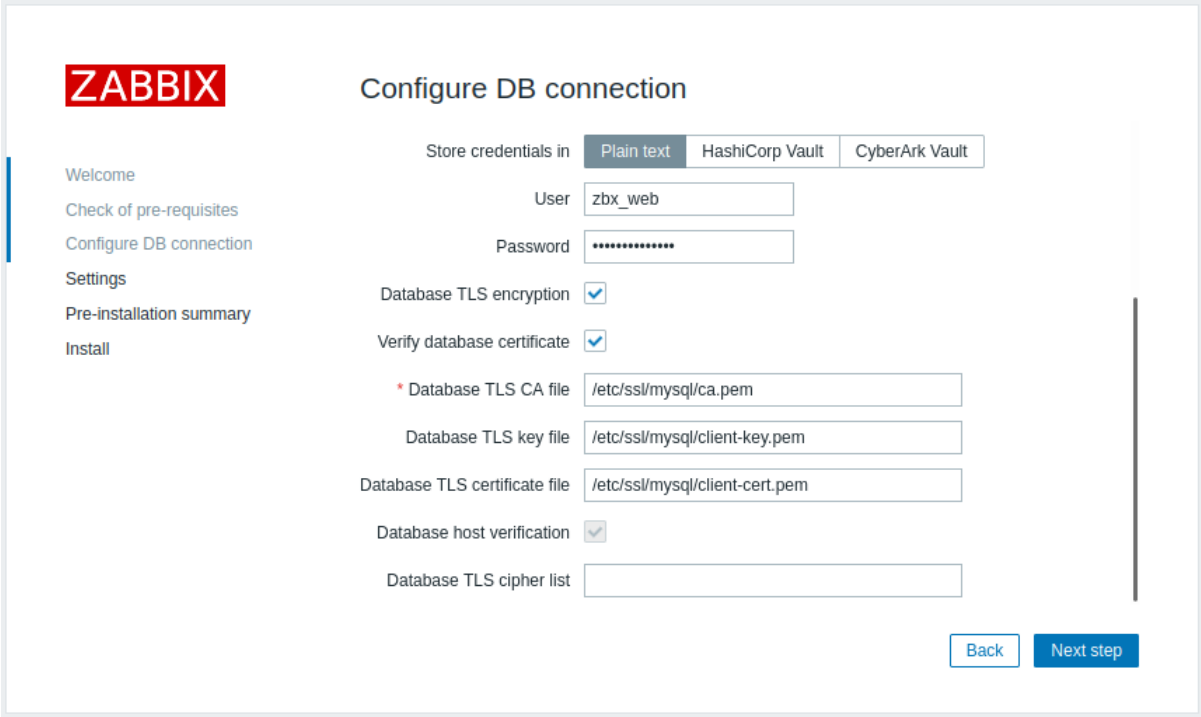
Веб-интерфейс

Чтобы включить шифрование с полной проверкой для соединений между веб-интерфейсом Zabbix и базой данных:

- Установите флажки *Database TLS encryption* и *Verify database certificate*
- Укажите путь к *Database TLS key file*
- Укажите путь к *Database TLS CA file*
- Укажите путь к *Database TLS certificate file*

Обратите внимание, что параметр *Database host verification* установлен и недоступен для изменения — этот шаг нельзя пропустить для MySQL.

Warning:
Если поле *Database TLS cipher list* оставлено пустым, будут включены общие шифры, разрешённые как веб-интерфейсом (клиентом), так и сервером. В качестве альтернативы шифры можно задать явно в соответствии с требованиями к настройке шифров.



В качестве альтернативы это можно задать в */etc/zabbix/web/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, настройте */etc/zabbix/zabbix_server.conf*:

```
...
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
...
```

2 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 слеша или поле пустое.

Предварительные требования

Установите базу данных PostgreSQL, используя официальный репозиторий.

PostgreSQL по умолчанию не настроен на прием TLS-соединений. Пожалуйста, следуйте инструкциям из документации PostgreSQL по подготовке сертификатов с помощью postgresql.conf, а также по управлению доступом пользователей через ph_hba.conf.

По умолчанию сокет PostgreSQL привязан к localhost; чтобы разрешить удаленные сетевые подключения, необходимо включить прослушивание на реальном сетевом интерфейсе.

Настройки 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 управления доступом настройте */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

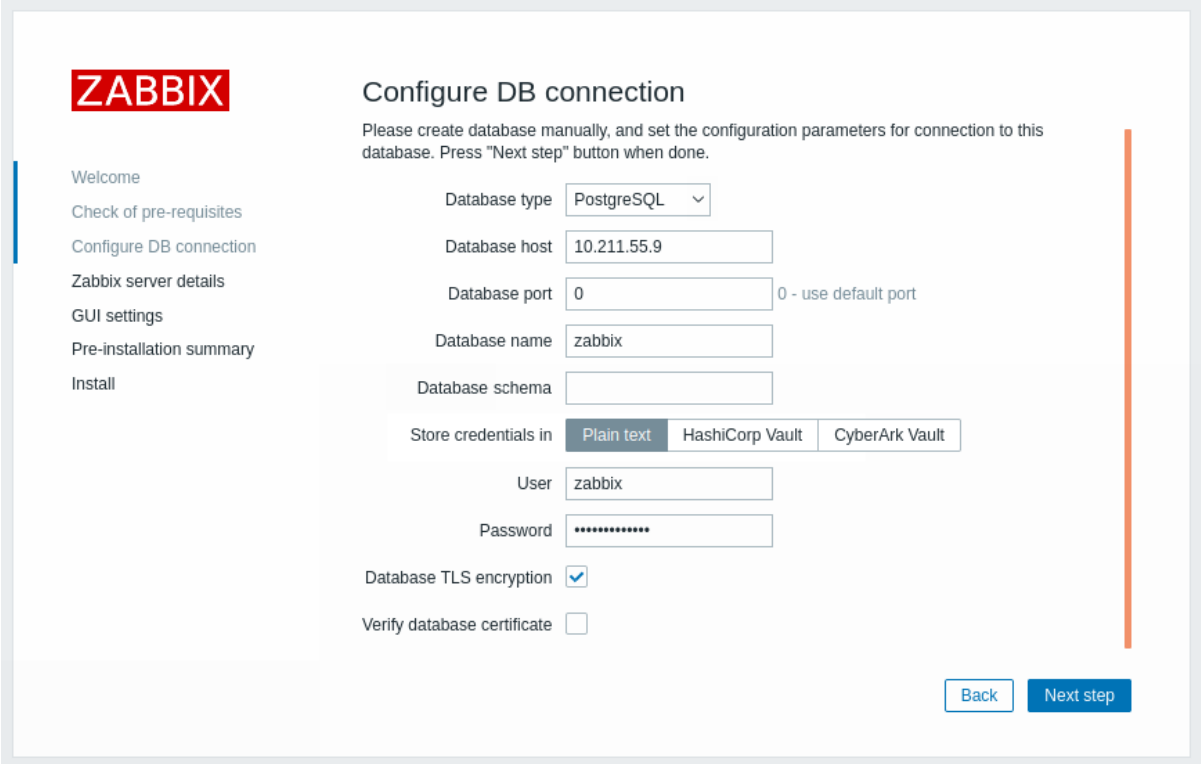
### verify full
hostssl all all 0.0.0.0/0 md5 clientcert=verify-full
...
```

Шифрование только для передачи данных

Веб-интерфейс

Чтобы включить шифрование только транспорта для соединений между веб-интерфейсом Zabbix и базой данных:

- Установите флажок *Database TLS encryption*
- Оставьте *Verify database certificate* неотмеченным



Сервер

Чтобы включить шифрование только на транспортном уровне для соединений между сервером и базой данных, настройте */etc/zabbix/zabbix_server.conf*:

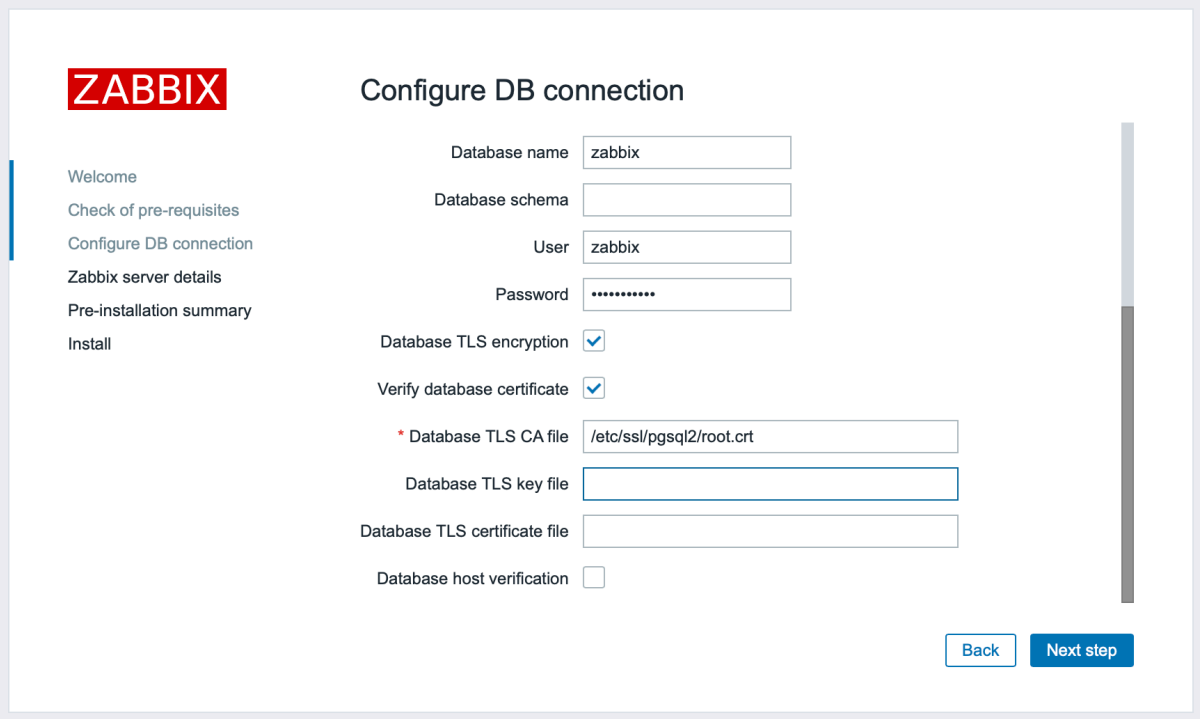
```
...
DBHost=10.211.55.9
DBName=zabbix
DBUser=zbx_srv
DBPassword=<strong_password>
DBTLSConnect=required
...
```

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 *Database TLS encryption* и *Verify database certificate*
- Укажите путь к *Database TLS CA file*



В качестве альтернативы это можно задать в */etc/zabbix/web/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, настройте */etc/zabbix/zabbix_server.conf:*

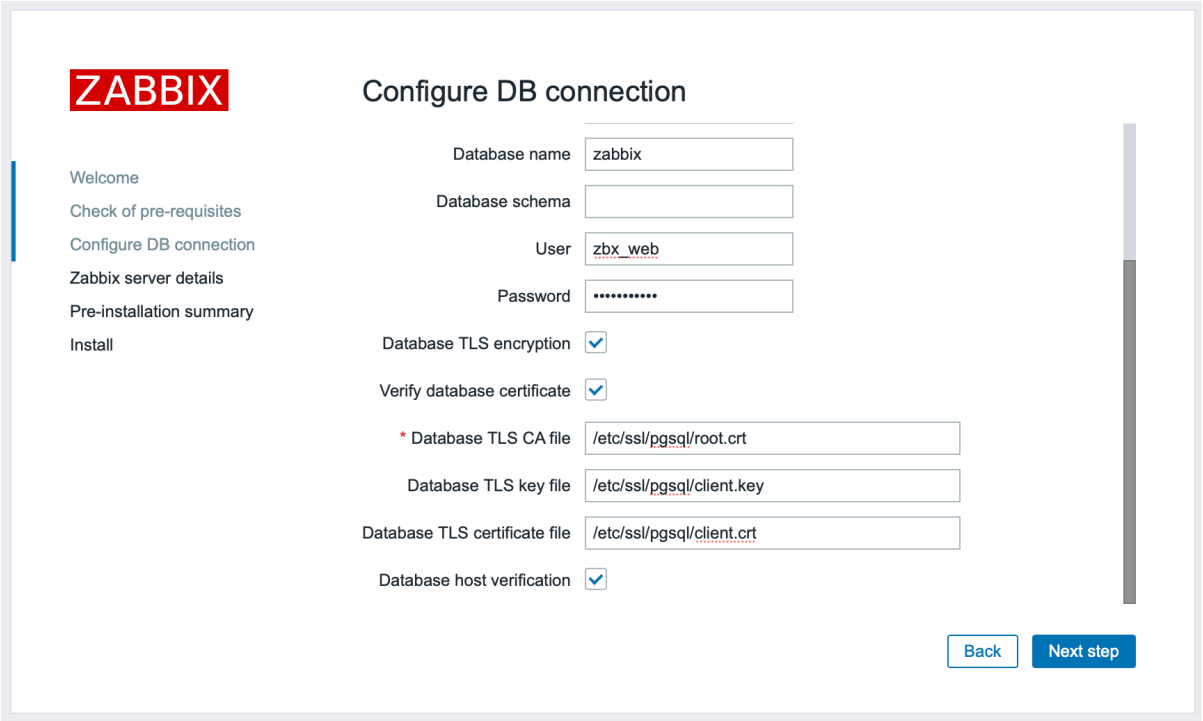
```
...
DBHost=10.211.55.9
DBName=zabbix
DBUser=zbx_srv
DBPassword=<strong_password>
DBTLSConnect=verify_ca
DBTLSCAFile=/etc/ssl/pgsql/root.crt
...
```

Шифрование с полной проверкой

Веб-интерфейс

Чтобы включить шифрование с использованием сертификата и проверку подлинности узла сети базы данных для соединений между веб-интерфейсом Zabbix и базой данных:

- Установите флажки *Database TLS encryption* и *Verify database certificate*
- Укажите путь к *Database TLS key file*
- Укажите путь к *Database TLS CA file*
- Укажите путь к *Database TLS certificate file*
- Установите флажок *Database host verification*



В качестве альтернативы это можно задать в */etc/zabbix/web/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 использованием сертификата и проверку подлинности узла сети базы данных для соединений между сервером Zabbix и базой данных, настройте */etc/zabbix/zabbix_server.conf*:

```
...
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
...
```

6 Безопасное подключение к веб-интерфейсу

Обзор

В этом разделе приведены шаги по настройке Zabbix и примеры конфигурации для защищённых 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 на обеих сторонах. Ниже приведен пример самого простого способа сделать это.

1. Сгенерируйте 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 common name):

```
sudo openssl genrsa -out ca.key 4096
sudo openssl req -new -x509 -days 3650 -key ca.key -out ca.crt -subj "/CN=MyZabbixCA/"
```

Сгенерируйте закрытый ключ и сертификат для сервера Zabbix (измените значение `zabbix-server.example.com` в соответствии с фактическим common name):

```
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
```

Сгенерируйте закрытый ключ и сертификат для веб-интерфейса Zabbix (измените значение `zabbix-frontend-node` в соответствии с фактическим common name):

```
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 правильные разрешения.

Для сервера Zabbix (измените владельца/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/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 `zabbix_server.conf` добавьте:

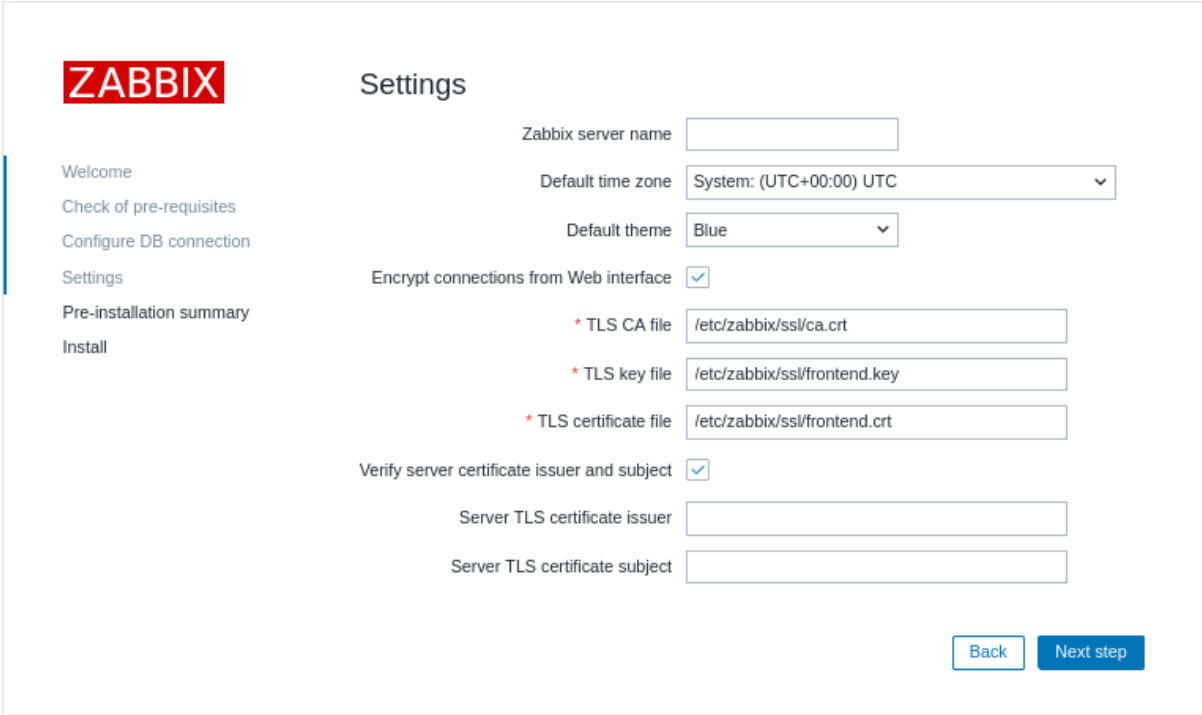
```
TLSFrontendAccept=cert
TLSCertFile=/etc/zabbix/ssl/server.crt
TLSKeyFile=/etc/zabbix/ssl/server.key
TLSCAFile=/etc/zabbix/ssl/ca.crt
#### Optionally:
#### TLSFrontendCertIssuer=/CN=MyZabbixCA/
#### TLSFrontendCertSubject=/CN=zabbix-frontend-node/
```

Затем перезапустите сервер:

```
sudo systemctl restart zabbix-server
```

4. Настройте веб-интерфейс Zabbix.

Во время установки веб-интерфейса включите опцию *Encrypt connections from Web interface* (а также опцию *Verify server certificate issuer and subject*, если необходимо) и заполните поля *TLS CA file*, *TLS key file*, *TLS certificate file* (а также *Server TLS certificate issuer* и *Server TLS certificate subject*, если необходимо):



Параметр	Описание
TLS CA file	Укажите полный путь к файлу сертификата центра сертификации (CA), используемому для проверки сертификата сервера.
TLS key file	Укажите полный путь к файлу закрытого ключа клиента, соответствующего сертификату клиента.
TLS certificate file	Укажите полный путь к файлу сертификата клиента, если требуется взаимная TLS-аутентификация.
Server TLS certificate issuer	Укажите distinguished Name (DN) издателя для сопоставления с сертификатом сервера.
Server TLS certificate subject	Укажите distinguished Name (DN) субъекта для сопоставления с сертификатом сервера.

В существующих установках отредактируйте следующие поля в `zabbix.conf.php`:

```
$ZBX_SERVER_TLS['ACTIVE'] = '1';
$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';
// Optionally:
// $ZBX_SERVER_TLS['CERTIFICATE_ISSUER']  = '/CN=MyZabbixCA/';
// $ZBX_SERVER_TLS['CERTIFICATE_SUBJECT'] = '/CN=zabbix-server.example.com/';
```

5. Проверьте шифрование, убедившись, что в веб-интерфейсе Zabbix или в файле журнала сервера Zabbix отсутствуют сообщения об ошибках:

```
tail -f /var/log/zabbix/zabbix_server.log
```

7 Настройка TimescaleDB

Обзор

Zabbix поддерживает TimescaleDB — решение для баз данных на основе PostgreSQL, автоматически разделяющее данные на временные фрагменты для обеспечения более высокой производительности при масштабировании.

Warning:
В настоящее время TimescaleDB не поддерживается прокси Zabbix.

Инструкции на этой странице можно использовать в следующих сценариях:

- Создание базы данных TimescaleDB или миграция существующих таблиц PostgreSQL в TimescaleDB (см. Настройка).
- Обновление схемы существующей базы данных TimescaleDB при обновлении Zabbix (см. 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 поддерживаемую версию 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 предложение SCHEMA в приведённую выше команду. Например:
> `echo "CREATE EXTENSION IF NOT EXISTS timescaledb SCHEMA yourschema CASCADE;" | sudo -u postgres psql zabbix`

Затем запустите скрипт `postgresql/timescaledb/schema.sql`. Для новых установок скрипт необходимо запускать после создания обычной базы данных PostgreSQL с исходной схемой/данными (см. создание базы данных).

```
cat /usr/share/zabbix/sql-scripts/postgresql/timescaledb/schema.sql | sudo -u zabbix psql zabbix
```

> **Attention:**
> Пожалуйста, игнорируйте предупреждающие сообщения о несоблюдении рекомендаций по лучшим практикам при выполнении скрипта `schema.sql` на TimescaleDB версии 2.9.0 и выше. Несмотря на это предупреждение, настройка будет успешно завершена.

Миграция существующих данных истории, трендов и журнала аудита может занять много времени. Сервер Zabbix и веб-интерфейс должны быть остановлены на время миграции.

Скрипт `schema.sql` устанавливает следующие параметры очистки данных:

- Переопределить период хранения истории элемента данных
- Переопределить период хранения трендов элемента данных

Чтобы использовать секционированную очистку данных для истории и трендов, обе эти опции должны быть включены. Также можно включить переопределение отдельно только для истории или только для трендов.

Для PostgreSQL и TimescaleDB скрипт `postgresql/timescaledb/schema.sql` устанавливает два дополнительных параметра:

- Включить сжатие
- Сжимать записи старше 7 дней

Чтобы housekeeper успешно удалял сжатые данные, должны быть включены оба параметра: *Переопределить период хранения истории элементов данных* и *Переопределить период хранения трендов элементов данных*. Если переопределение отключено, а таблицы содержат сжатые чанки, housekeeper не будет удалять данные из этих таблиц, и в разделах *Очистка истории* и *Информация о системе* будут отображаться предупреждения о некорректной конфигурации.

Все эти параметры можно изменить в разделе *Администрирование* > *Очистка истории* после установки.

> **Note:**
> Для оптимизации параметров конфигурации PostgreSQL в вашем `postgresql.conf` вы можете использовать инструмент timescaledb-tune, предоставляемый TimescaleDB.

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

Встроенное 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 Timescale Community ”timescale” и не поддерживается по лицензии Apache 2.0 ”apache”. Если Zabbix обнаружит, что сжатие не поддерживается, в журнал сервера Zabbix будет записано предупреждающее сообщение, и пользователи не смогут включить сжатие в веб-интерфейсе.

Note:
Перед использованием сжатия пользователям рекомендуется ознакомиться с возможностями сжатия в документации TimescaleDB.

Обратите внимание, что сжатие накладывает определенные ограничения, а именно:

- Изменение сжатых chunk’ов (insert, delete, update) не допускается
- Изменение схемы для сжатых таблиц не допускается.

Параметры сжатия можно изменить в блоке *History, trends and audit log compression* в разделе *Administration* > *Housekeeping* веб-интерфейса Zabbix.

Parameter	Default	Comments
Enable compression	Enabled	Установка или снятие этого флажка не активирует/деактивирует сжатие немедленно. Поскольку сжатием управляет Housekeeper, изменения вступят в силу в течение не более 2 × `HousekeepingFrequency` часов (задается в zabbix_server.conf) После отключения сжатия новые chunk’и, попадающие в период сжатия, сжиматься не будут. Однако все ранее сжатые данные останутся сжатыми. Чтобы распаковать ранее сжатые chunk’и, следуйте инструкциям в документации TimescaleDB.
Compress records older than	7d	При обновлении со старых версий Zabbix с поддержкой TimescaleDB сжатие по умолчанию включено не будет. Значение этого параметра не может быть меньше 7 дней. Из-за неизменяемости сжатых chunk’ов все поздно поступившие данные (например, данные, задержанные прокси), которые старше этого значения, будут отброшены.

Для повышения производительности обновления трендов можно 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 — следовать рекомендациям TimescaleDB и гарантировать, что размер chunk’ов остается в пределах доступных ресурсов системы.

8 Настройка Elasticsearch

Zabbix может хранить исторические данные в Elasticsearch в качестве альтернативы реляционной базе данных.

> **Attention:**
> Поддержка Elasticsearch в настоящее время является экспериментальной.

Это руководство описывает настройку для Elasticsearch 7.X. Если вы используете другую версию, часть функциональности может работать не так, как ожидается.

Настройка включает создание хранилища Elasticsearch для каждого типа значений, настройку предобработки (при необходимости) и подключение Zabbix к вашему экземпляру Elasticsearch.

Elasticsearch может хранить следующие типы значений:

Тип значения элемента данных	Таблица базы данных	Тип Elasticsearch
Числовой (без знака)	history_uint	uint
Числовой (с плавающей точкой)	history	dbl
Символьный	history_str	str
Журнал	history_log	log
Текст	history_text	text
Двоичный	history_bin	не поддерживается Zabbix
JSON	history_json	json

Важные примечания

- Elasticsearch требует libcurl. Подробности см. в требованиях.
- housekeeper не удаляет данные из Elasticsearch.
- Если все исторические данные хранятся в Elasticsearch, тренды **не** вычисляются и не сохраняются в базе данных. Рассмотрите возможность увеличения периода хранения истории.
- При использовании Elasticsearch запросы диапазона, извлекающие значения из базы данных, ограничиваются временной меткой периода хранения данных.
- Elasticsearch не поддерживается для Zabbix прокси; вместо него используйте SQLite.

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 исторических данных 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 их хранения с течением времени.

Чтобы включить это, необходимо:

- Создать шаблон индекса для каждого типа значения, который вы хотите хранить — это указывает Elasticsearch, какое сопоставление применять при автоматическом создании нового индекса на основе даты.
- Создать конвейер обработки для каждого типа значения — он обрабатывает каждое входящее значение и направляет его в правильный индекс на основе даты.
- Настроить параметр `HistoryStorageDateIndex` в файле конфигурации сервера Zabbix — это включает хранение значений в нескольких индексах 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 вашего сервера Zabbix (`zabbix_server.conf`) задайте следующие параметры:

- `HistoryStorageURL` — URL вашего экземпляра Elasticsearch.
- `HistoryStorageTypes` — разделённый запятыми список типов значений для хранения в Elasticsearch.

Например, чтобы хранить в Elasticsearch значения типов *Character*, *Log*, *Text* и *JSON* (при этом сохраняя значения типа *Numeric* в базе данных):

```
HistoryStorageURL=http://localhost:9200
HistoryStorageTypes=str,log,text,json
```

Если вы используете индексы на основе даты для **всех** значений, хранящихся в Elasticsearch, также задайте параметр `HistoryStorageDateIndex`:

```
HistoryStorageDateIndex=1
```

После внесения изменений перезапустите сервер Zabbix:

```
systemctl restart zabbix-server
```

Настройка веб-интерфейса Zabbix

В файле конфигурации веб-интерфейса Zabbix (`zabbix.conf.php`) объявите `$HISTORY` как глобальную переменную и задайте значения `url` и `types` в соответствии с конфигурацией сервера:

```
//                          Zabbix.
global $DB, $HISTORY;

$HISTORY['url']   = 'http://localhost:9200';
$HISTORY['types'] = ['str', 'log', 'text', 'json'];
```

Устранение неполадок

Следующие шаги могут помочь вам устранить проблемы с настройкой Elasticsearch:

1. 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": {} }
```

2. Убедитесь, что mappings, шаблоны и ingest pipelines настроены правильно, отправив запросы GET на соответствующие URL-адреса:

```
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.

3. Проверьте, нет ли shards в состоянии сбоя; перезапуск Elasticsearch может решить эту проблему.
4. Убедитесь, что ваша конфигурация Elasticsearch разрешает доступ с сервера Zabbix и веб-интерфейса Zabbix.
5. Используйте параметр конфигурации сервера Zabbix LogSlowQueries для выявления медленных запросов.
6. Проверьте журналы Elasticsearch на наличие ошибок.
7. 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/*"
```

9 Примечания по настройке Nginx для Zabbix в зависимости от дистрибутива

RHEL

В RHEL 8, 9 и 10 Nginx включён в AppStream — дополнительные репозитории не требуются. Просто включите и установите его с помощью:

```
dnf module list nginx
dnf -y install nginx
```

В RHEL 7 Nginx можно установить из репозитория CentOS 7 Extras (если используется CentOS) с помощью:

```
yum -y install nginx
```

Если вы предпочитаете самые последние upstream-сборки, настройте официальный репозиторий NGINX в /etc/yum.repos.d/nginx

```
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 service pack):

```
cp /etc/php8/fpm/php-fpm.conf{.default,}
cp /etc/php8/fpm/php-fpm.d/www.conf{.default,}
sed -i 's/user = nobody/user = wwwrun/; s/group = nobody/group = www/' /etc/php8/fpm/php-fpm.d/www.conf
```

10 Запуск агента от имени root

Начиная с Zabbix **5.0.0**, файл службы systemd для агента Zabbix в официальных пакетах явно включает директивы `User` и `Group`. Обе имеют значение `zabbix`.

Больше 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 описанные ниже изменения.

Zabbix агент

Чтобы переопределить пользователя и группу по умолчанию для Zabbix агент, выполните:

```
systemctl edit zabbix-agent
```

Затем добавьте следующее содержимое:

```
[Service]
User=root
Group=root
```

Перезагрузите демоны и перезапустите службу zabbix-agent:

```
systemctl daemon-reload
systemctl restart zabbix-agent
```

Для **Zabbix агент** это повторно включает возможность настройки пользователя в файле `zabbix_agentd.conf`. Теперь необходимо задать параметры конфигурации `User=root` и `AllowRoot=1` в агент configuration file.

Zabbix агент 2

Чтобы переопределить пользователя и группу по умолчанию для Zabbix агент 2, выполните:

```
systemctl edit zabbix-agent2
```

Затем добавьте следующее содержимое:

```
[Service]
User=root
Group=root
```

Перезагрузите демоны и перезапустите сервис zabbix-agent2:

```
systemctl daemon-reload
systemctl restart zabbix-agent2
```

Для **Zabbix agent2** это полностью определяет пользователя, от имени которого он запускается. Никаких дополнительных изменений не требуется.

11 Агент Zabbix в Microsoft Windows

Настройка агента

Оба поколения агентов Zabbix работают как служба Windows. Для Zabbix agent 2 замените *agentd* на *agent2* в приведенных ниже инструкциях.

Вы можете запускать один экземпляр агента Zabbix или несколько экземпляров агента на узле сети Microsoft Windows. Один экземпляр может использовать либо:

- файл конфигурации по умолчанию, расположенный в том же каталоге, что и исполняемый файл агента;
- файл конфигурации, указанный в командной строке.

В случае нескольких экземпляров каждый экземпляр агента должен иметь свой собственный файл конфигурации (один из экземпляров может использовать файл конфигурации по умолчанию).

Пример файла конфигурации доступен в архиве исходного кода Zabbix как:

- `conf/zabbix_agentd.conf` для Zabbix agent;
- `conf/zabbix_agent2.conf` для Zabbix agent2.

Если вы хотите установить Zabbix agent/agent 2 для Windows как службу из архива без явного указания файла конфигурации, то перед установкой агента:

- `conf/zabbix_agentd.conf` следует вручную скопировать в каталог, где будет установлен zabbix_agentd.exe;
- `conf/zabbix_agent2.conf` и каталог `conf/zabbix_agent2.d` следует вручную скопировать в каталог, где будет установлен zabbix_agent2.exe.

Подробные сведения о настройке агента Zabbix для Windows см. в разделе файл конфигурации.

Параметр Hostname

Чтобы выполнять активные проверки на узле сети, агент Zabbix должен иметь заданное имя узла сети. Более того, значение имени узла сети, заданное на стороне агента, должно в точности совпадать с "Имя узла сети", настроенным для этого узла сети в веб-интерфейсе.

Значение имени узла сети на стороне агента может быть задано с помощью параметра **Hostname** или **HostnameItem** в файле конфигурации агента — либо будут использованы значения по умолчанию, если какой-либо 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 опрашивает поставщиков пространств имен для определения имени локального 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 этих 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 от 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 записанное только 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, будет возвращена 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 три различных подхода:

1. Оставить параметры **Hostname** или **HostnameItem** неопределенными, и агент Zabbix будет использовать имя узла сети NetBIOS в качестве имени узла сети.

2. Оставить параметр **Hostname** неопределенным и задать **HostnameItem** следующим образом:
   HostnameItem=system.hostname[host] — чтобы агент Zabbix использовал полное фактическое имя узла сети Windows (с учетом регистра) в качестве имени узла сети
   HostnameItem=system.hostname[shorthost,lower] — чтобы агент Zabbix использовал только часть имени узла сети до первой точки, преобразованную в нижний регистр.
   HostnameItem=system.hostname[fqdn] — чтобы агент Zabbix использовал полное доменное имя 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` то агент будет установлен как служба Windows "`Zabbix Agent [Windows_db_server]`". Поэтому, чтобы каждая копия агента Zabbix имела собственное имя службы Windows, каждая копия должна использовать различное 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 Zabbix агент с файлом конфигурации по умолчанию:

```
zabbix_agentd.exe --install
```

Attention:
В 64-разрядной системе для корректной работы всех проверок, связанных с запущенными 64-разрядными процессами, требуется 64-разрядная версия Zabbix агента.

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 Zabbix агента 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, вы можете использовать Панель управления или сделать это из командной строки.

Чтобы запустить один экземпляр Zabbix агента с файлом конфигурации по умолчанию:

```
zabbix_agentd.exe --start
```

Чтобы запустить один экземпляр Zabbix агента с другим файлом конфигурации:

```
zabbix_agentd.exe --config <your_configuration_file> --start
```

Чтобы запустить один из нескольких экземпляров Zabbix агента:

```
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 Zabbix агента, запущенный с файлом конфигурации по умолчанию:

```
zabbix_agentd.exe --stop
```

Чтобы остановить один экземпляр Zabbix агента, запущенный с другим файлом конфигурации:

```
zabbix_agentd.exe --config <your_configuration_file> --stop
```

Чтобы остановить один из нескольких экземпляров Zabbix агента:

```
zabbix_agentd.exe --config <configuration_file_for_this_instance> --stop --multiple-agents
```

Удаление службы Windows агента

Чтобы удалить один экземпляр агента Zabbix, используя файл конфигурации по умолчанию:

```
zabbix_agentd.exe --uninstall
```

Чтобы удалить один экземпляр агента Zabbix, используя файл конфигурации не 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 по NUMA-узлам неравномерно. Если логические процессоры распределены неравномерно, то метрики производительности ЦП могут быть недоступны для некоторых процессоров. Например, если имеется 72 логических процессора и 2 NUMA-узла, то на каждом узле должно быть по 36 процессоров.

12 Настройка SAML с Microsoft Entra ID

Обзор

В этом разделе приведены рекомендации по настройке единого входа и автоматического предоставления учетных записей пользователей в Zabbix из Microsoft Entra ID (ранее Microsoft Azure Active Directory) с использованием аутентификации SAML 2.0.

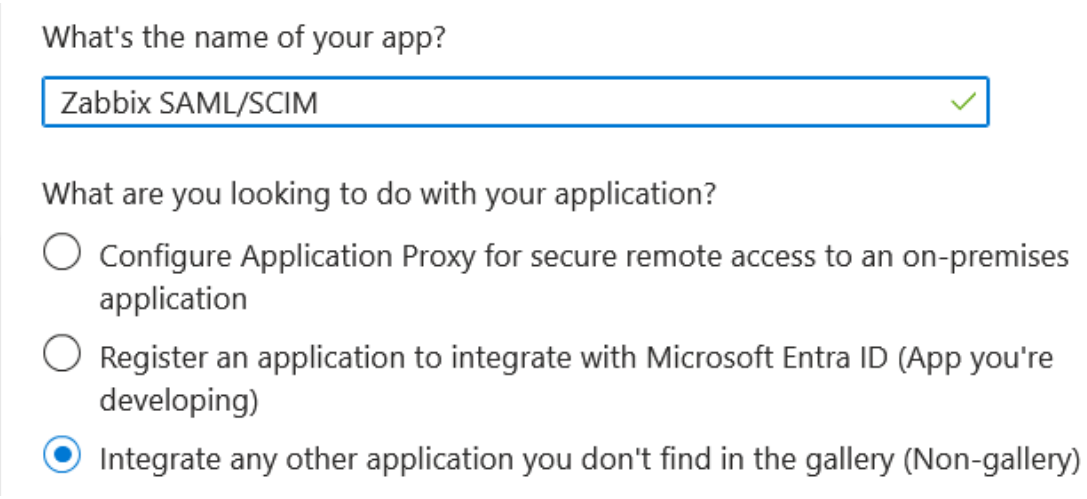
Настройка Microsoft Entra ID

Создание приложения

1. Войдите в центр администрирования Microsoft Entra по адресу 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. Укажите имя вашего приложения и выберите вариант *Integrate any other application...*. После этого нажмите *Create*.

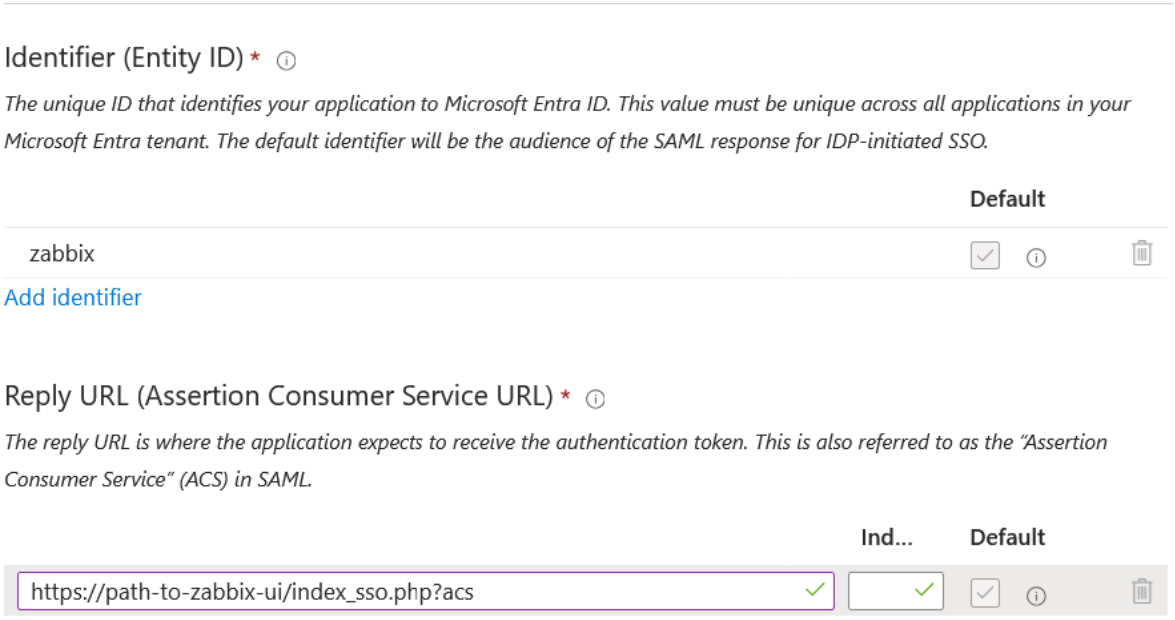


Настройка единого входа

1. На странице вашего приложения перейдите в раздел *Set up single sign on* и нажмите *Get started*. Затем выберите *SAML*.

2. Отредактируйте *Basic SAML Configuration*:

- В поле *Identifier (Entity ID)* задайте уникальное имя для идентификации вашего 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-`



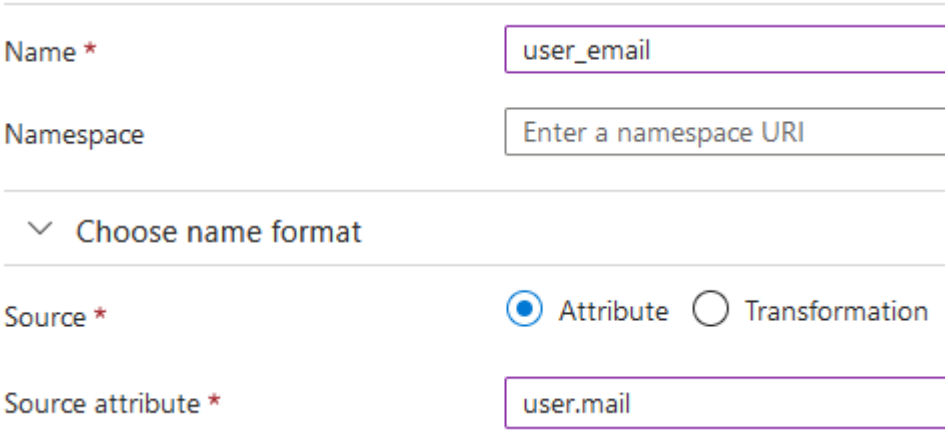
Обратите внимание, что требуется использовать "https". Чтобы это работало с Zabbix, необходимо добавить в `conf/zabbix.conf.php` следующую строку:

```
$SSO['SETTINGS'] = ['use_proxy_headers' => true];
```

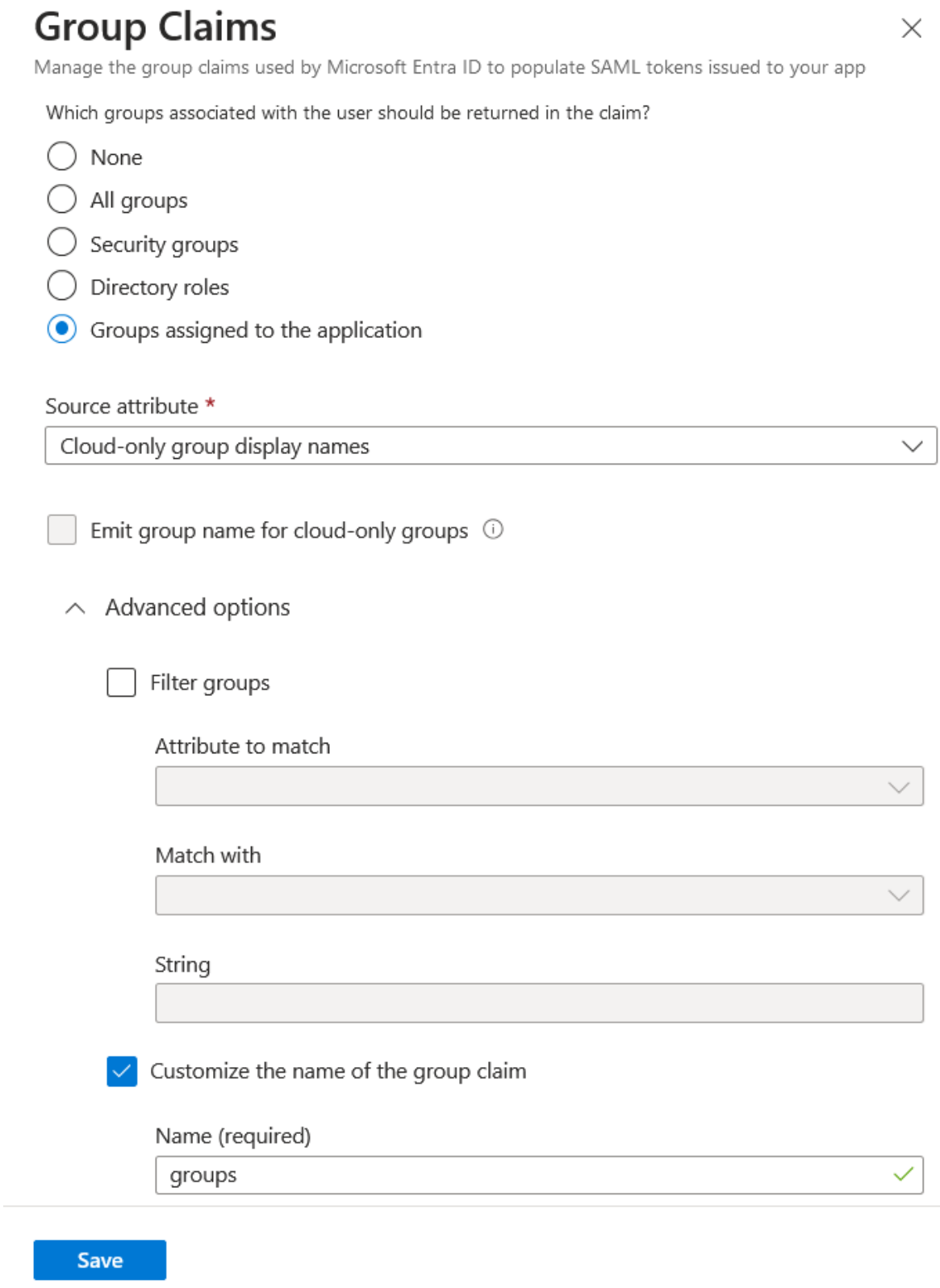
3. Отредактируйте *Attributes & Claims*. Необходимо добавить все атрибуты, которые вы хотите передавать в Zabbix (user_name, user_lastname, user_email, user_mobile, groups).

Имена атрибутов могут быть произвольными. Можно использовать разные имена атрибутов, однако требуется, чтобы они соответствовали значениям соответствующих полей в настройках Zabbix SAML.

- Нажмите *Add new claim*, чтобы добавить атрибут:



- Нажмите *Add a group claim*, чтобы добавить атрибут для передачи групп в Zabbix:



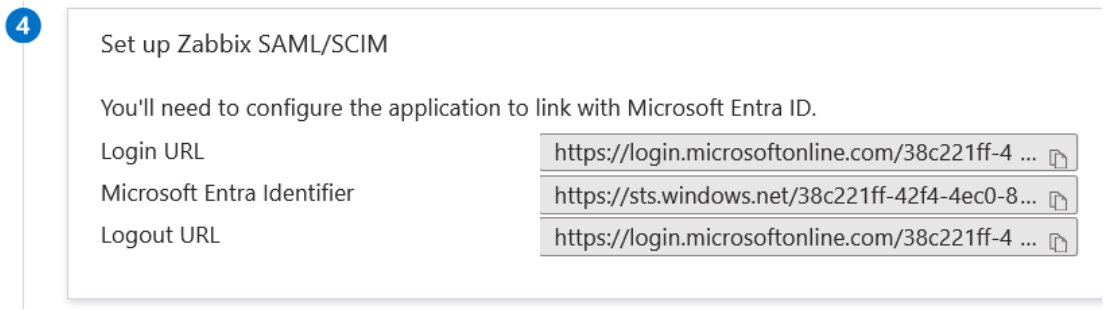
Важно, чтобы в этом утверждении в Zabbix передавались имена групп (а не идентификаторы групп) через выбранный *Source attribute*. В противном случае JIT-подготовка пользователей не будет работать должным образом.

4. В разделе *SAML Certificates* скачайте сертификат Base64, предоставленный Entra ID, и поместите его в `conf/certs` установки веб-интерфейса Zabbix.

Установите для него права 644, выполнив:

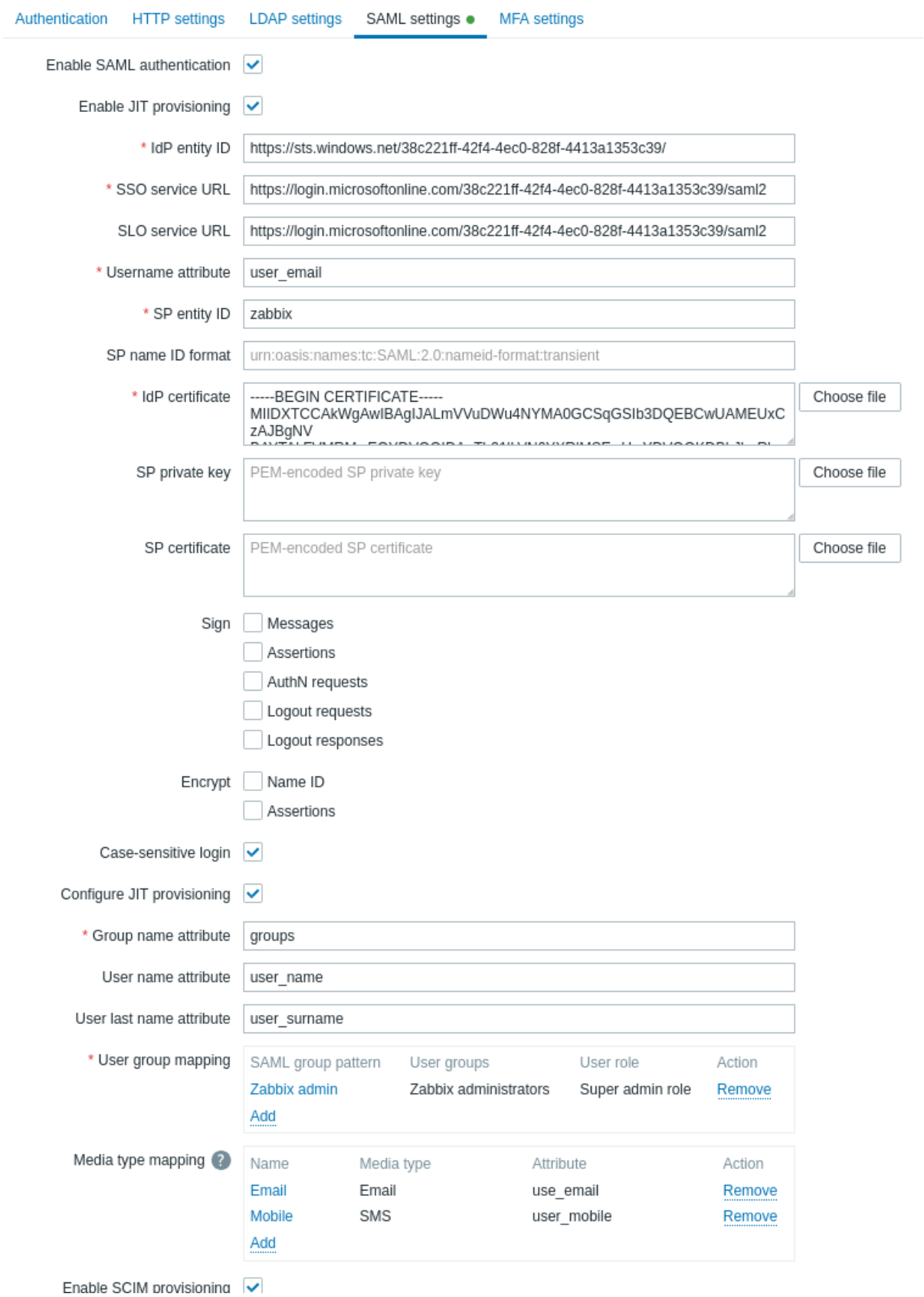
```
chmod 644 entra.cer
```

5. Используйте значения из раздела *Set up <your app name>* в Entra ID для настройки SAML-аутентификации Zabbix (см. следующий раздел):



Конфигурация Zabbix

1. В Zabbix перейдите в раздел настроек SAML и заполните параметры конфигурации на основе конфигурации Entra ID:



Поле Zabbix	Поле настройки в Entra ID	Пример значения
ID сущности IdP	Идентификатор Microsoft Entra	
URL сервиса SSO	URL входа	
URL сервиса SLO	URL выхода	
ID сущности SP	Идентификатор (Entity ID)	
Атрибут имени пользователя	Пользовательский атрибут (claim)	`user_email`
Атрибут имени группы	Пользовательский атрибут (claim)	`groups`
Атрибут имени	Пользовательский атрибут (claim)	`user_name`

Поле Zabbix	Поле настройки в Entra ID	Пример значения
Атрибут фамилии пользователя	Пользовательский атрибут (claim)	`user_lastname`

2. Добавьте сертификат Base64, предоставленный Entra ID.

Если в zabbix.conf.php задано `$SSO['CERT_STORAGE'] = 'database'`, вы можете вставить текст сертификата или загрузить файл сертификата в веб-интерфейс во время настройки SAML — файлы в файловой системе не требуются.

Если в zabbix.conf.php задано `$SSO['CERT_STORAGE'] = 'file'`, сертификат должен быть доступен в файловой системе (по умолчанию в ui/conf/certs или по пути, настроенному в zabbix.conf.php), и веб-интерфейс не будет хранить сертификаты в базе данных.
Обратите внимание, что если `$SSO['CERT_STORAGE']` не задан или закомментирован, предполагается файловое хранилище, и сертификаты считываются из ui/conf/certs.

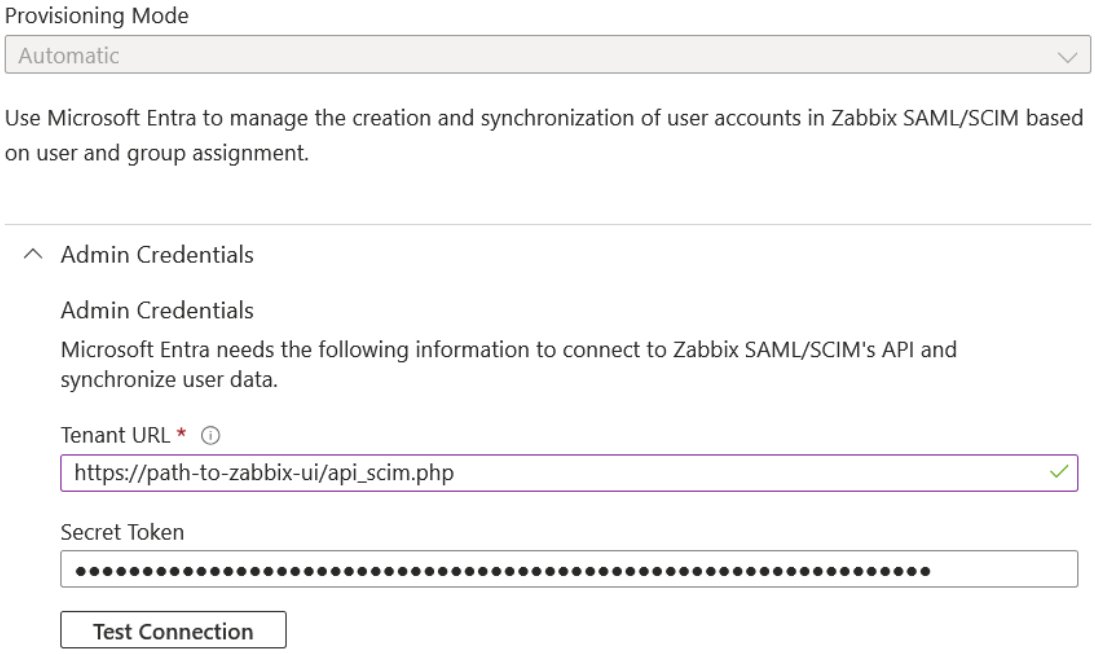
Также необходимо настроить сопоставление групп пользователей. Сопоставление медиа является необязательным.

3. Нажмите кнопку *Update*, чтобы сохранить эти настройки.

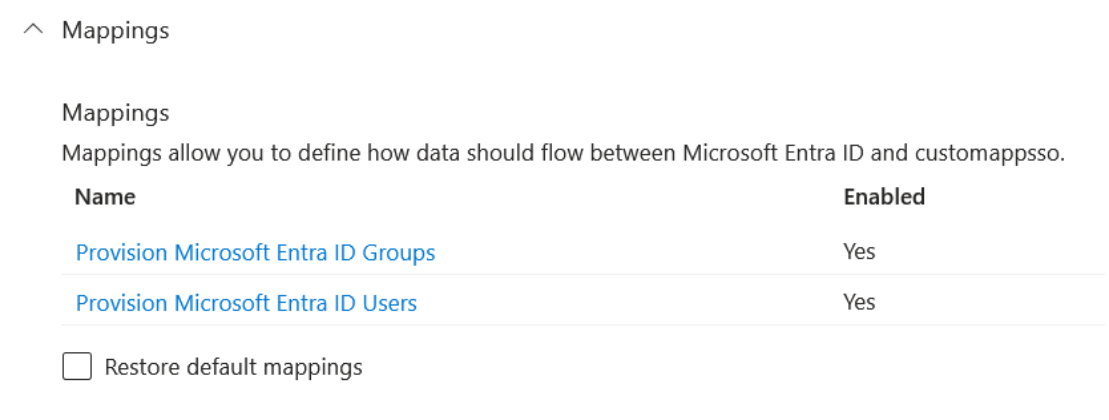
Подготовка пользователей SCIM

1. На странице вашего приложения Entra ID в главном меню откройте страницу Provisioning. Нажмите *Get started*, а затем выберите режим автоматической подготовки:

- В поле *Tenant URL* укажите следующее значение: `https://<path-to-zabbix-ui>/api_scim.php`
- В поле *Secret token* введите токен API Zabbix с правами Super admin.
- Нажмите *Test connection*, чтобы проверить, установлено ли соединение.

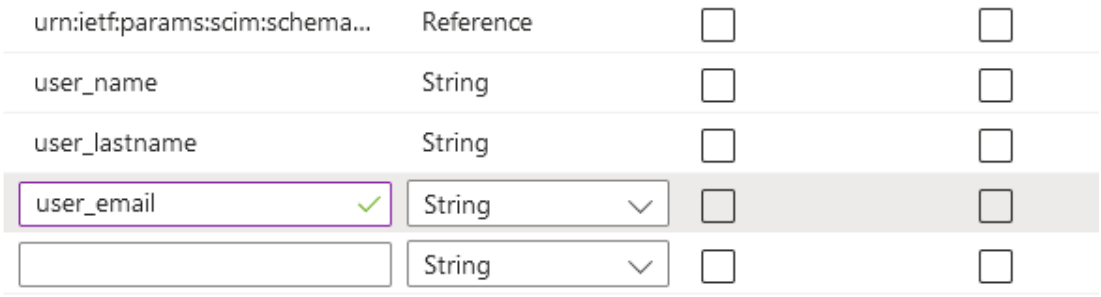


2. Теперь вы можете добавить все атрибуты, которые будут передаваться через SCIM в Zabbix. Для этого нажмите *Mappings*, а затем *Provision Microsoft Entra ID Users*.



В нижней части списка Attribute Mapping включите *Show advanced options*, затем нажмите *Edit attribute list for customappsso*.

В нижней части списка атрибутов добавьте собственные атрибуты с типом 'String':



Сохраните список.

3. Теперь вы можете добавить сопоставления для добавленных атрибутов. В нижней части списка Attribute Mapping нажмите *Add New Mapping* и создайте сопоставления, как показано ниже:



Когда все сопоставления будут добавлены, 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. Когда пользователи и группы будут созданы в Entra ID, вы можете перейти в меню *Users and groups* вашего приложения и добавить их в приложение.

6. Перейдите в меню *Provisioning* вашего приложения и нажмите *Start provisioning*, чтобы подготовить пользователей в Zabbix.

Обратите внимание, что запрос PATCH для Users в Entra ID не поддерживает изменения в способах оповещения.

Подпись запросов аутентификации

Можно настроить Entra ID на проверку подписи подписанных запросов аутентификации.

Чтобы это работало, создайте открытый/закрытый ключи:

```
openssl req -x509 -newkey rsa:4096 -keyout /usr/share/zabbix/conf/certs/request-sign.key -out /usr/share/z
```

Назначьте права доступа:

```
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

Проблемы с аутентификацией могут возникать в браузерах MicroSoft Edge, когда пользователь, пытающийся войти в Zabbix через SAML, уже вошел в систему с использованием профиля MicroSoft Edge. Признаком такой проблемы может быть то, что пользователь может войти в Zabbix с помощью MicroSoft Edge в режиме InPrivate.

Чтобы избежать проблем с аутентификацией в этом случае, может потребоваться установить `requestedAuthnContext` в значение "false" в файле конфигурации веб-интерфейса Zabbix (*zabbix.conf.php*).

```
$SSO['SETTINGS'] = [
    'security' => [
        'requestedAuthnContext' => false
    ]
];
```

13 Настройка SAML с Okta

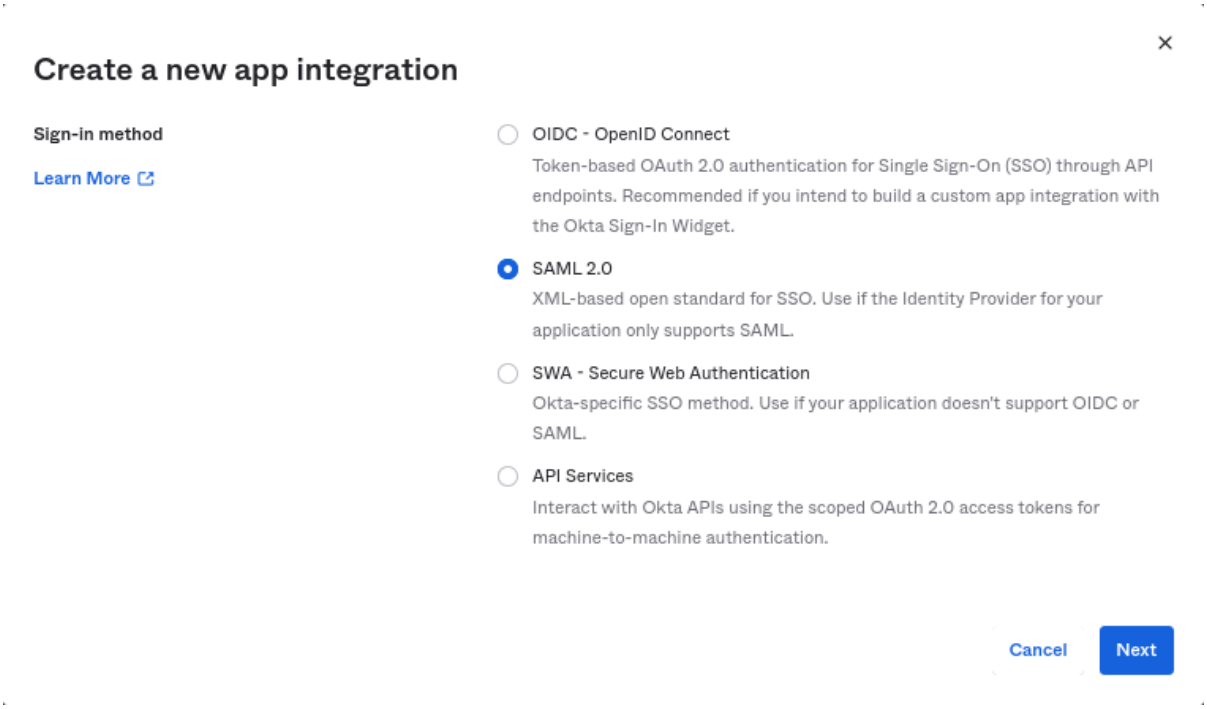
В этом разделе приведены рекомендации по настройке Okta для включения аутентификации SAML 2.0 и подготовки пользователей в Zabbix.

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*.

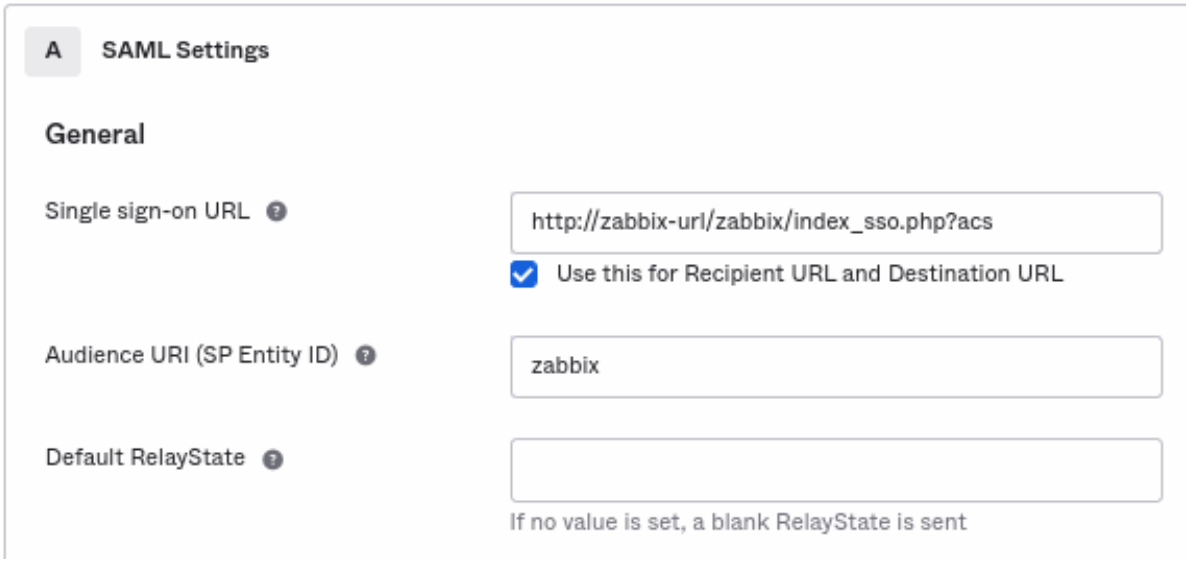
3. Нажмите *Create App Integration*.



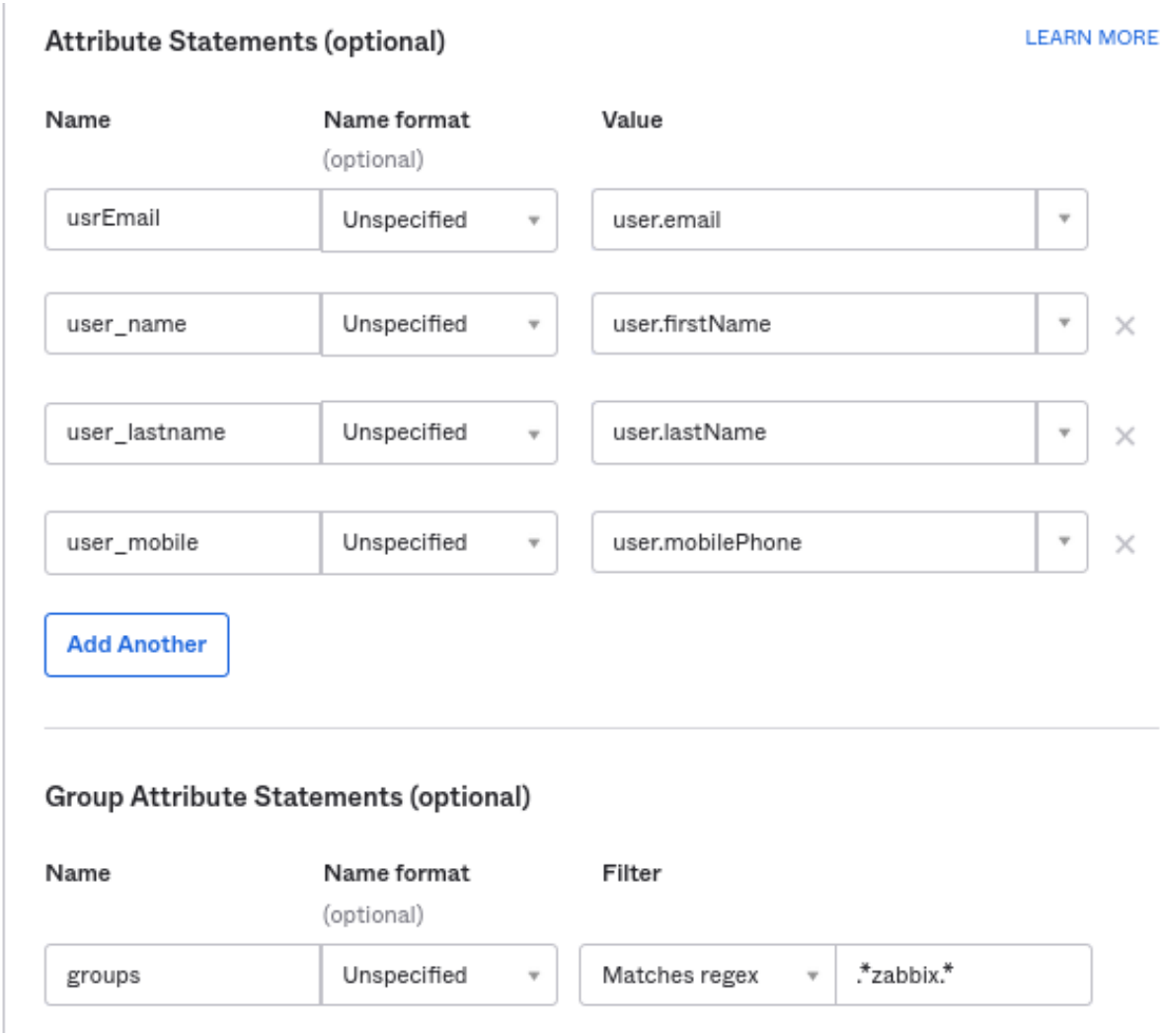
Выберите "SAML 2.0" в качестве метода входа и нажмите *Next*.

4. В общих настройках укажите имя приложения и нажмите *Next*.

5. В настройках SAML введите приведенные ниже значения, затем нажмите *Next*.



- В разделе **General** добавьте:
  - *Single sign-on URL*: `http://<your-zabbix-url>/zabbix/index_sso.php?acs`
    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`
    Обратите внимание, что это значение будет использоваться в SAML-утверждении 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 в соответствии с вашими предпочтениями.
- В разделе **Attribute Statements/Group Attribute Statements** добавьте:



Эти операторы атрибутов вставляются в SAML-утверждения, передаваемые в Zabbix.

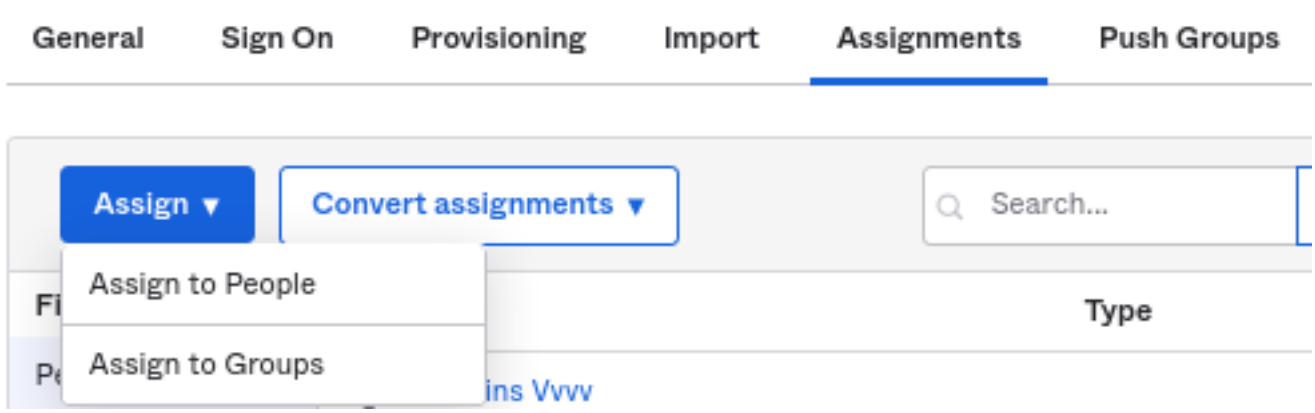
Используемые здесь имена атрибутов являются произвольными примерами. Вы можете использовать другие имена атрибутов, однако необходимо, чтобы они совпадали со значением соответствующего поля в настройках Zabbix SAML.

Если вы хотите настроить вход в Zabbix через SAML *без* JIT-подготовки пользователей, 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 *Assertion Encryption* установлено в 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" из выпадающего списка.



8. Во всплывающем окне назначьте приложение пользователям, которые будут использовать SAML 2.0 для аутентификации в Zabbix, затем нажмите *Save and go back*.

9. Загрузите сертификат IdP. Установите для него права 644, выполнив:

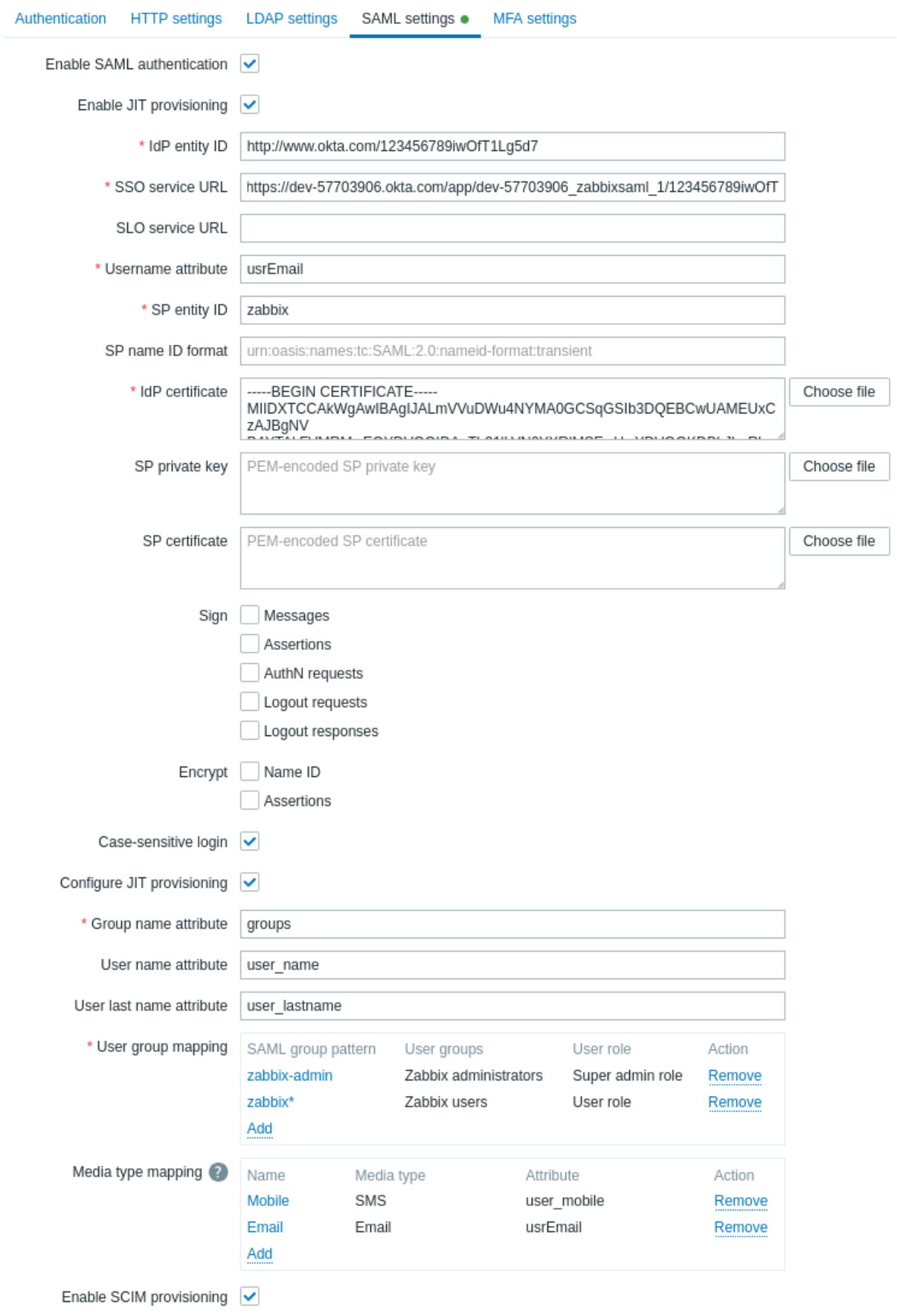
```
chmod 644 idp.crt
```

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 настроек SAML и заполните параметры конфигурации на основе инструкций по настройке из Okta:



Поле Zabbix	Поле настройки в Okta	Пример значения
IdP entity ID	Identity Provider Issuer	
SSO service URL	Identity Provider Single Sign-On URL	
Username attribute	Attribute name	`usrEmail`
SP entity ID	Audience URI	`zabbix`
Group name attribute	Attribute name	`groups`

Поле Zabbix	Поле настройки в Okta	Пример значения
User name attribute	Attribute name	`user_name`
User last name attribute	Attribute name	`user_lastname`

Также необходимо настроить сопоставление групп пользователей и способов оповещения.

2. Добавьте сертификат Base64, указанный в инструкциях по настройке Okta SAML.

Если в zabbix.conf.php задано `$SSO['CERT_STORAGE'] = 'database'`, вы можете вставить текст сертификата или загрузить файл сертификата в веб-интерфейсе во время настройки SAML — файлы в файловой системе не требуются.

Если в zabbix.conf.php задано `$SSO['CERT_STORAGE'] = 'file'`, сертификат должен быть доступен в файловой системе (по умолчанию в ui/conf/certs или по пути, настроенному в zabbix.conf.php), и веб-интерфейс не будет хранить сертификаты в базе данных.
Обратите внимание, что если `$SSO['CERT_STORAGE']` не задан или закомментирован, предполагается хранение в файлах, и сертификаты считываются из ui/conf/certs.

3. Если в Okta для *Assertion Encryption* установлено значение "Encrypted", в Zabbix также следует установить флажок "Assertions" у параметра *Encrypt*.

4. Нажмите кнопку *Update*, чтобы сохранить эти настройки.

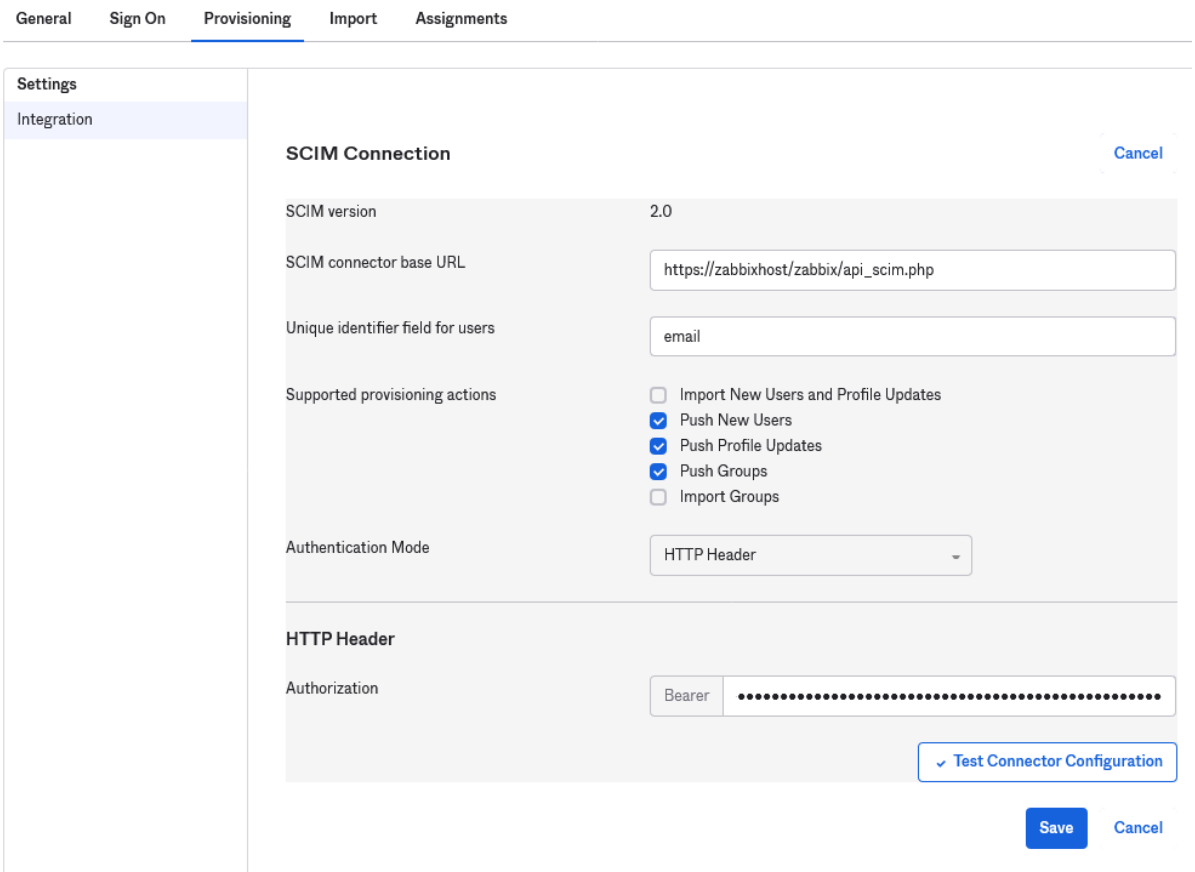
SCIM-подготовка

1. Чтобы включить SCIM-подготовку, перейдите в "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-соединение:

- В *SCIM connector base URL* укажите путь к веб-интерфейс 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 Super admin



Attention:
Если у вас возникают проблемы с аутентификацией, см. Authorization header forwarding.

3. Нажмите *Test Connector Configuration*, чтобы проверить соединение. Если все настроено правильно, будет показано сообщение об успешном выполнении.

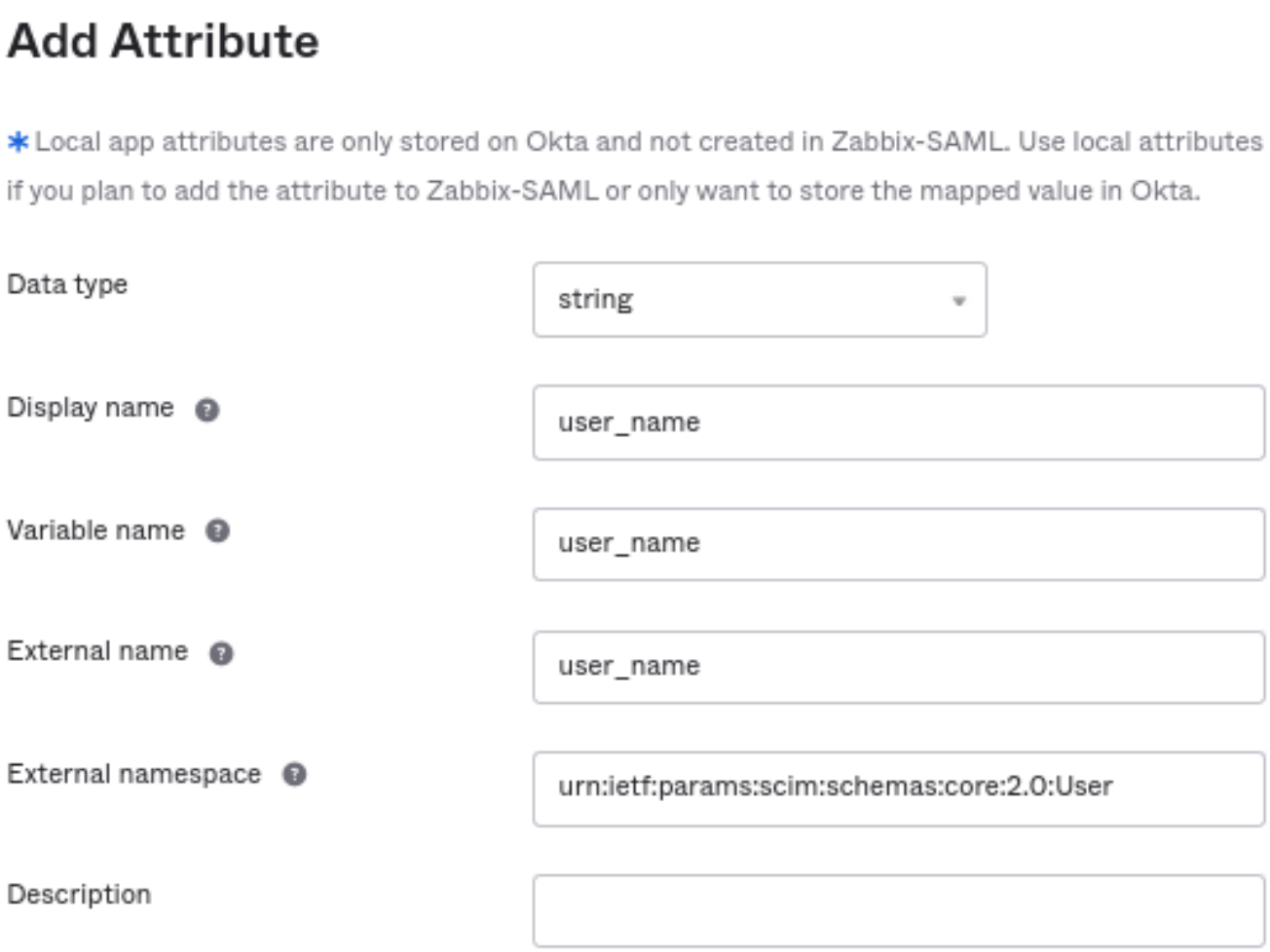
4. В разделе ”Provisioning” -> ”To App” убедитесь, что установлены следующие флажки:

- Create Users
- Update User Attributes
- Deactivate Users

Это гарантирует, что запросы этих типов будут отправляться в Zabbix.

5. Убедитесь, что все атрибуты, определенные в SAML, определены и в SCIM. Вы можете открыть редактор профиля для вашего приложения в ”Provisioning” -> ”To App”, нажав *Go to Profile Editor*.

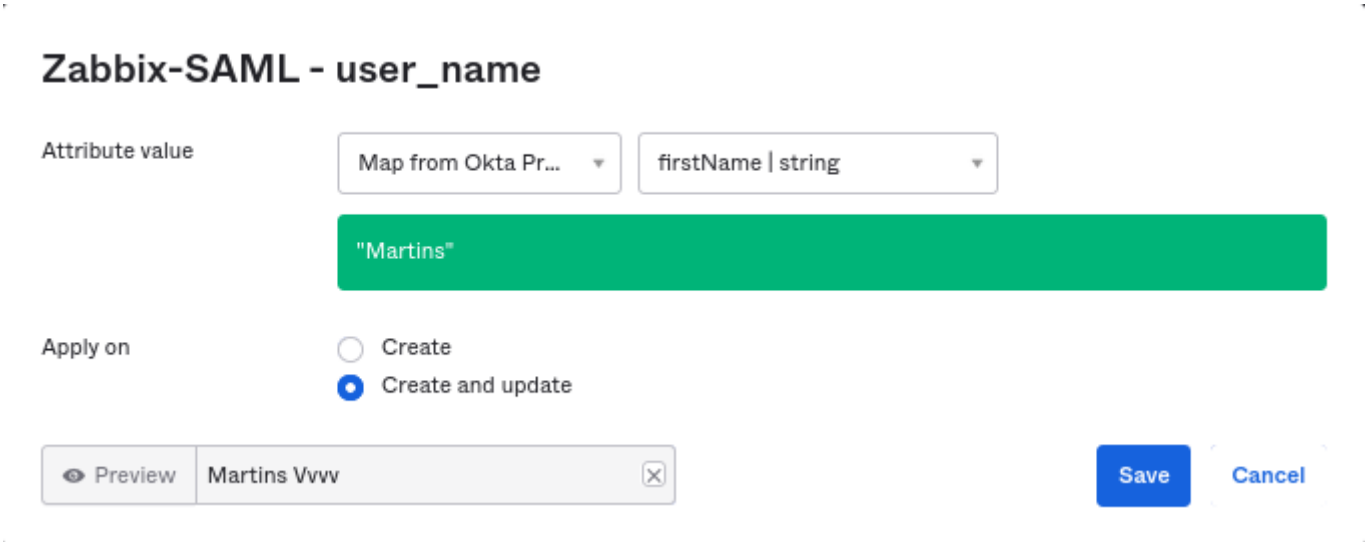
Нажмите *Add Attribute*. Заполните значения *Display name*, *Variable name*, *External name* именем атрибута SAML, например `user_name`.



Значение *External namespace* должно совпадать со схемой пользователя: `urn:ietf:params:scim:schemas:core:2.0:User`

6. Перейдите в ”Provisioning” -> ”To App” -> ”Attribute Mappings” вашего приложения. Нажмите *Show Unmapped Attributes* внизу страницы. Появятся недавно добавленные атрибуты.

7. Сопоставьте каждый добавленный атрибут.



8. Добавьте пользователей на вкладке ”Assignments”. Предварительно эти пользователи должны быть добавлены в *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 Zabbix SAML должен соответствовать группе, указанной здесь. Если совпадения нет, пользователь не сможет быть создан в Zabbix.

Информация о членах группы отправляется каждый раз при внесении каких-либо изменений.

14 Настройка SAML с OneLogin

Обзор

В этом разделе приведены рекомендации по настройке единого входа и автоматического предоставления учетных записей пользователей в Zabbix из OneLogin с использованием аутентификации SAML 2.0.

Настройка OneLogin

Создание приложения

1. Войдите в свою учетную запись OneLogin. Для целей 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. Инструкции на этой странице основаны на примере приложения *SCIM Provisioner with SAML (SCIM v2 Enterprise, full SAML)*.

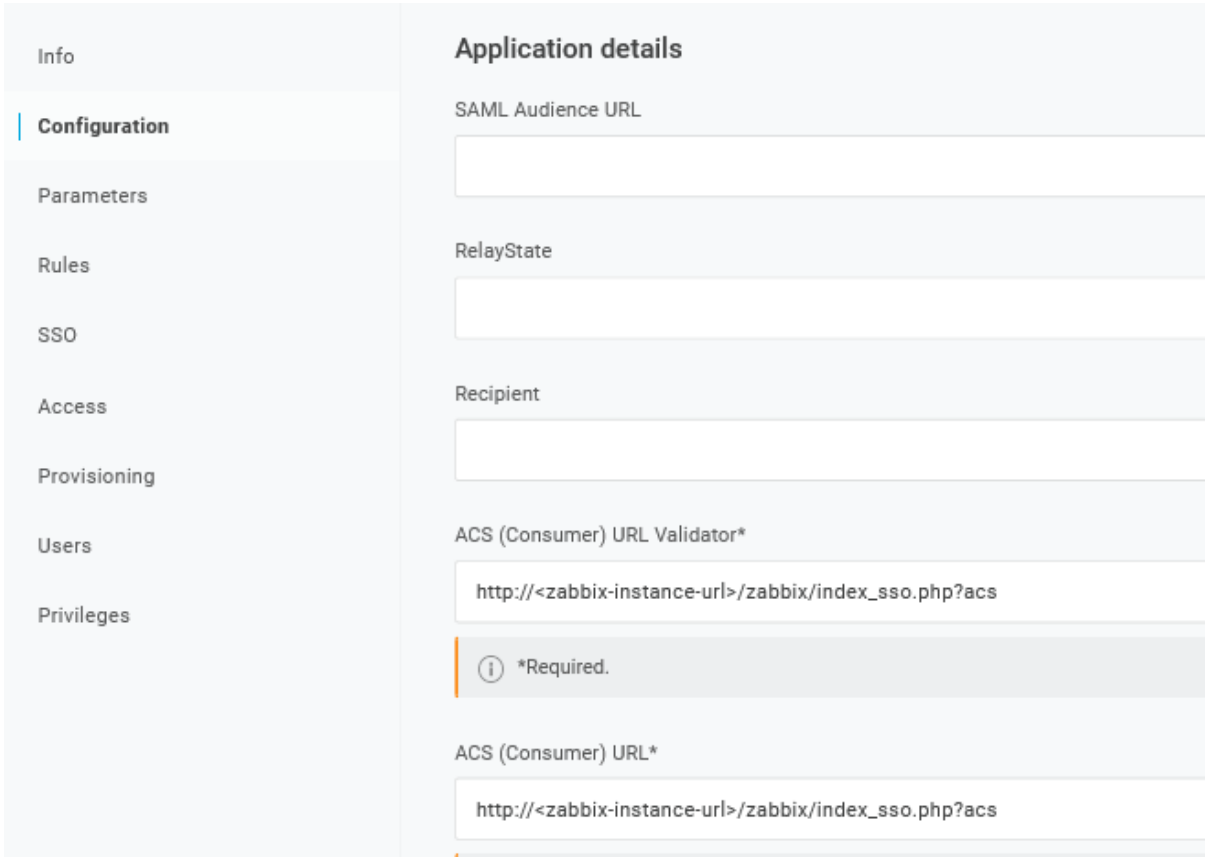
4. Для начала вы можете настроить отображаемое имя вашего приложения. Вы также можете добавить значок и сведения о приложении. После этого нажмите *Save*.

Настройка SSO/SCIM-подготовки пользователей

1. В разделе *Configuration -> Application details* укажите конечную точку единого входа Zabbix `http://<zabbix-instance-url>/zab` в качестве значения следующих полей:

- *ACS (Consumer) URL Validator*
- *ACS (Consumer) URL*

Обратите внимание на использование ”http”, а не ”https”, чтобы параметр `acs` не был обрезан в запросе.



Также возможно использовать "https". Чтобы это работало с Zabbix, необходимо добавить в `conf/zabbix.conf.php` следующую строку:

```
$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 вы хотите передавать в Zabbix через SCIM, например `user_name`, `user_lastname`, `user_email` и `user_mobile`:

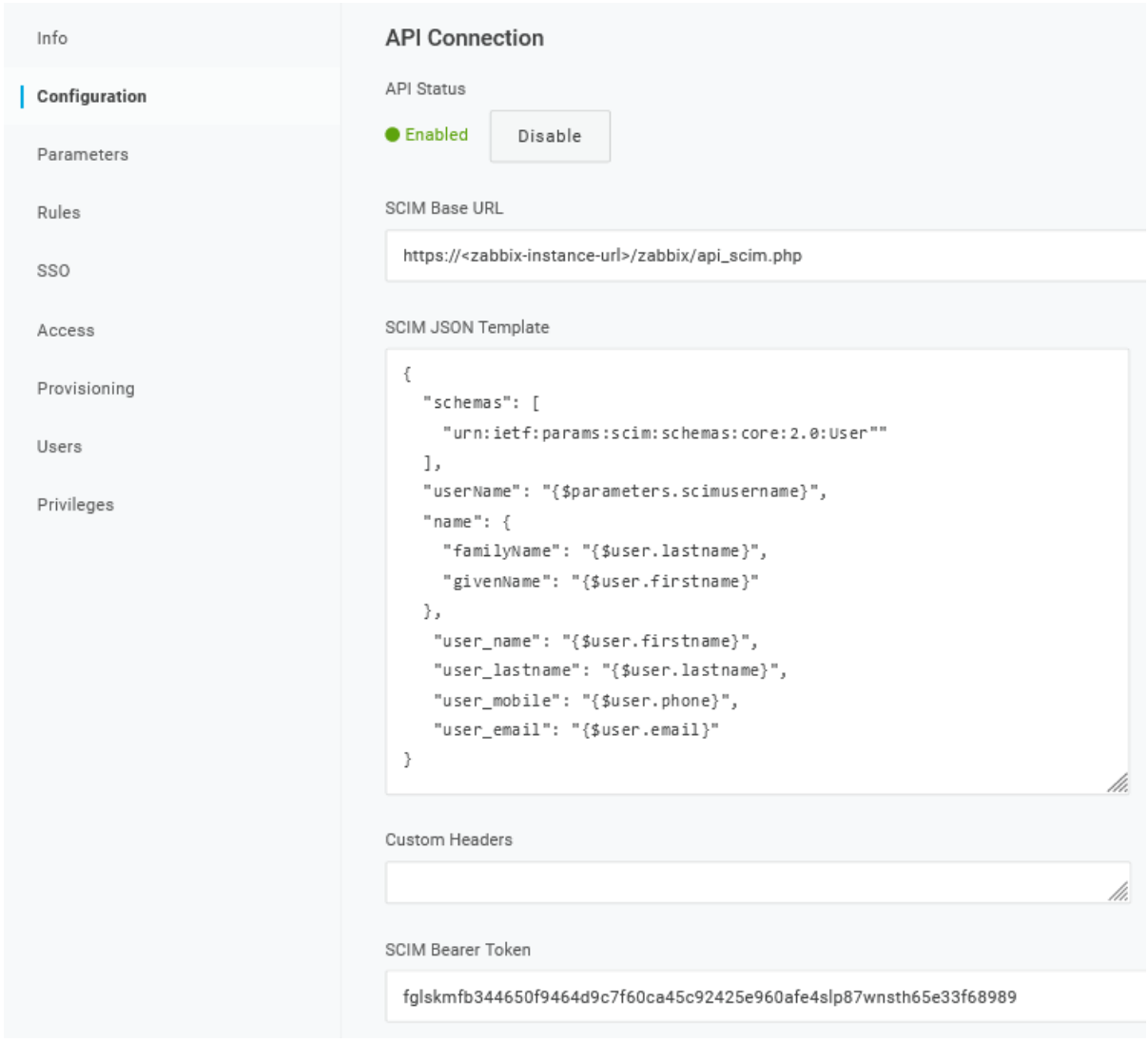
```
{
  "schemas": [
    "urn:ietf:params:scim:schemas:core:2.0:User"
  ],
  "userName": "{$parameters.scimusername}",
  "name": {
    "familyName": "{$user.lastname}",
    "givenName": "{$user.firstname}"
  },
   "user_name": "{$user.firstname}",
   "user_lastname": "{$user.lastname}",
   "user_mobile": "{$user.phone}",
   "user_email": "{$user.email}"
}
```

Имена атрибутов являются произвольными. Можно использовать другие имена атрибутов, однако требуется, чтобы они совпадали со значением соответствующего поля в настройках Zabbix SAML.

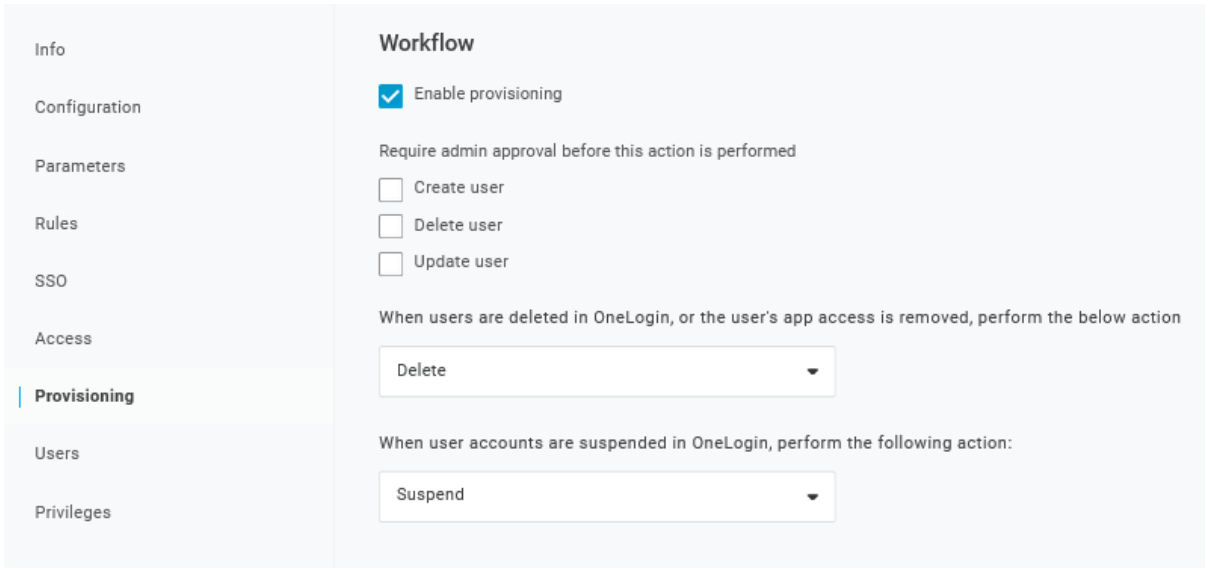
Обратите внимание, что для работы подготовки пользователей OneLogin должен получать в ответе атрибут 'name' с 'givenName' и 'familyName', даже если это не требовалось поставщиком услуг. Поэтому необходимо указать это в схеме в разделе конфигурации приложения.

- *SCIM Bearer Token*: введите токен Zabbix API с правами Super admin.

Нажмите *Enable*, чтобы активировать соединение.

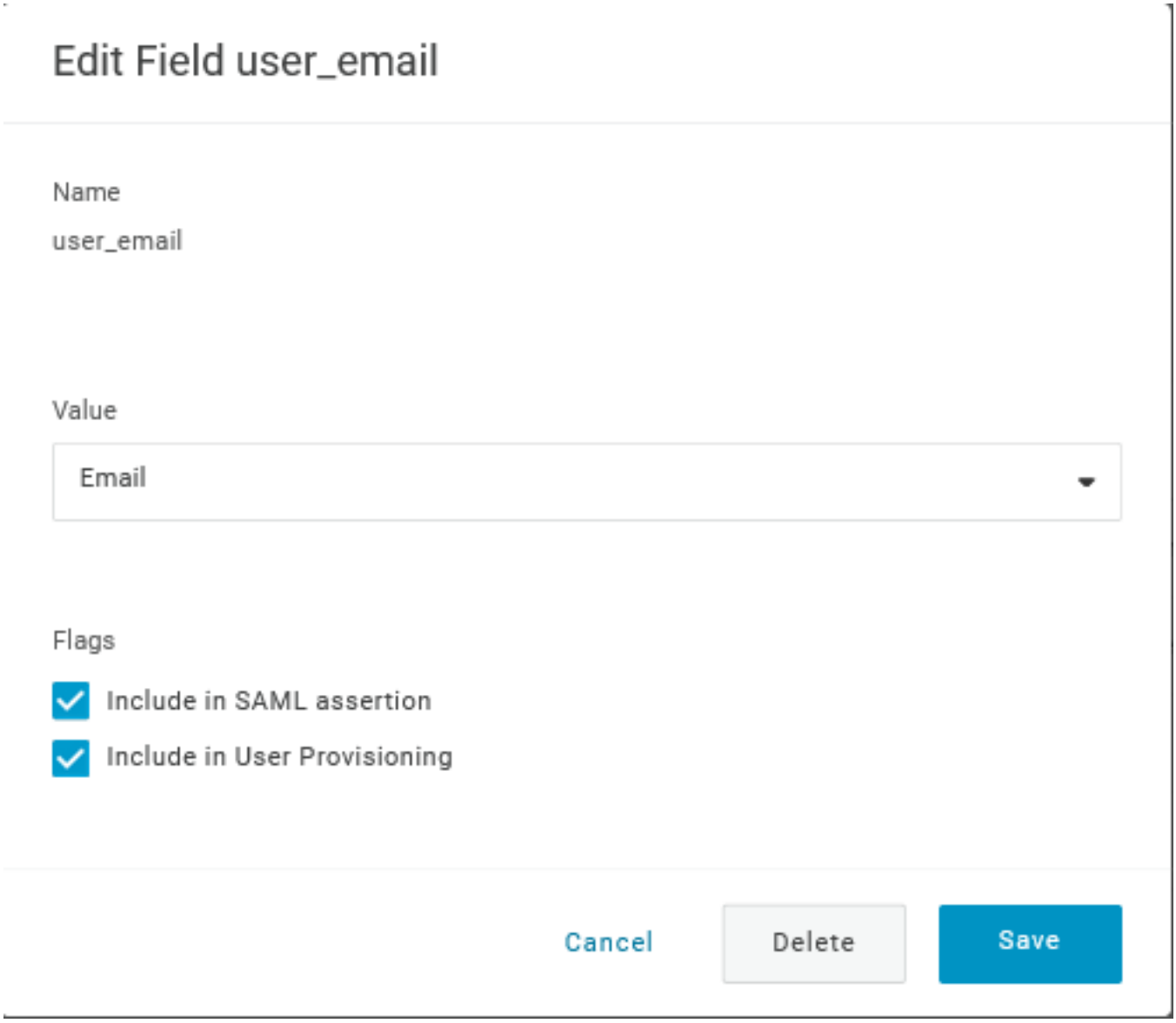


3. На странице *Provisioning* включите параметр Provisioning:



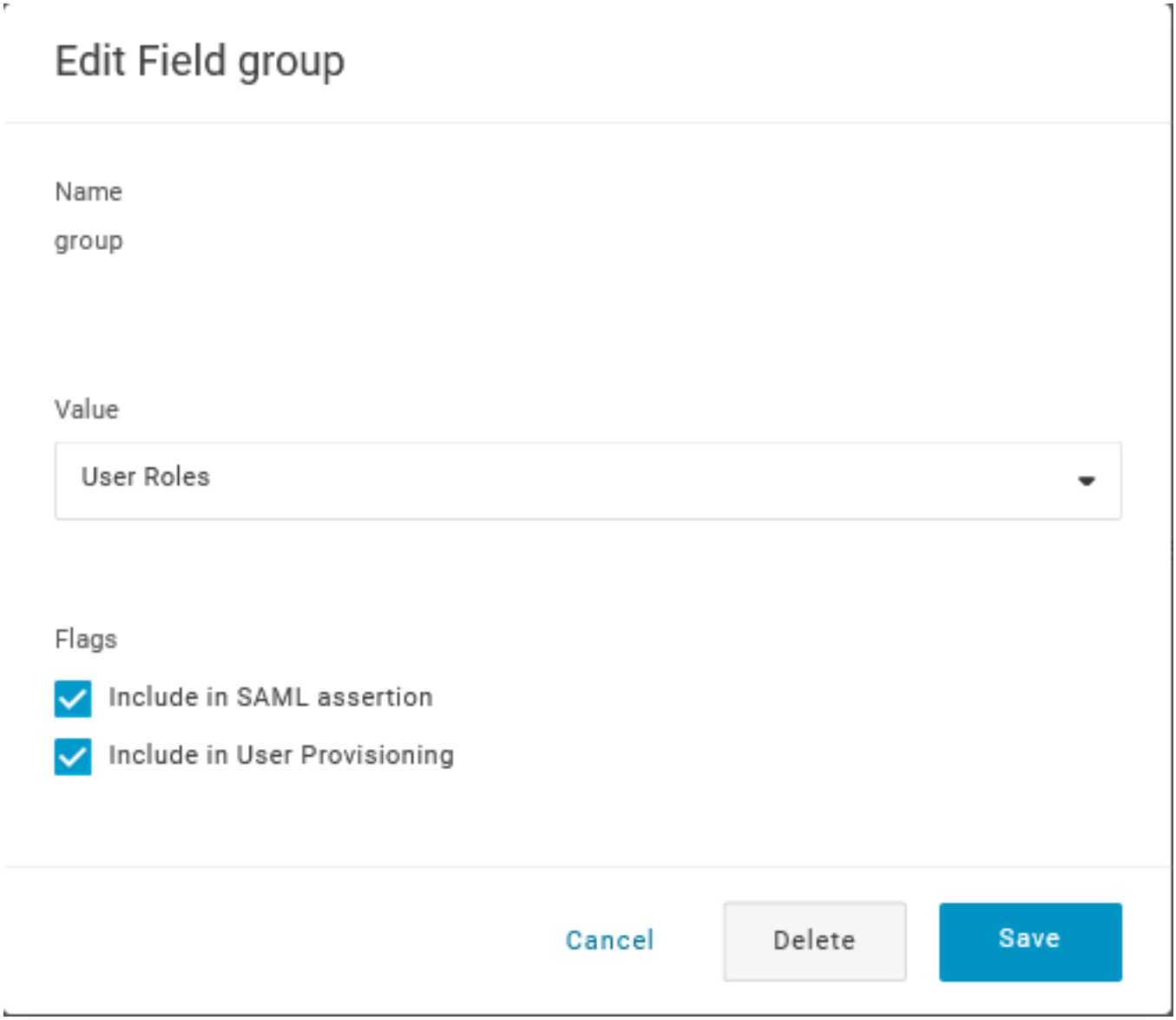
4. На странице *Parameters* содержится список параметров по умолчанию:

- Убедитесь, что 'scimusername' соответствует значению логина пользователя в OneLogin (например, email);
- Отметьте параметр *Include in User Provisioning* для параметра 'Groups';
- Нажмите "+", чтобы создать пользовательские параметры, необходимые для SAML-утверждений и подготовки пользователей, например `user_name`, `user_lastname`, `user_email` и `user_mobile`:

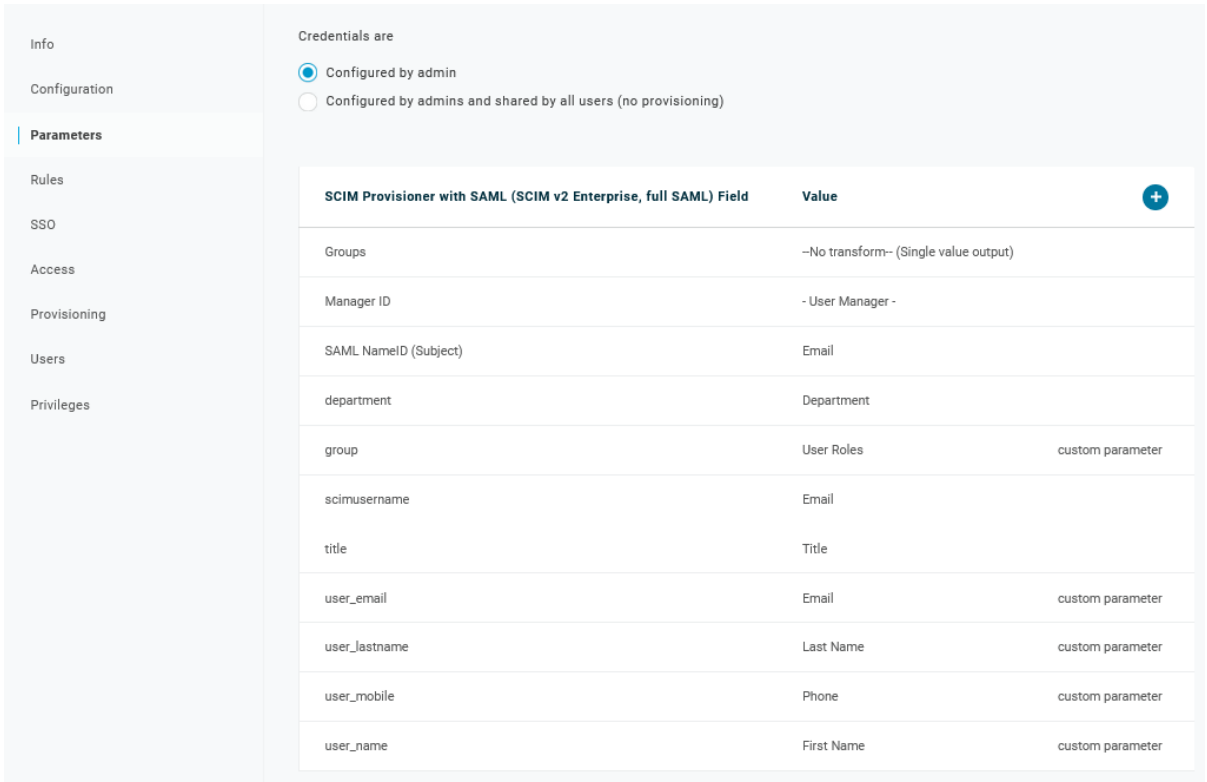


При добавлении параметра обязательно отметьте оба параметра: *Include in SAML assertion* и *Include in User Provisioning*.

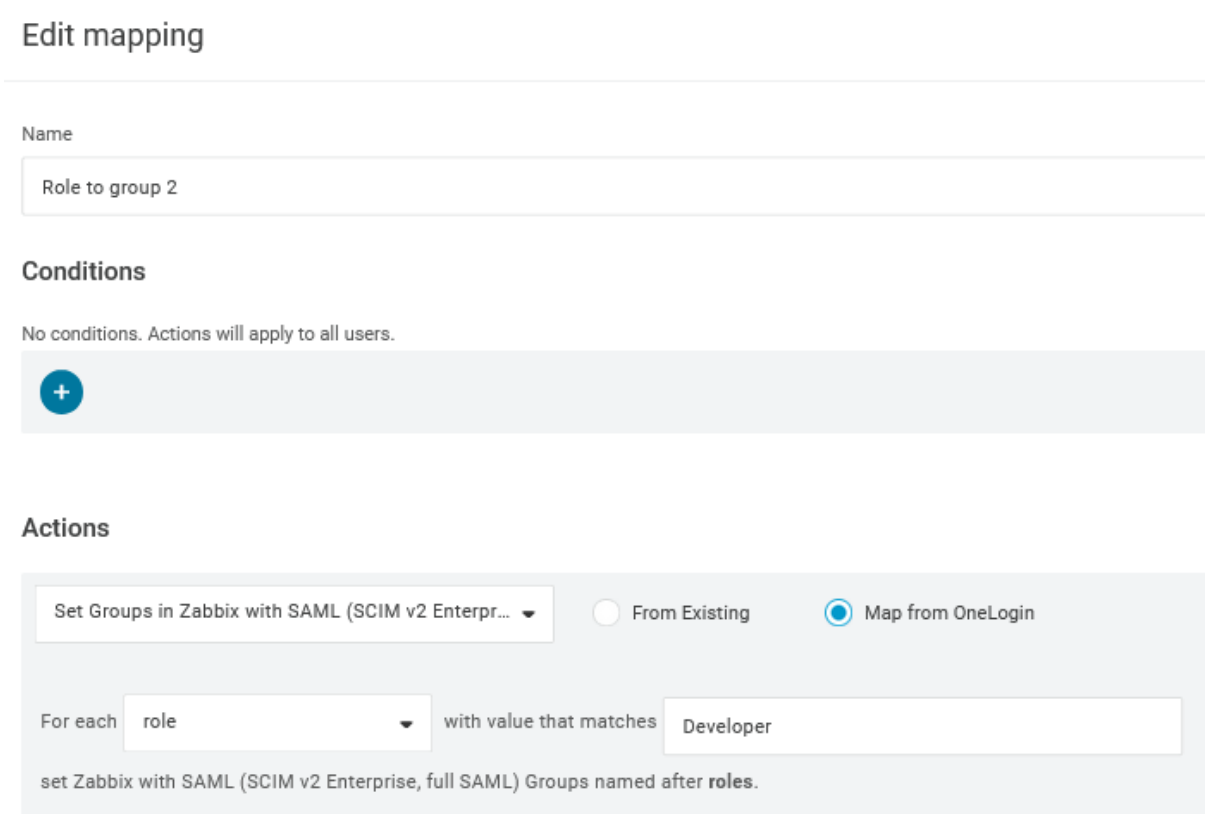
- Добавьте параметр 'group', соответствующий ролям пользователей в OneLogin. Роли пользователей будут передаваться в виде строки, разделённой точкой с запятой `;`. Роли пользователей OneLogin будут использоваться для создания групп пользователей в Zabbix:



Проверьте список параметров:



5. На странице *Rules* создайте сопоставления ролей пользователей со стандартным параметром Groups.



Вы можете использовать регулярное выражение для передачи определённых ролей как групп. Имена ролей не должны содержать `;`, так как OneLogin использует этот символ как разделитель при отправке атрибута с несколькими ролями.

6. Загрузите сертификат IdP. Установите для него права 644, выполнив:

```
chmod 644 idp.crt
```

Конфигурация Zabbix

1. В Zabbix перейдите в настройки SAML и заполните параметры конфигурации на основе конфигурации OneLogin:



Поле Zabbix	Поле настройки в OneLogin	Пример значения
IdP entity ID	Issuer URL (см. вкладку SSO вашего приложения в OneLogin)	

Поле Zabbix	Поле настройки в OneLogin	Пример значения
SSO service URL	SAML 2.0 Endpoint (HTTP) (см. вкладку SSO вашего приложения в OneLogin)	
SLO service URL	SLO Endpoint (HTTP) (см. вкладку SSO вашего приложения в OneLogin)	
Username attribute	Пользовательский параметр	`user_email`
Group name attribute	Пользовательский параметр	`group`
User name attribute	Пользовательский параметр	`user_name`
User last name attribute	Пользовательский параметр	`user_lastname`

Также необходимо настроить сопоставление групп пользователей. Сопоставление способов оповещения является необязательным. Нажмите *Update*, чтобы сохранить эти настройки.

2. Добавьте сертификат Base64, предоставленный OneLogin.

Если в zabbix.conf.php задано `$SSO['CERT_STORAGE'] = 'database'`, вы можете вставить текст сертификата или загрузить файл сертификата в веб-интерфейс во время настройки SAML — файлы в файловой системе не требуются.

Если в zabbix.conf.php задано `$SSO['CERT_STORAGE'] = 'file'`, сертификат должен быть доступен в файловой системе (по умолчанию в ui/conf/certs или по пути, настроенному в zabbix.conf.php), и веб-интерфейс не будет хранить сертификаты в базе данных. Обратите внимание, что если `$SSO['CERT_STORAGE']` не задан или закомментирован, предполагается файловое хранилище, и сертификаты считываются из ui/conf/certs.

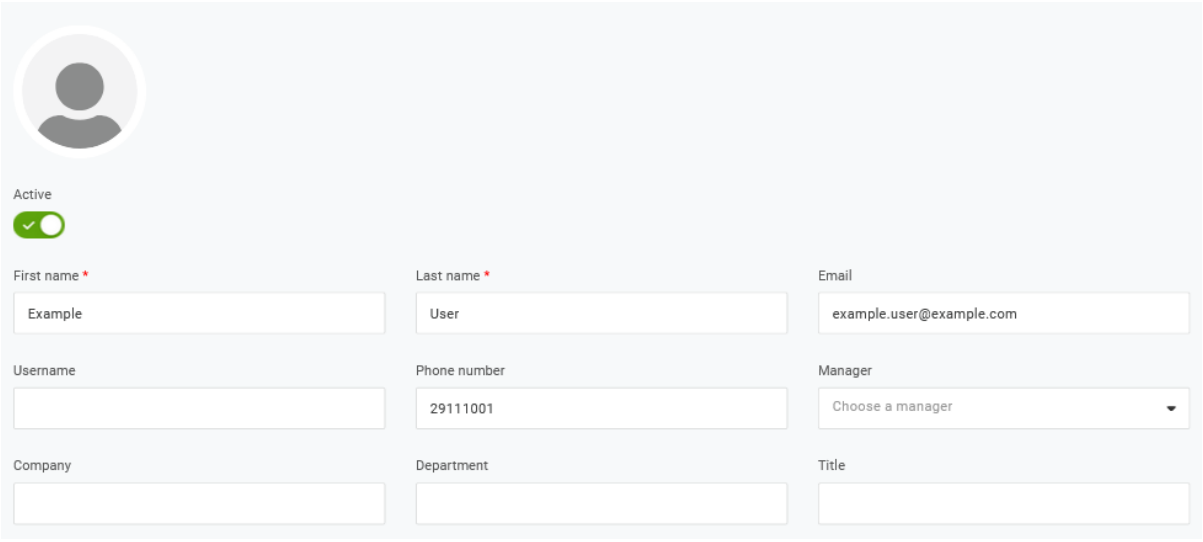
Доступ к загрузке сертификата в OneLogin можно получить в *Applications* -> *SSO* -> нажмите *View details* под текущим сертификатом.

3. Нажмите кнопку *Update*, чтобы сохранить эти настройки.

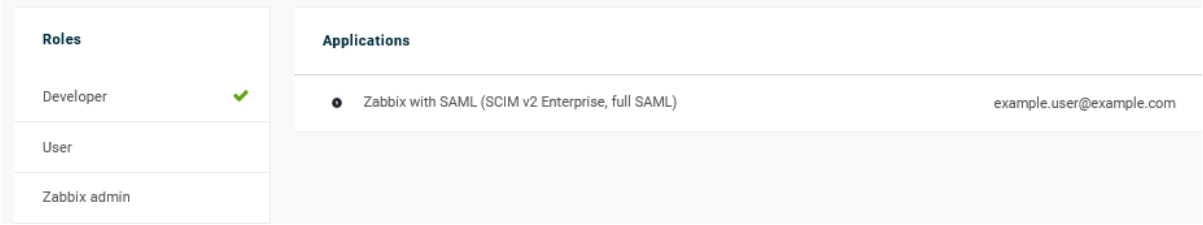
SCIM-подготовка пользователей

Если подготовка пользователей включена, теперь можно добавлять/обновлять пользователей и их роли в OneLogin, и они будут немедленно подготовлены в Zabbix.

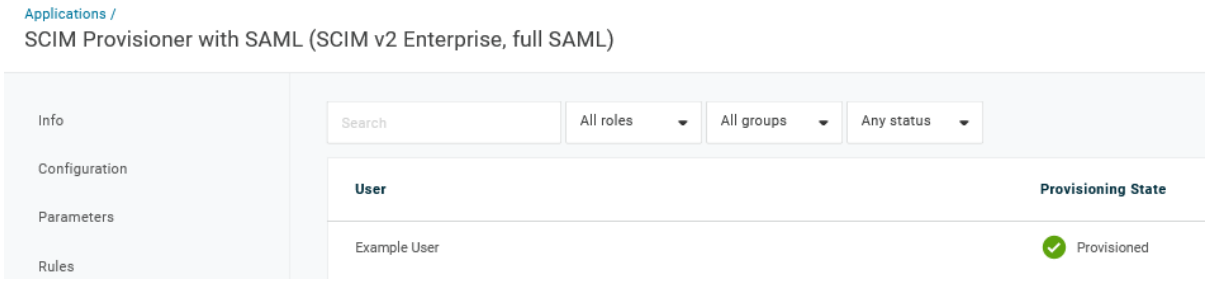
Например, вы можете создать нового пользователя:



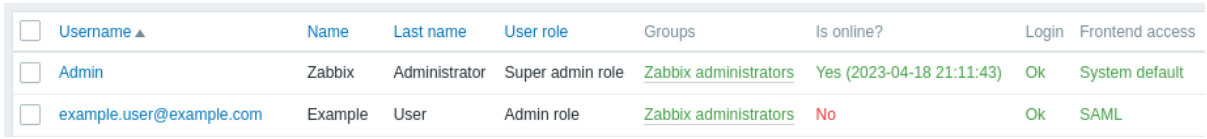
Добавьте его к роли пользователя и приложению, которое будет выполнять подготовку пользователя:



При сохранении пользователя он будет подготовлен в Zabbix. В Application -> Users вы можете проверить статус подготовки текущих пользователей приложения:



Если подготовка выполнена успешно, пользователь будет виден 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

15 Настройка отчетов по расписанию

Обзор

В этом разделе приведены инструкции по установке веб-службы Zabbix и настройке Zabbix для включения генерации отчетов по расписанию.

Установка

Для включения генерации отчетов по расписанию необходимо установить новый процесс веб-службы Zabbix и браузер Google Chrome. Веб-служба может быть установлена на той же машине, где установлен сервер Zabbix, или на другой машине. Браузер Google Chrome должен быть установлен на той же машине, где установлена веб-служба.

Официальный пакет zabbix-web-service доступен в репозитории Zabbix. Браузер Google Chrome не входит в состав этих пакетов и должен быть установлен отдельно.

Чтобы скомпилировать веб-службу Zabbix из исходного кода, смотрите Установка веб-службы Zabbix.

После установки запустите zabbix_web_service на машине, где установлена веб-служба:

```
zabbix_web_service
```

Конфигурация

Чтобы обеспечить корректное взаимодействие между всеми задействованными элементами, убедитесь, что файл конфигурации сервера и параметры конфигурации веб-интерфейса настроены должным образом.

Сервер Zabbix

Следующие параметры в файле конфигурации сервера Zabbix необходимо обновить: *WebServiceURL* и *StartReportWriters*.

WebServiceURL

Этот параметр необходим для включения взаимодействия с веб-службой. URL-адрес должен быть в формате `http[s]://host:port/report`.

- По умолчанию веб-служба прослушивает порт 10053. Другой порт можно указать в файле конфигурации веб-службы.
- Указание пути `/report` является обязательным (путь жестко задан 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, сколько процессов записи отчетов должно быть запущено. Если он не задан или равен 0, генерация отчетов отключена. В зависимости от требуемого количества и частоты отчетов можно включить от 1 до 100 процессов записи отчетов.

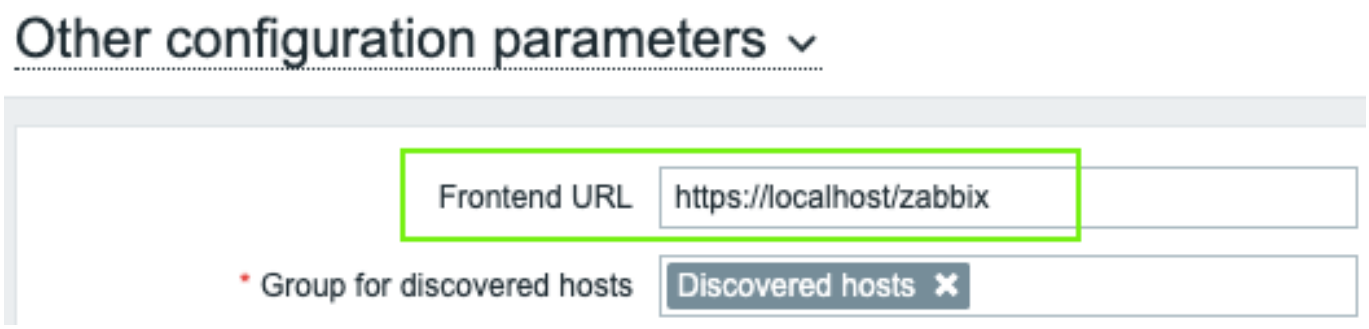
Пример:

```
StartReportWriters=3
```

Веб-интерфейс Zabbix

Для включения взаимодействия между веб-интерфейсом Zabbix и веб-службой Zabbix необходимо задать параметр *Frontend URL*:

- Перейдите в раздел меню веб-интерфейса *Administration* > *General* > *Other*
- Укажите полный URL веб-интерфейса Zabbix в параметре *Frontend URL*



Note:
После завершения процедуры настройки вы можете настроить и отправить тестовый отчет, чтобы убедиться, что все работает корректно.

16 Дополнительные языки веб-интерфейса

Обзор

Чтобы использовать в веб-интерфейсе Zabbix любой язык, отличный от английского, его локаль должна быть установлена на веб-сервере. Кроме того, для работы переводов требуется PHP-расширение 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.

Теперь локали должны быть установлены. Для появления новых языков может потребоваться перезагрузить страницу веб-интерфейс Zabbix в браузере с помощью Ctrl + F5.

Установка Zabbix

Если Zabbix устанавливается напрямую из git-репозитория Zabbix, файлы перевода необходимо сгенерировать вручную. Чтобы сгенерировать файлы перевода, выполните:

```
make gettext
locale/make_mo.sh
```

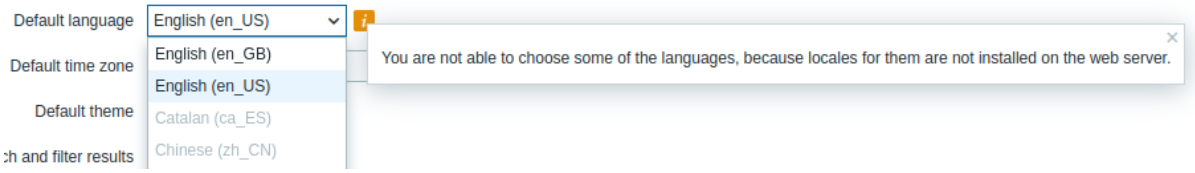
Этот шаг не требуется при установке Zabbix из пакетов или исходных файлов tar.gz.

Выбор языка

Существует несколько способов выбрать язык в веб-интерфейсе Zabbix:

- При установке веб-интерфейса — в мастере установки frontend. Выбранный язык будет установлен как системный язык по умолчанию.
- После установки системный язык по умолчанию можно изменить в разделе меню *Administration→General→GUI*.
- Язык для конкретного пользователя можно изменить в профиле пользователя.

Если локаль для какого-либо языка не установлена на машине, этот язык будет отображаться серым цветом в селекторе языков Zabbix. Если отсутствует хотя бы одна локаль, рядом с селектором языков отображается оранжевый значок. При нажатии на этот значок будет показано следующее сообщение: ”Вы не можете выбрать некоторые языки, потому что локали для них не установлены на веб-сервере.”



2 Настройка процесса

Пожалуйста, используйте боковую панель для доступа к содержимому этого раздела.

1 Zabbix сервер

Обзор

В этом разделе перечислены параметры, поддерживаемые файлом конфигурации сервера Zabbix (zabbix_server.conf).

Параметры перечислены без дополнительной информации. Нажмите на параметр, чтобы увидеть полные сведения.

Parameter	Description
AlertScriptsPath	Расположение пользовательских скриптов оповещений.
AllowRoot	Разрешить запуск сервера от имени ’root’.
AllowSoftwareUpdateCheck	Разрешить пользовательскому интерфейсу Zabbix получать информацию об обновлениях программного обеспечения с zabbix.com.
AllowUnsupportedDBVersions	Разрешить серверу работать с неподдерживаемыми версиями базы данных.
CacheSize	Размер кэша конфигурации.
CacheUpdateFrequency	Этот параметр определяет, как часто Zabbix будет обновлять кэш конфигурации, в секундах.
DBHost	Имя узла базы данных.
DBName	Имя базы данных.
DBPassword	Пароль базы данных.
DBPort	Порт базы данных, если не используется локальный сокет.
DBSchema	Имя схемы базы данных. Используется для PostgreSQL.
DBSocket	Путь к файлу сокета MySQL.
DBUser	Пользователь базы данных.
DBTLSConnect	Установка этой опции в указанное значение принудительно включает использование TLS-соединения с базой данных.
DBTLSCAFile	Полный путь к файлу, содержащему сертификаты корневых центров сертификации (CA) для проверки сертификата базы данных.
DBTLSCertFile	Полный путь к файлу, содержащему сертификат сервера Zabbix для аутентификации в базе данных.
DBTLSKeyFile	Полный путь к файлу, содержащему закрытый ключ для аутентификации в базе данных.
DBTLSCipher	Список шифров шифрования, которые сервер Zabbix разрешает для протоколов TLS вплоть до TLS v1.2. Поддерживается только для MySQL.
DBTLSCipher13	Список наборов шифров, которые сервер Zabbix разрешает для протокола TLS v1.3. Поддерживается только для MySQL, начиная с версии 8.0.16.
DebugLevel	Укажите уровень отладки.
EnableGlobalScripts	Включить глобальные скрипты на сервере Zabbix.
ExportDir	Каталог для экспорта в реальном времени событий, истории и трендов в формате JSON с разделением строками. Если задан, включает экспорт в реальном времени.
ExportFileSize	Максимальный размер одного файла экспорта в байтах.
ExportType	Список типов сущностей, разделённых запятыми (events, history, trends), для экспорта в реальном времени (по умолчанию все типы).
ExternalScripts	Расположение внешних скриптов.
Fping6Location	Расположение fping6.
FpingLocation	Расположение fping.

Parameter	Description
FrontendAllowedIP	Список IP-адресов или диапазонов CIDR, разделённых запятыми, которым разрешено подключение из веб-интерфейса.
HANodeName	Имя узла кластера высокой доступности.
HistoryCacheSize	Размер кэша истории.
HistoryIndexCacheSize	Размер кэша индекса истории.
HistoryStorageDateIndex	Включить предварительную обработку значений истории в хранилище истории для сохранения значений в разных индексах в зависимости от даты.
HistoryStorageURL	HTTP[S] URL хранилища истории.
HistoryStorageTypes	Список типов значений, разделённых запятыми, которые должны отправляться в хранилище истории.
HousekeepingFrequency	Этот параметр определяет, как часто Zabbix будет выполнять процедуру очистки, в часах.
Include	В файл конфигурации можно включать отдельные файлы или все файлы из каталога.
JavaGateway	IP-адрес (или имя узла) Zabbix Java gateway.
JavaGatewayPort	Порт, на котором прослушивает Zabbix Java gateway.
ListenBacklog	Максимальное количество ожидающих соединений в очереди TCP.
ListenIP	Список IP-адресов, разделённых запятыми, которые trapper должен прослушивать.
ListenPort	Порт прослушивания для trapper.
LoadModule	Модуль, загружаемый при запуске сервера.
LoadModulePath	Полный путь к расположению модулей сервера.
LogFile	Имя файла журнала.
LogFileSize	Максимальный размер файла журнала.
LogSlowQueries	Определяет, сколько времени может занимать запрос к базе данных до его записи в журнал, в миллисекундах.
LogType	Тип вывода журнала.
MaxConcurrentChecksPerPoller	Максимальное количество асинхронных проверок, которые могут одновременно выполняться каждым HTTP агент poller, агент poller или SNMP poller.
MaxHousekeeperDelete	За одну задачу в одном цикле очистки будет удалено не более ’MaxHousekeeperDelete’ строк (соответствующих [tablename], [field], [value]).
NodeAddress	IP-адрес или имя узла с необязательным портом для переопределения способа подключения веб-интерфейса к серверу.
PidFile	Имя PID-файла.
ProblemHousekeepingFrequency	Определяет, как часто Zabbix будет удалять проблемы для удалённых триггеров.
ProxyConfigFrequency	Определяет, как часто сервер Zabbix отправляет данные конфигурации прокси Zabbix.
ProxyDataFrequency	Определяет, как часто сервер Zabbix запрашивает данные истории у прокси Zabbix.
ServiceManagerSyncFrequency	Определяет, как часто Zabbix будет синхронизировать конфигурацию менеджера сервисов.
SMSDevices	Список файлов модемов, разделённых запятыми, которые разрешено использовать серверу Zabbix.
SNMPTrapperFile	Временный файл, используемый для передачи данных от демона SNMP trap серверу.
SocketDir	Каталог для хранения IPC-сокетов, используемых внутренними службами Zabbix.
SourceIP	Исходный IP-адрес.
SSHKeyLocation	Расположение открытых и закрытых ключей для SSH-проверок и действий.
SSLCertLocation	Расположение файлов клиентских SSL-сертификатов для аутентификации клиента.
SSLKeyLocation	Расположение файлов закрытых SSL-ключей для аутентификации клиента.
SSLCALocation	Переопределяет расположение файлов центра сертификации (CA) для проверки SSL-сертификата сервера.
StartAgentPollers	Количество предварительно созданных экземпляров асинхронных poller’ов Zabbix агента.
StartAlerters	Количество предварительно созданных экземпляров alerter’ов.
StartBrowserPollers	Количество предварительно созданных экземпляров poller’ов элементов данных браузера.
StartConnectors	Количество предварительно созданных экземпляров рабочих процессов коннекторов.
StartDBSyncers	Количество предварительно созданных экземпляров синхронизаторов истории.
StartDiscoverers	Количество предварительно созданных экземпляров процессов обнаружения.
StartEscalators	Количество предварительно созданных экземпляров escalator’ов.
StartHistoryPollers	Количество предварительно созданных экземпляров poller’ов истории.
StartHTTPAgentPollers	Количество предварительно созданных экземпляров асинхронных HTTP агент poller’ов.
StartHTTPPollers	Количество предварительно созданных экземпляров HTTP poller’ов.
StartIPMIPollers	Количество предварительно созданных экземпляров IPMI poller’ов.
StartJavaPollers	Количество предварительно созданных экземпляров Java poller’ов.
StartLLDProcessors	Количество предварительно созданных экземпляров процессов низкоуровневого обнаружения (LLD).
StartODBCPollers	Количество предварительно созданных экземпляров ODBC poller’ов.

Parameter	Description
StartPingers	Количество предварительно созданных экземпляров ICMP pinger'ов.
StartPollersUnreachable	Количество предварительно созданных экземпляров poller'ов для недоступных узлов сети (включая IPMI и Java).
StartPollers	Количество предварительно созданных экземпляров poller'ов.
StartPreprocessors	Количество предварительно запущенных экземпляров процессов предобработки.
StartProxyPollers	Количество предварительно созданных экземпляров poller'ов для пассивных прокси.
StartReportWriters	Количество предварительно созданных экземпляров генераторов отчётов.
StartSNMPPollers	Количество предварительно созданных экземпляров асинхронных SNMP poller'ов.
StartSNMPTrapper	Если установлено значение 1, будет запущен процесс SNMP trapper.
StartTimers	Количество предварительно созданных экземпляров таймеров.
StartTrappers	Количество предварительно созданных экземпляров trapper'ов.
StartVMwareCollectors	Количество предварительно созданных экземпляров сборщиков VMware.
StatsAllowedIP	Список IP-адресов, разделённых запятыми, при необходимости в нотации CIDR, или DNS-имён внешних экземпляров Zabbix. Запрос статистики будет принят только с адресов, перечисленных здесь.
Timeout	Указывает, как долго ждать (в секундах) установления соединения и обмена данными с прокси Zabbix, агентом, веб-службой, а также для SNMP-проверок (кроме элементов данных SNMP `walk[OID]` и `get[OID]`).
TLSCAFile	Полный путь к файлу, содержащему сертификаты корневых центров сертификации (CA) для проверки сертификата узла, используемому для шифрованного взаимодействия между компонентами Zabbix.
TLSCertFile	Полный путь к файлу, содержащему сертификат сервера или цепочку сертификатов, используемому для шифрованного взаимодействия между компонентами Zabbix.
TLSCipherAll	Строка приоритетов GnuTLS или строка шифров OpenSSL (TLS 1.2). Переопределяет критерии выбора наборов шифров по умолчанию для шифрования на основе сертификатов и PSK.
TLSCipherAll13	Строка шифров для OpenSSL 1.1.1 или новее в TLS 1.3. Переопределяет критерии выбора наборов шифров по умолчанию для шифрования на основе сертификатов и PSK.
TLSCipherCert	Строка приоритетов GnuTLS или строка шифров OpenSSL (TLS 1.2). Переопределяет критерии выбора наборов шифров по умолчанию для шифрования на основе сертификатов.
TLSCipherCert13	Строка шифров для OpenSSL 1.1.1 или новее в TLS 1.3. Переопределяет критерии выбора наборов шифров по умолчанию для шифрования на основе сертификатов.
TLSCipherPSK	Строка приоритетов GnuTLS или строка шифров OpenSSL (TLS 1.2). Переопределяет критерии выбора наборов шифров по умолчанию для шифрования на основе PSK.
TLSCipherPSK13	Строка шифров для OpenSSL 1.1.1 или новее в TLS 1.3. Переопределяет критерии выбора наборов шифров по умолчанию для шифрования на основе PSK.
TLSCRLFile	Полный путь к файлу, содержащему отозванные сертификаты. Этот параметр используется для шифрованного взаимодействия между компонентами Zabbix.
TLSFrontendAccept	Какие входящие соединения принимать от веб-интерфейса.
TLSFrontendCertIssuer	Разрешённый издатель сертификата веб-интерфейса.
TLSFrontendCertSubject	Разрешённый субъект сертификата веб-интерфейса.
TLSKeyFile	Полный путь к файлу, содержащему закрытый ключ сервера, используемому для шифрованного взаимодействия между компонентами Zabbix.
TLSListen	Управляет TLS на сокете trapper.
TmpDir	Временный каталог.
TrapperTimeout	Указывает тайм-аут в секундах для: - получения исторических данных от прокси Zabbix; - отправки данных конфигурации прокси Zabbix; - выполнения глобального скрипта или удалённой команды на сервере Zabbix.
TrendCacheSize	Размер кэша трендов.
TrendFunctionCacheSize	Размер кэша функций трендов.
UnavailableDelay	Определяет, как часто узел сети проверяется на доступность в период недоступности.
UnreachableDelay	Определяет, как часто узел сети проверяется на доступность в период недосягаемости.
UnreachablePeriod	Определяет, через сколько секунд недосягаемости узел сети считается недоступным.
User	Сбросить привилегии до конкретного существующего пользователя в системе.
ValueCacheSize	Размер кэша значений истории.
Vault	Указывает поставщика vault.
VaultDBPath	Указывает расположение, откуда учётные данные базы данных должны извлекаться по ключам.
VaultPrefix	Пользовательский префикс для пути или запроса vault.

Parameter	Description
VaultTLSCertFile	Имя файла SSL-сертификата, используемого для аутентификации клиента.
VaultTLSKeyFile	Имя файла закрытого SSL-ключа, используемого для аутентификации клиента.
VaultToken	Токен аутентификации HashiCorp vault.
VaultURL	HTTP[S] URL сервера vault.
VMwareCacheSize	Размер общей памяти для хранения данных VMware.
VMwareFrequency	Задержка в секундах между сбором данных от одной службы VMware.
VMwarePerfFrequency	Задержка в секундах между получением статистики счётчиков производительности от одной службы VMware.
VMwareTimeout	Максимальное количество секунд, в течение которых сборщик vmware будет ждать ответа от службы VMware.
WebDriverURL	HTTP[S] URL интерфейса WebDriver.
WebServiceURL	HTTP[S] URL веб-службы Zabbix в формате <host:port>/report.

Все параметры не являются обязательными, если явно не указано, что параметр обязателен.

Обратите внимание:

- Значения по умолчанию отражают значения демона по умолчанию, 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

AlertScriptsPath

Расположение пользовательских скриптов оповещений (зависит от переменной установки времени компиляции *datadir*).

По умолчанию: `/usr/local/share/zabbix/alertscripts`

AllowRoot

Разрешить серверу работать от имени ’root’. Если отключено и сервер запущен пользователем ’root’, сервер попытается вместо этого переключиться на пользователя ’zabbix’. Не имеет эффекта, если сервер запущен от имени обычного пользователя.

По умолчанию: 0
 Значения: 0 — не разрешать; 1 — разрешать

AllowSoftwareUpdateCheck

Разрешить Zabbix UI получать информацию об обновлениях ПО с zabbix.com.

По умолчанию: 1
 Значения: 0 - не разрешать; 1 - разрешать

AllowUnsupportedDBVersions

Разрешить серверу работать с неподдерживаемыми версиями баз данных.

По умолчанию: 0
 Значения: 0 - не разрешать; 1 - разрешать

CacheSize

Размер кэша конфигурации в байтах. Размер общей памяти для хранения данных об узлах сети, элементах данных и триггерах.

По умолчанию: `32M`
 Диапазон: 128K-64G

CacheUpdateFrequency

Этот параметр определяет, как часто Zabbix будет обновлять кэш конфигурации, в секундах. См. также параметры runtime control.

По умолчанию: 10
 Диапазон: 1-3600

DBHost

Имя узла базы данных.
Для MySQL значение `localhost` или пустая строка приводят к использованию сокета. Для PostgreSQL, если указана пустая строка, используется Unix-domain socket по умолчанию; если указан путь (например, /var/run/pgbouncer), используется указанный Unix-domain socket.

По умолчанию: `localhost`

DBName

Имя базы данных.

Обязательно: Да

DBPassword

Пароль базы данных. Закомментируйте эту строку, если пароль не используется.

DBPort

Порт базы данных, если не используется Unix-domain socket по умолчанию.[3]

По умолчанию для MySQL: 3306

По умолчанию для PostgreSQL: 5432

Диапазон: 1024-65535

DBSchema

Имя схемы базы данных. Используется для PostgreSQL.

DBSocket

Путь к файлу сокета MySQL.[3]

DBUser

Пользователь базы данных.

DBTLSConnect

Установка этого параметра на следующие значения принудительно использует TLS-подключение к базе данных:
required - подключение с использованием TLS
verify_ca - подключение с использованием TLS и проверка сертификата
verify_full - подключение с использованием TLS, проверка сертификата и проверка того, что идентификатор базы данных, указанный DBHost, соответствует ее сертификату

В MySQL, начиная с версии 5.7.11, и PostgreSQL поддерживаются следующие значения: `required`, `verify_ca`, `verify_full`.
В MariaDB, начиная с версии 10.2.6, поддерживаются значения `required` и `verify_full`.
По умолчанию не установлено ни одно значение, и поведение зависит от конфигурации базы данных.

DBTLSCAFile

Полный путь к файлу, содержащему сертификаты CA верхнего уровня для проверки сертификата базы данных.

Обязательно: нет (да, если DBTLSConnect установлен на *verify_ca* или *verify_full*)

DBTLSCertFile

Полный путь к файлу, содержащему сертификат сервера Zabbix для аутентификации в базе данных.

DBTLSKeyFile

Полный путь к файлу, содержащему закрытый ключ для аутентификации в базе данных.

DBTLSCipher

Список шифров шифрования, которые сервер Zabbix разрешает для протоколов TLS вплоть до TLS v1.2. Поддерживается только для MySQL.

DBTLSCipher13

Список наборов шифров шифрования, которые сервер Zabbix разрешает для протокола TLS v1.3. Поддерживается только для MySQL, начиная с версии 8.0.16.

DebugLevel

Укажите уровень отладки:
0 — основная 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 ещё 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, EnableGlobalScripts явно устанавливается в 0 (отключено).

По умолчанию: 1
 Значения: 0 — отключить; 1 — включить

ExportDir

Каталог для экспорта в реальном времени событий, истории и трендов в формате JSON с разделением строками новой строки. Если задан, включает экспорт в реальном времени.

ExportFileSize

Максимальный размер каждого файла экспорта в байтах. Используется для ротации, если задан `ExportDir`.

По умолчанию: 1G
 Диапазон: 1M-1G

ExportType

Список типов сущностей, разделённых запятыми (events, history, trends), для экспорта в реальном времени (по умолчанию — все типы). Допустимо только если задан ExportDir.
Обратите внимание, что если указан ExportType, но не задан ExportDir, это является ошибкой конфигурации, и сервер не запустится.

Пример для экспорта history и trends:

```
ExportType=history,trends
```

Пример только для экспорта events:

```
ExportType=events
```

ExternalScripts

Расположение внешних скриптов (зависит от переменной установки времени компиляции `datadir`).

По умолчанию: `/usr/local/share/zabbix/externalscripts`

Fping6Location

Расположение fping6. Убедитесь, что бинарный файл fping6 принадлежит root и для него установлен флаг SUID. Оставьте пустым (”Fping6Location=”), если ваша утилита fping способна обрабатывать IPv6-адреса.

По умолчанию: `/usr/sbin/fping6`

FpingLocation

Расположение fping. Убедитесь, что бинарный файл fping принадлежит root и для него установлен флаг SUID.

По умолчанию: `/usr/sbin/fping`

FrontendAllowedIP

Разделённый запятыми список IP-адресов или диапазонов CIDR, которым разрешено подключение из веб-интерфейса. Подключение веб-интерфейса будет приниматься только с адресов, перечисленных здесь, если этот параметр задан. 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’ разрешает 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

Имя узла кластера высокой доступности. Если значение пустое, сервер работает в автономном режиме, и создаётся узел с пустым именем.

HistoryCacheSize

Размер кэша истории в байтах. Размер общей памяти для хранения данных истории.

По умолчанию: 16M
 Диапазон: 128K-16G

HistoryIndexCacheSize

Размер кэша индекса истории в байтах. Размер общей памяти для индексирования данных истории, хранящихся в кэше истории. Для кэширования одного элемента данных размер кэша индекса должен составлять примерно 100 байт.

По умолчанию: 4M
 Диапазон: 128K-16G

HistoryStorageDateIndex

Включить предобработку значений истории в хранилище истории, чтобы сохранять значения в разных индексах в зависимости от даты.

По умолчанию: 0
 Значения: 0 — отключено; 1 — включено

HistoryStorageURL

URL HTTP[S] хранилища истории. Этот параметр используется для настройки Elasticsearch.

HistoryStorageTypes

Список типов значений, разделённых запятыми, которые будут отправляться в хранилище истории. Этот параметр используется для настройки Elasticsearch.

По умолчанию: `uint,dbl,str,log,text,json`

HousekeepingFrequency

Этот параметр определяет, как часто Zabbix будет выполнять процедуру очистки в часах. Очистка — это удаление устаревшей информации из базы данных.
Примечание: Чтобы предотвратить перегрузку housekeeper (например, когда периоды хранения истории и трендов значительно уменьшены), за один цикл очистки для каждого элемента данных удаляется не более чем устаревшая информация за 4 × HousekeepingFrequency часов. Таким образом, если HousekeepingFrequency равен 1, за 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 запуска сервера. Таким образом, если HousekeepingFrequency равен 1, самая первая процедура очистки после запуска сервера будет выполнена через 30 минут, а затем будет повторяться с интервалом в один час.
Можно отключить автоматическую очистку, установив HousekeepingFrequency в 0. В этом случае процедура очистки может быть запущена только с помощью опции управления во время выполнения *housekeeper_execute*, а период устаревшей информации, удаляемой за один цикл очистки, составляет 4-кратный период с момента последнего цикла очистки, но не менее 4 часов и не более 4 дней.
См. также опции управления во время выполнения и подробности о процедуре очистки.

По умолчанию: 1
 Диапазон: 0-24

Include

Вы можете включить отдельные файлы или все файлы в каталоге в файл конфигурации. Чтобы включать только подходящие файлы в указанном каталоге, для сопоставления с шаблоном поддерживается символ подстановки «звёздочка». См. специальные примечания об ограничениях.

Пример:

```
Include=/absolute/path/to/config/files/*.conf
```

JavaGateway

IP-адрес (или имя узла) Zabbix Java gateway. Требуется только в том случае, если запущены Java poller'ы.

JavaGatewayPort

Порт, который прослушивает шлюз Zabbix Java.

По умолчанию: 10052
 Диапазон: 1024-32767

ListenBacklog

Максимальное количество ожидающих соединений в очереди TCP.
Значение по умолчанию представляет собой жестко закодированную константу, которая зависит от системы.
Максимальное поддерживаемое значение зависит от системы, слишком высокие значения могут быть автоматически усечены до «максимум, определяемый реализацией».

По умолчанию: `SOMAXCONN`
 Диапазон: 0 – INT_MAX

ListenIP

Список IP-адресов, разделенных запятыми, которые должен прослушивать траппер.
Траппер будет прослушивать все сетевые интерфейсы, если этот параметр отсутствует.

По умолчанию: `0.0.0.0`

ListenPort

Порт прослушивания для ловушки.

По умолчанию: 10051
 Диапазон: 1024-32767

LoadModule

Модуль для загрузки при запуске сервера. Модули используются для расширения функциональности сервера. Модуль должен находиться в каталоге, указанном в LoadModulePath, либо перед именем модуля должен быть указан путь. Если указанный перед именем путь является абсолютным (начинается с '/'), то LoadModulePath игнорируется.
Форматы:
LoadModu
указывать несколько параметров LoadModule.

LoadModulePath

Полный путь к расположению модулей сервера. Значение по умолчанию зависит от параметров компиляции.

LogFile

Имя файла журнала.

Обязательно: Да, если LogType установлен на *file*; в противном случае нет

LogFileSize

Максимальный размер файла журнала в МБ.
0 — отключить автоматическую ротацию журнала.
Примечание: Если достигнут предельный размер файла журнала и ротация файла по какой-либо причине завершается неудачей, существующий файл журнала усекается и начинается заново.

По умолчанию: 1
 Диапазон: 0-1024
 Обязательный: Да, если LogType установлен в *file*; в противном случае — нет

LogSlowQueries

Определяет, сколько времени может занимать запрос к базе данных, прежде чем он будет записан в журнал, в миллисекундах.
0 — не записывать медленные запросы в журнал.
Этот параметр становится активным, начиная с DebugLevel=3.

По умолчанию: 0
 Диапазон: 0-3600000

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`

MaxConcurrentChecksPerPoller

Максимальное количество асинхронных проверок, которые могут одновременно выполняться каждым опрашивателем HTTP-агента, опрашивателем агента или опрашивателем SNMP. См. StartHTTPAgentPollers, StartAgentPollers и StartSNMP-Pollers.

По умолчанию: 1000
 Диапазон: 1-1000

MaxHousekeeperDelete

Таблица "housekeeper" содержит "tasks" для процедуры очистки в формате: `[housekeeperid]`, `[object]`, `[objectid]`.
 Не более `MaxHousekeeperDelete` строк из таблиц `history`, `history_str`, `history_log`, `history_uint`, `history_text`, `history_bin`, `history_json`, `trends`, `trends_uint` и `problem` будет удалено за одну задачу в одном цикле очистки.
 Если установлено значение 0, то ограничение не используется вообще. В этом случае вы должны понимать, что делаете, чтобы не перегрузить базу данных.[2]
Этот параметр применяется только к удалению данных, оставшихся после уже удалённых элементов данных.
 См. также подробности о процедуре очистки.

По умолчанию: 5000
 Диапазон: 0-1000000

NodeAddress

IP-адрес или имя узла с необязательным портом для переопределения способа подключения веб-интерфейса к серверу.
Формат: <address>[:<port>]

Если IP-адрес или имя узла 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 может быть переопределен адресом, указанным в конфигурации веб-интерфейса.

См. также: параметр HANodeName; Включение высокой доступности.

По умолчанию: `localhost:10051`

PidFile

Имя PID-файла.

По умолчанию: `/tmp/zabbix_server.pid`

ProblemHousekeepingFrequency

Определяет, как часто Zabbix будет удалять проблемы для удалённых триггеров, в секундах.
 См. также подробную информацию о процедуре очистки истории.

По умолчанию: 60
 Диапазон: 1-3600

ProxyConfigFrequency

Определяет, как часто сервер Zabbix отправляет данные конфигурации прокси Zabbix, в секундах. Используется только для прокси в пассивном режиме.

По умолчанию: 10
 Диапазон: 1-604800

ProxyDataFrequency

Определяет, как часто сервер Zabbix запрашивает данные истории у прокси Zabbix, в секундах. Используется только для прокси в пассивном режиме.

По умолчанию: 1
 Диапазон: 1-3600

ServiceManagerSyncFrequency

Определяет, как часто Zabbix будет синхронизировать конфигурацию менеджера служб, в секундах.

По умолчанию: 60
 Диапазон: 1-3600

SMSDevices

Список файлов модемов, разделённых запятыми, которым разрешено использоваться сервером Zabbix.
Отправка SMS невозможна, если этот параметр не задан.

Пример:

```
SMSDevices=/dev/ttyUSB0,/dev/ttyUSB1
```

SNMPTrapperFile

Временный файл, используемый для передачи данных от демона SNMP trap на сервер.
Должен совпадать с указанным в файле конфигурации zabbix_trap_receiver.pl или SNMPTT.

По умолчанию: `/tmp/zabbix_traps.tmp`

SocketDir

Каталог для хранения IPC-сокетов, используемых внутренними службами Zabbix.

По умолчанию: `/tmp`

SourceIP

IP-адрес источника для:

- исходящих подключений к Zabbix прокси и Zabbix агенту
- подключений без использования агента (VMware, SSH, JMX, SNMP, Telnet и простые проверки)
- подключений HTTP-агента
- HTTP-запросов JavaScript для элементов данных-скриптов
- HTTP-запросов JavaScript для предобработки
- отправки уведомлений по электронной почте (подключения к SMTP-серверу)
- уведомлений вебхука (HTTP-подключения JavaScript)
- подключений к Vault

SSHKeyLocation

Расположение открытых и закрытых ключей для проверок и действий SSH.

SSLCertLocation

Расположение файлов сертификатов клиента SSL для аутентификации клиента.
Этот параметр используется только в веб-мониторинге.

SSLKeyLocation

Расположение файлов закрытого ключа SSL для аутентификации клиента.
Этот параметр используется только в веб-мониторинге.

SSLCALocation

Переопределяет расположение файлов центра сертификации (CA) для проверки SSL-сертификата сервера. Если не задано, будет использоваться общесистемный каталог.
Обратите внимание, что значение этого параметра будет

установлено как параметр libcurl CURLOPT_CAPATH. Для версий libcurl до 7.42.0 это действует только в том случае, если libcurl был скомпилирован с использованием OpenSSL. Для получения дополнительной информации смотрите веб-страницу cURL.
Этот параметр используется в веб-мониторинге и при SMTP-аутентификации.

StartAgentPollers

Количество заранее созданных экземпляров опросчиков агента Zabbix. См. MaxConcurrentChecksPerPoller.

По умолчанию: 1
 Диапазон: 0-1000

StartAlerters

Количество предварительно созданных экземпляров оповещателей.

По умолчанию: 3
 Диапазон: 1-100

StartBrowserPollers

Количество предварительно созданных экземпляров опросчиков элементов данных браузера.

По умолчанию: 1
 Диапазон: 0-1000

StartConnectors

Количество предварительно созданных экземпляров рабочих процессов коннектора. Процесс менеджера коннекторов запускается автоматически при запуске рабочего процесса коннектора.

По умолчанию: 0
 Диапазон: 0-1000

StartDBSyncers

Количество предварительно созданных экземпляров синхронизаторов истории.
Примечание: Будьте осторожны при изменении этого значения, его увеличение может принести больше вреда, чем пользы. Ориентировочно, значения по умолчанию должно быть достаточно для обработки до 4000 NVPS.

По умолчанию: 4
 Диапазон: 1-100

StartDiscoverers

Количество предварительно созданных экземпляров обработчиков обнаружения[1].

По умолчанию: 5
 Диапазон: 0-1000

StartEscalators

Количество предварительно созданных экземпляров escalators.

По умолчанию: 1
 Диапазон: 1-100

StartHistoryPollers

Количество заранее созданных экземпляров history pollers.
Требуется только для вычисляемых проверок.

По умолчанию: 5
 Диапазон: 0-1000

StartHTTPAgentPollers

Количество предварительно созданных экземпляров HTTP агент опросчиков. См. MaxConcurrentChecksPerPoller.

По умолчанию: 1
 Диапазон: 0-1000

StartHTTPPollers

Количество предварительно созданных экземпляров HTTP pollers[1].

По умолчанию: 1
 Диапазон: 0-1000

StartIPMIPollers

Количество предварительно созданных экземпляров IPMI pollers.

По умолчанию: 0
 Диапазон: 0-1000

StartJavaPollers

Количество предварительно созданных экземпляров Java pollers[1].

По умолчанию: 0
 Диапазон: 0-1000

StartLLDProcessors

Количество заранее созданных экземпляров обработчиков низкоуровневого обнаружения (LLD)[1].
Процесс менеджера LLD автоматически запускается при запуске обработчика LLD.

По умолчанию: 2
 Диапазон: 1-100

StartODBCPollers

Количество предварительно созданных экземпляров опросчиков ODBC[1].

По умолчанию: 1
 Диапазон: 0-1000

StartPingers

Количество предварительно созданных экземпляров ICMP pingers[1].

По умолчанию: 1
 Диапазон: 0-1000

StartPollersUnreachable

Количество предварительно созданных экземпляров опросчиков для недоступных узлов сети (включая IPMI и Java)[1].
Если запущены обычные, IPMI или Java-опросчики, должен быть запущен как минимум один опросчик для недоступных узлов сети.

По умолчанию: 1
 Диапазон: 0-1000

StartPollers

Количество предварительно созданных экземпляров pollers[1].

По умолчанию: 5
 Диапазон: 0-1000

StartPreprocessors

Количество предварительно запущенных экземпляров потоков worker[1] предобработки должно быть не меньше количества доступных ядер CPU. Следует задать большее количество worker, если предобработка не ограничена CPU и содержит много сетевых запросов.

По умолчанию: 16
 Диапазон: 1-1000

StartProxyPollers

Количество предварительно созданных экземпляров опросчиков для пассивных прокси[1].

По умолчанию: 1
 Диапазон: 0-250

StartReportWriters

Количество предварительно созданных экземпляров процессов записи отчётов.
Если установлено значение 0, генерация отчётов по расписанию отключена.
Процесс менеджера отчётов запускается автоматически при запуске процесса записи отчётов.

По умолчанию: 0
 Диапазон: 0-100

StartSNMPPollers

Количество предварительно созданных экземпляров SNMP pollers. См. MaxConcurrentChecksPerPoller.

По умолчанию: 1
 Диапазон: 0-1000

StartSNMPTrapper

Если установлено значение 1, будет запущен процесс SNMP trapper.

По умолчанию: 0
 Диапазон: 0-1

StartTimers

Количество предварительно созданных экземпляров timers.
Timers обрабатывают периоды обслуживания.

По умолчанию: 1
 Диапазон: 1-1000

StartTrappers

Количество предварительно созданных экземпляров trappers[1].
Trappers принимают входящие соединения от Zabbix sender, активных агентов и активных прокси.

По умолчанию: 5
 Диапазон: 0-1000

StartVMwareCollectors

Количество предварительно созданных экземпляров сборщика VMware.

По умолчанию: 0
 Диапазон: 0-250

StatsAllowedIP

Список IP-адресов, разделённых запятыми, при необходимости в нотации CIDR, или DNS-имён внешних экземпляров Zabbix. Запрос статистики будет принят только с адресов, перечисленных здесь. Если этот параметр не задан, запросы статистики приниматься не будут.
Если поддержка IPv6 включена, тогда '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
```

Timeout

Указывает, как долго ждать (в секундах) установления соединения и обмена данными с Zabbix прокси, агентом, веб-службой и устаревшими SNMP-проверками (одиночный номер OID или строка).

Этот параметр определяет длительность различных операций обмена данными:

- выполнение удалённых команд на Zabbix агенте
- выполнение команд SSH/Telnet
- запросы к Zabbix web service
- тайм-аут обмена данными для тестового запроса типа оповещения и метода `history.push`
- перепланирование элементов данных, когда интерфейс IPMI становится недоступным
- отправка ответа Zabbix прокси при невозможности обмена данными из-за проблем с правами доступа или шифрованием
- предельное время ожидания для асинхронных IPC-сокетов и параметров управления во время выполнения
- JMX-соединения
- получение статистики с удалённого Zabbix прокси или сервера
- отправка ответов в Zabbix веб-интерфейс
- DNS-запросы асинхронного поллера
- ответ для heartbeat активной проверки
- получение данных (значений) Zabbix агента от активных агентов
- получение данных от Zabbix sender
- ответ, когда trapper не удаётся обработать запрос
- отправка списка активных проверок Zabbix агенту

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 отдельного элемента данных). Например, элементы данных SNMP `walk[OID]` и `get[OID]` используют тайм-аут, настроенный в веб-интерфейсе; устаревшие SNMP-проверки по-прежнему используют значение тайм-аута сервера.

По умолчанию: 3
 Диапазон: 1-30

TLSCAFile

Полный путь к файлу, содержащему сертификаты CA верхнего уровня для проверки сертификатов одноранговых узлов сети, используемые для зашифрованной связи между компонентами Zabbix.

TLSCertFile

Полный путь к файлу, содержащему сертификат сервера или цепочку сертификатов, используемых для зашифрованной связи между компонентами Zabbix.

TLSCipherAll

Строка приоритетов GnuTLS или строка 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

Строка шифров для OpenSSL 1.1.1 или новее 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

Строка приоритетов GnuTLS или строка 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

Строка 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

Строка приоритетов GnuTLS или строка 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

Строка 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
```

TLSCRLFile

Полный путь к файлу, содержащему отозванные сертификаты. Этот параметр используется для шифрованного обмена данными между компонентами Zabbix.

TLSFrontendAccept

Какие входящие соединения принимать от веб-интерфейса.

Можно указать несколько значений, разделённых запятой:

- unencrypted — принимать соединения без шифрования.
- cert — принимать соединения, защищённые TLS и сертификатом.

По умолчанию: `unencrypted`

TLSFrontendCertIssuer

Разрешённый издатель сертификата веб-интерфейса.

TLSFrontendCertSubject

Разрешённый субъект сертификата веб-интерфейса.

TLSKeyFile

Полный путь к файлу, содержащему закрытый ключ сервера, используемый для шифрованного обмена данными между компонентами Zabbix.

TLSListen

Управляет TLS на сокете trapper.

Поддерживаемые значения:

- required - принимать только TLS-соединения

TmpDir

Временный каталог.

По умолчанию: `/tmp`

TrapperTimeout

Указывает тайм-аут в секундах для:

- получения исторических данных от Zabbix прокси
- отправки данных конфигурации на Zabbix прокси
- выполнения глобального скрипта или удалённой команды на Zabbix сервере

По умолчанию: 300
 Диапазон: 1-300

TrendCacheSize

Размер кэша трендов в байтах.
Размер общей памяти для хранения данных трендов.

По умолчанию: `4M`
 Диапазон: 128K-16G

TrendFunctionCacheSize

Размер кэша функций трендов в байтах.
Размер общей памяти для кэширования вычисленных данных функций трендов.

По умолчанию: `4M`
 Диапазон: 128K-2G

UnavailableDelay

Определяет, как часто узел сети проверяется на доступность в течение периода недоступности, в секундах.

По умолчанию: 60
 Диапазон: 1-3600

UnreachableDelay

Определяет, как часто узел сети проверяется на доступность в течение периода недоступности, в секундах.

По умолчанию: 15
 Диапазон: 1-3600

UnreachablePeriod

Определяет, через сколько секунд недоступности узел сети считается недоступным.

По умолчанию: 45
 Диапазон: 1-3600

Пользователь

Сбросить привилегии до конкретного существующего пользователя в системе.
Имеет эффект только при запуске от имени 'root' и если AllowRoot отключён.

По умолчанию: `zabbix`

ValueCacheSize

Размер кэша значений истории в байтах.
Размер общей памяти для кэширования запросов данных истории элементов данных.
Значение 0 отключает кэш значений (не рекомендуется).
Когда у кэша значений заканчивается общая память, в журнал сервера каждые 5 минут записывается предупреждающее сообщение.

По умолчанию: `8M`
 Диапазон: 0,128K-64G

Хранилище секретов

Указывает поставщика хранилища секретов:
HashiCorp — HashiCorp KV Secrets Engine версии 2
CyberArk — CyberArk Central Credential Provider
Должен соответствовать поставщику хранилища секретов, заданному в веб-интерфейсе.

По умолчанию: `HashiCorp`

VaultDBPath

Путь или запрос Vault, в зависимости от Vault, откуда будут получены учетные данные для базы данных по ключам.

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

Пользовательский префикс для пути или запроса Vault, в зависимости от Vault. Если не указан, будут использованы наиболее подходящие значения по умолчанию.
Обратите внимание, что для HashiCorp после точки монтирования автоматически добавляется data, если VaultPrefix не указан.

Пример префикса для Hashicorp:

```
v1/secret/data/zabbix/
```

Пример префикса для Cyberark:

```
/AIMWebService/api/Accounts?
```

VaultTLSCertFile

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, задаётся параметром конфигурации SSLCertLocation.
Этот параметр можно не указывать, но он рекомендуется для хранилища CyberArkCCP vault.

VaultTLSKeyFile

Имя файла закрытого ключа SSL, используемого для аутентификации клиента.
 Файл закрытого ключа должен быть в формате PEM1.
 Каталог, содержащий этот файл, задаётся параметром конфигурации SSLKeyLocation.
Этот параметр можно не указывать, но это рекомендуется для хранилища CyberArkCCP.

VaultToken

Токен аутентификации HashiCorp Vault, который должен быть сгенерирован исключительно для сервера Zabbix с правами только на чтение для путей, указанных в макросах Vault, и правами только на чтение для пути, указанного в необязательном параметре конфигурации VaultDBPath.
Определение VaultToken и переменной окружения VAULT_TOKEN одновременно является ошибкой.

Обязательно: Да, если для Vault установлено значение *HashiCorp*; в противном случае нет

VaultURL

URL HTTP[S] сервера хранилища. Если SSLCALocation не указан, будет использоваться общесистемный каталог сертификатов CA.

По умолчанию: `https://127.0.0.1:8200`

VMwareCacheSize

Размер общей памяти для хранения данных VMware.
Для мониторинга использования кэша VMware можно использовать внутреннюю проверку VMware zabbix[vmware,buffer,...] (см. Внутренние проверки).
Обратите внимание, что общая память не выделяется, если не настроен запуск ни одного экземпляра сборщика vmware.

По умолчанию: 8M
 Диапазон: 256K-2G

VMwareFrequency

Задержка в секундах между сбором данных от отдельного сервиса VMware.
Это значение следует установить равным наименьшему интервалу обновления любого элемента данных мониторинга VMware.

По умолчанию: 60
 Диапазон: 10-86400

VMwarePerfFrequency

Задержка в секундах между получением статистики счётчиков производительности от одного сервиса VMware. Эта задержка должна быть установлена равной наименьшему интервалу обновления любого элемента данных мониторинга VMware item, который использует счётчики производительности VMware.

По умолчанию: 60
 Диапазон: 10-86400

VMwareTimeout

Максимальное количество секунд, в течение которых сборщик VMware будет ожидать ответа от службы VMware (гипервизор vCenter или ESX).

По умолчанию: 10
 Диапазон: 1-300

WebServiceURL

HTTP[S] URL веб-сервиса Zabbix в формате `http[s]://host:port/report`.

Пример:

```
WebServiceURL=http://localhost:10053/report
```

Примечание: схему (`http://`) можно опустить только для соединений без TLS (HTTP); если настроен TLS, необходимо использовать `https://`.

WebDriverURL

HTTP[S] URL интерфейса WebDriver.

Пример (используется с автономным сервером Selenium WebDriver):

```
WebDriverURL=http://localhost:4444
```

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 файловых дескрипторов на процесс для preprocessing manager.

Warning:
Исчерпание лимита файловых дескрипторов приведет к остановке сервера Zabbix, как правило, вскоре после запуска, хотя иногда это может занять больше времени. Чтобы избежать таких проблем, проверьте файл конфигурации сервера Zabbix, чтобы оптимизировать количество параллельных проверок и процессов. Кроме того, при необходимости убедитесь, что лимит файловых дескрипторов установлен достаточно высоким, проверив и скорректировав системные ограничения.

[2] При удалении большого количества элементов данных возрастает нагрузка на базу данных, поскольку housekeeper должен удалить все исторические данные, которые были у этих элементов данных. Например, если нужно удалить только 1 прототип элемента данных из шаблона, но этот шаблон связан с 50 узлами сети, и для каждого узла сети прототип разворачивается в 100 реальных элементов данных, то всего потребуется удалить 5000 элементов данных (1*50*100). Если для MaxHousekeeperDelete установлено 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.

[3] DBSocket и DBPort являются взаимоисключающими в конфигурации сервера. Укажите только один из них или оставьте оба неопределенными.

2 Zabbix прокси

Обзор

В этом разделе перечислены параметры, поддерживаемые файлом конфигурации прокси Zabbix (zabbix_proxy.conf).

Параметры перечислены без дополнительной информации. Нажмите на параметр, чтобы увидеть полные сведения.

Параметр	Описание
AllowRoot	Разрешить запуск прокси от имени 'root'.
AllowUnsupportedDBVersions	Разрешить прокси работать с неподдерживаемыми версиями базы данных.
CacheSize	Размер кэша конфигурации.
ConfigFrequency	Этот параметр **устарел** (вместо него используйте ProxyConfigFrequency). Как часто прокси получает данные конфигурации от сервера Zabbix, в секундах.
DataSenderFrequency	Прокси будет отправлять собранные данные на сервер каждые N секунд.
DBHost	Имя узла базы данных.
DBName	Имя базы данных или путь к файлу базы данных для SQLite3.

Параметр	Описание
DBPassword	Пароль базы данных.
DBPort	Порт базы данных, если не используется локальный сокет.
DBSchema	Имя схемы базы данных. Используется для PostgreSQL.
DBSocket	Путь к файлу сокета MySQL.
DBUser	Пользователь базы данных.
DBTLSConnect	Установка этой опции в указанное значение принудительно включает использование TLS-соединения с базой данных.
DBTLSCAFile	Полный путь к файлу, содержащему сертификаты корневых CA для проверки сертификата базы данных.
DBTLSCertFile	Полный путь к файлу, содержащему сертификат прокси Zabbix для аутентификации в базе данных.
DBTLSKeyFile	Полный путь к файлу, содержащему закрытый ключ для аутентификации в базе данных.
DBTLSCipher	Список шифров шифрования, которые прокси Zabbix разрешает для протоколов TLS вплоть до TLS v1.2. Поддерживается только для MySQL.
DBTLSCipher13	Список наборов шифров, которые прокси Zabbix разрешает для протокола TLS v1.3. Поддерживается только для MySQL, начиная с версии 8.0.16.
DebugLevel	Уровень отладки.
EnableRemoteCommands	Разрешены ли удалённые команды от сервера Zabbix.
ExternalScripts	Расположение внешних скриптов.
Fping6Location	Расположение fping6.
FpingLocation	Расположение fping.
HistoryCacheSize	Размер кэша истории.
HistoryIndexCacheSize	Размер кэша индекса истории.
Hostname	Уникальное имя прокси с учётом регистра.
HostnameItem	Элемент данных, используемый для установки Hostname, если он не определён.
HousekeepingFrequency	Как часто Zabbix будет выполнять процедуру очистки, в часах.
Include	В файл конфигурации можно включать отдельные файлы или все файлы из каталога.
JavaGateway	IP-адрес (или имя узла) Zabbix Java gateway.
JavaGatewayPort	Порт, на котором прослушивает Zabbix Java gateway.
ListenBacklog	Максимальное количество ожидающих соединений в очереди TCP.
ListenIP	Список IP-адресов, разделённых запятыми, на которых trapper должен прослушивать соединения.
ListenPort	Порт прослушивания для trapper.
LoadModule	Модуль, загружаемый при запуске прокси.
LoadModulePath	Полный путь к расположению модулей прокси.
LogFile	Имя файла журнала.
LogFileSize	Максимальный размер файла журнала.
LogRemoteCommands	Включить журналирование выполняемых shell-команд как предупреждений.
LogSlowQueries	Сколько времени может выполняться запрос к базе данных, прежде чем он будет записан в журнал.
LogType	Тип вывода журнала.
MaxConcurrentChecksPerPoller	Максимальное количество асинхронных проверок, которые могут одновременно выполняться каждым HTTP агент poller, агент poller или SNMP poller.
PidFile	Имя PID-файла.
ProxyBufferMode	Определяет механизм хранения данных истории, обнаружения и авторегистрации (disk/memory/hybrid).
ProxyConfigFrequency	Как часто прокси получает данные конфигурации от сервера Zabbix, в секундах.
ProxyLocalBuffer	Прокси будет хранить данные локально N часов, даже если эти данные уже были синхронизированы с сервером.
ProxyMemoryBufferAge	Максимальный возраст данных в буфере памяти прокси, в секундах.
ProxyMemoryBufferSize	Размер кэша общей памяти для собранных данных истории, обнаружения и авторегистрации.
ProxyMode	Режим работы прокси (active/passive).
ProxyOfflineBuffer	Прокси будет хранить данные N часов в случае отсутствия соединения с сервером Zabbix.
Server	Если ProxyMode установлен в active mode: IP-адрес или DNS-имя сервера Zabbix (address:port) либо кластер (address:port;address2:port), откуда получать данные конфигурации и куда отправлять данные. Если ProxyMode установлен в passive mode: список IP-адресов сервера Zabbix, разделённых запятыми, при необходимости в нотации CIDR, либо DNS-имён.
SNMPTrapperFile	Временный файл, используемый для передачи данных от демона SNMP trap прокси.
SocketDir	Каталог для хранения IPC-сокетов, используемых внутренними службами Zabbix.

Параметр	Описание
SourceIP	Исходный IP-адрес.
SSHKeyLocation	Расположение открытых и закрытых ключей для SSH-проверок и действий.
SSLCertLocation	Расположение файлов SSL-сертификатов клиента для аутентификации клиента.
SSLKeyLocation	Расположение файлов закрытых ключей SSL для аутентификации клиента.
SSLCALocation	Переопределяет расположение файлов центра сертификации (CA) для проверки SSL-сертификата сервера.
StartAgentPollers	Количество заранее созданных экземпляров асинхронных poller'ов агента Zabbix.
StartBrowserPollers	Количество заранее созданных экземпляров poller'ов браузерных элементов данных.
StartDBSyncers	Количество заранее созданных экземпляров синхронизаторов истории.
StartDiscoverers	Количество заранее созданных экземпляров процессов обнаружения.
StartHTTPAgentPollers	Количество заранее созданных экземпляров асинхронных HTTP агент poller'ов.
StartHTTPPollers	Количество заранее созданных экземпляров HTTP poller'ов.
StartIPMIPollers	Количество заранее созданных экземпляров IPMI poller'ов.
StartJavaPollers	Количество заранее созданных экземпляров Java poller'ов.
StartODBCPollers	Количество заранее созданных экземпляров ODBC poller'ов.
StartPingers	Количество заранее созданных экземпляров ICMP pinger'ов.
StartPollersUnreachable	Количество заранее созданных экземпляров poller'ов для недоступных узлов сети (включая IPMI и Java).
StartPollers	Количество заранее созданных экземпляров poller'ов.
StartPreprocessors	Количество заранее запущенных экземпляров процессов предобработки.
StartSNMPPollers	Количество заранее созданных экземпляров асинхронных SNMP poller'ов.
StartSNMPTrapper	Если установлено значение 1, будет запущен процесс SNMP trapper.
StartTrappers	Количество заранее созданных экземпляров trapper'ов.
StartVMwareCollectors	Количество заранее созданных экземпляров сборщиков VMware.
StatsAllowedIP	Список IP-адресов, разделённых запятыми, при необходимости в нотации CIDR, либо DNS-имён внешних экземпляров Zabbix. Запрос статистики будет принят только с адресов, перечисленных здесь.
Timeout	Определяет, как долго ждать (в секундах) установления соединения и обмена данными с прокси Zabbix, агентом, веб-службой, а также для SNMP-проверок (кроме элементов данных SNMP `walk[OID]` и `get[OID]`).
TLSAccept	Какие входящие соединения от сервера Zabbix принимать.
TLSCAFile	Полный путь к файлу, содержащему сертификаты корневых CA для проверки сертификата удалённой стороны, используемому для шифрованного взаимодействия между компонентами Zabbix.
TLSCertFile	Полный путь к файлу, содержащему сертификат сервера или цепочку сертификатов, используемому для шифрованного взаимодействия между компонентами Zabbix.
TLSCipherAll	Строка приоритетов GnuTLS или строка шифров OpenSSL (TLS 1.2). Переопределяет критерии выбора наборов шифров по умолчанию для шифрования на основе сертификатов и PSK.
TLSCipherAll13	Строка шифров для OpenSSL 1.1.1 или новее в TLS 1.3. Переопределяет критерии выбора наборов шифров по умолчанию для шифрования на основе сертификатов и PSK.
TLSCipherCert	Строка приоритетов GnuTLS или строка шифров OpenSSL (TLS 1.2). Переопределяет критерии выбора наборов шифров по умолчанию для шифрования на основе сертификатов.
TLSCipherCert13	Строка шифров для OpenSSL 1.1.1 или новее в TLS 1.3. Переопределяет критерии выбора наборов шифров по умолчанию для шифрования на основе сертификатов.
TLSCipherPSK	Строка приоритетов GnuTLS или строка шифров OpenSSL (TLS 1.2). Переопределяет критерии выбора наборов шифров по умолчанию для шифрования на основе PSK.
TLSCipherPSK13	Строка шифров для OpenSSL 1.1.1 или новее в TLS 1.3. Переопределяет критерии выбора наборов шифров по умолчанию для шифрования на основе PSK.
TLSConnect	Как прокси должен подключаться к серверу Zabbix.
TLSCRLFile	Полный путь к файлу, содержащему отозванные сертификаты. Этот параметр используется для шифрованного взаимодействия между компонентами Zabbix.
TLSKeyFile	Полный путь к файлу, содержащему закрытый ключ прокси, используемому для шифрованного взаимодействия между компонентами Zabbix.
TLSListen	Управляет TLS на сокете trapper.
TLSPSKFile	Полный путь к файлу, содержащему предварительно разделяемый ключ прокси, используемый для шифрованного взаимодействия с сервером Zabbix.
TLSPSKIdentity	Строка идентификатора предварительно разделяемого ключа, используемая для шифрованного взаимодействия с сервером Zabbix.
TLSServerCertIssuer	Разрешённый издатель сертификата сервера.

Параметр	Описание
TLSServerCertSubject	Разрешённый субъект сертификата сервера.
TmpDir	Временный каталог.
TrapperTimeout	Определяет тайм-аут в секундах для: - получения данных конфигурации от сервера Zabbix; - выполнения глобального скрипта или удалённой команды.
UnavailableDelay	Как часто узел сети проверяется на доступность в период недоступности.
UnreachableDelay	Как часто узел сети проверяется на доступность в период недосягаемости.
UnreachablePeriod	Через сколько секунд недосягаемости считать узел сети недоступным.
User	Сбросить привилегии до указанного существующего пользователя в системе.
Vault	Провайдер vault.
VaultDBPath	Расположение, откуда учётные данные базы данных должны извлекаться по ключам.
VaultPrefix	Пользовательский префикс для пути или запроса vault.
VaultTLSCertFile	Имя файла SSL-сертификата, используемого для аутентификации клиента.
VaultTLSKeyFile	Имя файла закрытого ключа SSL, используемого для аутентификации клиента.
VaultToken	Токен аутентификации HashiCorp vault.
VaultURL	HTTP[S] URL сервера vault.
VMwareCacheSize	Размер общей памяти для хранения данных VMware.
VMwareFrequency	Задержка в секундах между сбором данных от одной службы VMware.
VMwarePerfFrequency	Задержка в секундах между получением статистики счётчиков производительности от одной службы VMware.
VMwareTimeout	Максимальное количество секунд, в течение которых сборщик vmware будет ждать ответа от службы VMware.
WebDriverURL	HTTP[S] URL интерфейса WebDriver.

Все параметры являются необязательными, если явно не указано, что параметр обязателен.

Обратите внимание:

- Значения по умолчанию отражают значения демона по умолчанию, 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

AllowRoot

Разрешить прокси работать от имени ’root’. Если отключено и прокси запущен от имени ’root’, прокси попытается вместо этого переключиться на пользователя ’zabbix’. Не имеет эффекта, если запущен от имени обычного пользователя.

По умолчанию: 0
 Значения: 0 — не разрешать; 1 — разрешать

AllowUnsupportedDBVersions

Разрешить прокси работать с неподдерживаемыми версиями базы данных.

По умолчанию: 0
 Значения: 0 — не разрешать; 1 — разрешать

CacheSize

Размер кэша конфигурации в байтах. Размер общей памяти для хранения данных узлов сети и элементов данных.

По умолчанию: 32M
 Диапазон: 128K-64G

ConfigFrequency

Этот параметр **устарел** (используйте ProxyConfigFrequency вместо него).
Как часто прокси получает данные конфигурации от сервера Zabbix, в секундах.
Параметр активного прокси. Игнорируется для пассивных прокси (см. параметр ProxyMode).

По умолчанию: 3600
 Диапазон: 1-604800

DataSenderFrequency

Прокси будет отправлять собранные данные на сервер каждые N секунд. Обратите внимание, что активный прокси по-прежнему будет опрашивать Zabbix server каждую секунду на предмет задач удалённых команд.
Параметр активного прокси. Игнорируется для пассивных прокси (см. параметр ProxyMode).

По умолчанию: 1
 Диапазон: 1-3600

DBHost

Имя узла базы данных.
Для MySQL значение `localhost` или пустая строка приводят к использованию сокета. Для PostgreSQL только пустая строка приводит к попытке использовать сокет.

По умолчанию: `localhost`

DBName

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.

Обязательно: Да

DBPassword

Пароль базы данных. Закомментируйте эту строку, если пароль не используется. Игнорируется для SQLite.

DBPort

Порт базы данных при использовании не локального сокета.[1] Игнорируется для SQLite.

По умолчанию для MySQL: 3306

По умолчанию для PostgreSQL: 5432

Диапазон: 1024-65535

DBSchema

Имя схемы базы данных. Используется для PostgreSQL.

DBSocket

Путь к файлу сокета MySQL.[1]
Порт базы данных, если не используется локальный сокет. Игнорируется для SQLite.

По умолчанию: 3306

DBUser

Пользователь базы данных. Игнорируется для SQLite.

DBTLSConnect

Установка этой опции принудительно включает использование TLS-соединения 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, а также проверять, что идентификатор базы данных, указанный в DBHost, соответствует её сертификату
В MySQL, начиная с версии 5.7.11, и в 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

Полный путь к файлу, содержащему сертификаты корневого(ых) центра(ов) сертификации (CA) для проверки сертификата базы данных.

Обязательно: нет (да, если DBTLSConnect установлено в *verify_ca* или *verify_full*)

DBTLSCertFile

Полный путь к файлу, содержащему сертификат прокси Zabbix для аутентификации в базе данных.

DBTLSKeyFile

Полный путь к файлу, содержащему закрытый ключ для аутентификации в базе данных.

DBTLSCipher

Список шифров шифрования, которые прокси Zabbix разрешает для протоколов TLS вплоть до TLS v1.2. Поддерживается только для MySQL.

DBTLSCipher13

Список наборов шифров шифрования, которые прокси Zabbix разрешает для протокола TLS v1.3. Поддерживается только для MySQL, начиная с версии 8.0.16.

DebugLevel

Укажите уровень отладки:
0 — основная 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 ещё больше информации).

По умолчанию: 3
 Диапазон: 0-5

EnableRemoteCommands

Разрешены ли удалённые команды от сервера Zabbix.

По умолчанию: 0
 Значения: 0 — не разрешены; 1 — разрешены

ExternalScripts

Расположение внешних скриптов (зависит от переменной установки времени компиляции `datadir`).

По умолчанию: `/usr/local/share/zabbix/externalscripts`

Fping6Location

Расположение fping6. Убедитесь, что бинарный файл fping6 принадлежит root и для него установлен флаг SUID. Оставьте пустым (”Fping6Location=”), если ваша утилита fping способна обрабатывать IPv6-адреса.

По умолчанию: `/usr/sbin/fping6`

FpingLocation

Расположение fping. Убедитесь, что бинарный файл fping принадлежит root и для него установлен флаг SUID.

По умолчанию: `/usr/sbin/fping`

HistoryCacheSize

Размер кэша истории в байтах. Размер общей памяти для хранения данных истории.

По умолчанию: `16M`
 Диапазон: 128K-16G

HistoryIndexCacheSize

Размер кэша индекса истории в байтах. Размер общей памяти для индексирования данных истории, хранящихся в кэше истории. Для кэширования одного элемента данных размер кэша индекса должен составлять примерно 100 байт.

По умолчанию: `4M`
 Диапазон: 128K-16G

Hostname

Уникальное имя прокси с учетом регистра. Убедитесь, что имя прокси известно серверу.
Допустимые символы: буквы и цифры, ’.’, ’ ’, ’_’ и ’-’. Максимальная длина: 128

По умолчанию: задается параметром HostnameItem

HostnameItem

Элемент данных, используемый для задания Hostname, если он не определён (он будет выполняться на прокси аналогично тому, как на агент). Игнорируется, если Hostname задан.
Не поддерживает UserParameters, счётчики производительности или псевдонимы, но поддерживает system.run[].

По умолчанию: system.hostname

HousekeepingFrequency

Как часто Zabbix будет выполнять процедуру очистки устаревших данных (в часах). Очистка устаревших данных — это удаление устаревшей информации из базы данных.
Примечание: Чтобы снизить нагрузку при запуске прокси, очистка устаревших данных откладывается на 30 минут после запуска прокси. Таким образом, если HousekeepingFrequency равно 1, самая первая процедура очистки устаревших данных после запуска прокси будет выполнена через 30 минут, а затем будет повторяться каждый час.
Можно отключить автоматическую очистку устаревших данных, установив HousekeepingFrequency в 0. В этом случае процедуру очистки устаревших данных можно запустить только с помощью параметра управления во время выполнения *housekeeper_execute*.
См. также параметры runtime control и подробности о процедуре очистки устаревших данных.

По умолчанию: 1
 Диапазон: 0-24

Include

Вы можете включать отдельные файлы или все файлы в каталоге в файл конфигурации.
Чтобы включать только подходящие файлы в указанном каталоге, для сопоставления с шаблоном поддерживается символ подстановки звёздочка.
См. специальные примечания об ограничениях.

Пример:

```
Include=/absolute/path/to/config/files/*.conf
```

JavaGateway

IP-адрес (или имя узла) Zabbix Java gateway. Требуется только в том случае, если запущены Java poller'ы.

JavaGatewayPort

Порт, который прослушивает Zabbix Java gateway.

По умолчанию: 10052
 Диапазон: 1024-32767

ListenBacklog

Максимальное количество ожидающих соединений в очереди TCP.
Значение по умолчанию — это жестко заданная константа, которая зависит от системы.
Максимально поддерживаемое значение также зависит от системы; слишком большие значения могут быть молча усечены до «максимума, определяемого реализацией».

По умолчанию: SOMAXCONN
 Диапазон: 0 - INT_MAX

ListenIP

Список IP-адресов, разделённых запятыми, на которых trapper должен прослушивать соединения.
Если этот параметр не указан, trapper будет прослушивать все сетевые интерфейсы.

По умолчанию: 0.0.0.0

ListenPort

Порт прослушивания для trapper.

По умолчанию: 10051
 Диапазон: 1024-32767

LoadModule

Модуль для загрузки при запуске прокси. Модули используются для расширения функциональности прокси. Модуль должен находиться в каталоге, указанном в LoadModulePath, либо перед именем модуля должен быть указан путь. Если указанный перед именем путь является абсолютным (начинается с '/'), то LoadModulePath игнорируется.
Форматы:
LoadModu
указывать несколько параметров LoadModule.

LoadModulePath

Полный путь к расположению модулей прокси. Значение по умолчанию зависит от параметров компиляции.

LogFile

Имя файла журнала.

Обязательно: Да, если для LogType установлено значение *file*; в противном случае — нет

LogFileSize

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 удалась, существующий файл журнала будет усечён и начат заново.

По умолчанию: 1
 Диапазон: 0-1024

LogRemoteCommands

Включить журналирование выполняемых shell-команд как предупреждений.

По умолчанию: 0
 Значения: 0 — отключено, 1 — включено

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

LogSlowQueries

Как долго может выполняться запрос к базе данных, прежде чем он будет записан в журнал (в миллисекундах).
0 — не записывать медленные запросы в журнал.
Этот параметр становится доступным, начиная с DebugLevel=3.

По умолчанию: 0
 Диапазон: 0-3600000

MaxConcurrentChecksPerPoller

Максимальное количество асинхронных проверок, которые могут одновременно выполняться каждым опрашивателем HTTP agent, опрашивателем агента или опрашивателем SNMP. См. StartHTTPAgentPollers, StartAgentPollers и StartSNMP-Pollers.

По умолчанию: 1000
 Диапазон: 1-1000

PidFile

Имя PID-файла.

По умолчанию: `/tmp/zabbix_proxy.pid`

ProxyBufferMode

Указывает механизм хранения данных истории, сетевого обнаружения и авторегистрации: *disk* — данные хранятся в базе данных и выгружаются из базы данных; *memory* — данные хранятся в памяти и выгружаются из памяти. Если буферу не хватает памяти, старые данные будут отброшены. При завершении работы буфер очищается. *hybrid* — буфер прокси обычно работает как в режиме memory, пока не исчерпает память или пока возраст самой старой записи не превысит настроенное значение. Если это происходит, буфер сбрасывается в базу данных и работает как в режиме disk, пока все данные не будут выгружены, 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

Как часто прокси получает данные конфигурации от Zabbix server, в секундах.
Параметр активного прокси. Игнорируется для пассивных прокси (см. параметр ProxyMode).

По умолчанию: 10
 Диапазон: 1-604800

ProxyLocalBuffer

Прокси будет хранить данные локально в течение N часов, даже если эти данные уже были синхронизированы с сервером.
Этот параметр можно использовать, если локальные данные будут использоваться сторонними приложениями.

По умолчанию: 0
 Диапазон: 0-720

ProxyMemoryBufferAge

Максимальный возраст данных в буфере памяти прокси, в секундах. Если параметр включен (не равен нулю) и записи в буфере памяти прокси старше этого значения, то буфер прокси принудительно переключается в режим базы данных до тех пор, пока все записи не будут выгружены на сервер. Этот параметр должен быть меньше или равен параметру ProxyOfflineBuffer.

По умолчанию: 0
 Диапазон: 0;600-864000

ProxyMemoryBufferSize

Размер кэша общей памяти для собранных данных истории, обнаружения и авторегистрации, в байтах. Если включено (не равно нулю), прокси будет хранить данные истории, обнаружения и авторегистрации в памяти, пока кэш не заполнится или пока сохранённые записи не станут старше значения, заданного в ProxyMemoryBufferAge. Этот параметр нельзя использовать вместе с параметром ProxyLocalBuffer.

По умолчанию: 0
 Диапазон: 0;128K-2G

ProxyMode

Режим работы прокси.
0 — прокси в активном режиме
1 — прокси в пассивном режиме
Обратите внимание, что (конфиденциальные) данные конфигурации прокси могут стать доступными сторонам, имеющим доступ к порту trapper сервера Zabbix, при использовании активного прокси. Это возможно, потому что любой может выдать себя за активный прокси и запросить данные конфигурации; аутентификация не выполняется.

По умолчанию: 0
 Диапазон: 0-1

ProxyOfflineBuffer

Прокси будет хранить данные в течение N часов в случае отсутствия соединения с сервером Zabbix.
Более старые данные будут потеряны.

По умолчанию: 1
 Диапазон: 1-720

Сервер

Если ProxyMode установлен в *активный режим*:
IP-адрес или DNS-имя сервера Zabbix (адрес:порт) или кластера (адрес:порт;адрес2:порт), 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 ProxyMode установлен в *пассивный режим*:
Список IP-адресов, разделённых запятыми, при необходимости в нотации CIDR, или DNS-имён сервера Zabbix. 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' считаются эквивалентными.
'::/0' разрешит 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
```

Обязательный: да

SNMPTrapperFile

Временный файл, используемый для передачи данных от демона SNMP trap прокси.
Должен совпадать с указанным в файле конфигурации zabbix_trap_receiver.pl или SNMPTT.

По умолчанию: `/tmp/zabbix_traps.tmp`

SocketDir

Каталог для хранения IPC-сокетов, используемых внутренними службами Zabbix.

По умолчанию: `/tmp`

SourceIP

IP-адрес источника для:

- исходящих подключений к серверу Zabbix
- подключений без использования агента (VMware, SSH, JMX, SNMP, Telnet и простые проверки)
- подключений HTTP agent
- HTTP-запросов JavaScript элемента данных script
- HTTP-запросов JavaScript предобработки
- подключений к Vault

SSHKeyLocation

Расположение открытых и закрытых ключей для SSH-проверок и действий.

SSLCertLocation

Расположение файлов клиентских SSL-сертификатов для аутентификации клиента.
Этот параметр используется только в веб-мониторинге.

SSLKeyLocation

Расположение файлов закрытого SSL-ключа для аутентификации клиента.
Этот параметр используется только в веб-мониторинге.

SSLCALocation

Расположение файлов центра сертификации (CA) для проверки SSL-сертификата сервера.
Обратите внимание, что значение этого параметра будет установлено как параметр libcurl CURLOPT_CAPATH. Для версий libcurl до 7.42.0 это действует только в том случае, если libcurl был скомпилирован для использования OpenSSL. Для получения дополнительной информации смотрите веб-страницу cURL.
Этот параметр используется в веб-мониторинге и при SMTP-аутентификации.

StartAgentPollers

Количество предварительно созданных экземпляров опросчиков агента Zabbix. См. MaxConcurrentChecksPerPoller.

По умолчанию: 1
 Диапазон: 0-1000

StartBrowserPollers

Количество предварительно созданных экземпляров pollers элементов данных браузера.

По умолчанию: 1
 Диапазон: 0-1000

StartDBSyncers

Количество заранее созданных экземпляров синхронизаторов истории.
Примечание: Будьте осторожны при изменении этого значения, его увеличение может принести больше вреда, чем пользы.

По умолчанию: 4
 Диапазон: 1-100

StartDiscoverers

Количество предварительно созданных экземпляров процессов обнаружения.

По умолчанию: 5
 Диапазон: 0-1000

StartHTTPAgentPollers

Количество предварительно созданных экземпляров опросчиков HTTP агент.

См. MaxConcurrentChecksPerPoller.

По умолчанию: 1
 Диапазон: 0-1000

StartHTTPPollers

Количество предварительно созданных экземпляров HTTP pollers.

По умолчанию: 1
 Диапазон: 0-1000

StartIPMIPollers

Количество предварительно созданных экземпляров опросчиков IPMI.

По умолчанию: 0
 Диапазон: 0-1000

StartJavaPollers

Количество предварительно созданных экземпляров Java pollers.

По умолчанию: 0
 Диапазон: 0-1000

StartODBCPollers

Количество предварительно созданных экземпляров опросчиков ODBC.

По умолчанию: 1
 Диапазон: 0-1000

StartPingers

Количество предварительно созданных экземпляров ICMP pingers.

По умолчанию: 1
 Диапазон: 0-1000

StartPollersUnreachable

Количество предварительно созданных экземпляров опросчиков для недоступных узлов сети (включая IPMI и Java). Если запущены обычные, IPMI- или Java-опросчики, должен быть запущен как минимум один опросчик для недоступных узлов сети.

По умолчанию: 1
 Диапазон: 0-1000

StartPollers

Количество предварительно созданных экземпляров pollers.

По умолчанию: 5
 Диапазон: 0-1000

StartPreprocessors

Количество заранее запущенных экземпляров потоков worker предобработки должно быть не меньше количества доступных ядер CPU. Следует указать большее количество worker, если предобработка не ограничена производительностью CPU и включает множество сетевых запросов.

По умолчанию: 16
 Диапазон: 1-1000

StartSNMPPollers

Количество предварительно созданных экземпляров SNMP-опросчиков. См. MaxConcurrentChecksPerPoller.

По умолчанию: 1
 Диапазон: 0-1000

StartSNMPTrapper

Если установлено значение 1, будет запущен процесс SNMP trapper.

По умолчанию: 0
 Диапазон: 0-1

StartTrappers

Количество предварительно созданных экземпляров trappers.
Trappers принимают входящие соединения от Zabbix sender и активных агентов.

По умолчанию: 5
 Диапазон: 0-1000

StartVMwareCollectors

Количество предварительно созданных экземпляров сборщика VMware.

По умолчанию: 0
 Диапазон: 0-250

StatsAllowedIP

Список IP-адресов внешних экземпляров Zabbix, разделённых запятыми, при необходимости в нотации CIDR, или DNS-имён. Запрос статистики будет принят только от адресов, перечисленных здесь. 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
```

Timeout

Указывает, как долго ждать (в секундах) установления соединения и обмена данными с сервером Zabbix, агентом, веб-службой и устаревшими SNMP-проверками (одиночный номер OID или строка).

Этот параметр определяет длительность различных операций обмена данными:

- Выполнение удалённых команд на агенте Zabbix
- Выполнение команд SSH / Telnet
- Перепланирование элементов данных, когда интерфейс IPMI становится недоступным
- Отправка ответа серверу Zabbix при невозможности обмена данными из-за проблем с правами доступа или шифрованием
- Предельное время ожидания для асинхронных IPC-сокетов и параметров управления во время выполнения
- DNS-запросы асинхронного поллера
- Ответ для heartbeat активной проверки
- Получение данных (значений) агента Zabbix от активных агентов
- Получение данных от Zabbix sender
- 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, на уровне прокси или отдельного элемента данных).
Например, элементы данных SNMP `walk[OID]` и `get[OID]` используют тайм-аут, настроенный в веб-интерфейсе; устаревшие SNMP-проверки по-прежнему используют значение тайм-аута сервера.

По умолчанию: 3
 Диапазон: 1-30

TLSAccept

Какие входящие соединения принимать от сервера Zabbix. Используется для пассивного прокси, игнорируется на активном прокси. Можно указать несколько значений, разделенных запятой:
unencrypted — принимать соединения без шифрования (по умолчанию)
psk — принимать соединения с TLS и предварительно общим ключом (PSK)
cert — принимать соединения с TLS и сертификатом

Обязательно: да для пассивного прокси, если определены параметры TLS-сертификата или PSK (даже для соединения *unencrypted*); в противном случае — нет

TLSCAFile

Полный путь к файлу, содержащему сертификаты корневого(ых) центра(ов) сертификации (CA) для проверки сертификата другой стороны, используемый для шифрованного обмена данными между компонентами Zabbix.

TLSCertFile

Полный путь к файлу, содержащему сертификат прокси или цепочку сертификатов, используемый для шифрованного обмена данными между компонентами Zabbix.

TLSCipherAll

Строка приоритетов GnuTLS или строка 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

Строка шифров для OpenSSL 1.1.1 или новее 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

Строка приоритетов GnuTLS или строка 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

Строка 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

Строка приоритетов GnuTLS или строка 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

Строка 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

Как прокси должен подключаться к серверу Zabbix. Используется для активного прокси, игнорируется на пассивном 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 подключения *unencrypted*); в противном случае — нет

TLSCRLFile

Полный путь к файлу, содержащему отозванные сертификаты. Этот параметр используется для шифрованного обмена данными между компонентами Zabbix.

TLSKeyFile

Полный путь к файлу, содержащему закрытый ключ прокси, используемый для шифрованного обмена данными между компонентами Zabbix.

TLSListen

Управляет TLS на сокете trapper.

Поддерживаемые значения:

- required — принимать только TLS-соединения

TLSPSKFile

Полный путь к файлу, содержащему предварительно согласованный ключ прокси, используемый для шифрованного обмена данными с сервером Zabbix.

TLSPSKIdentity

Строка идентификатора предварительно общего ключа, используемая для зашифрованного обмена данными с сервером Zabbix.

TLSServerCertIssuer

Допустимый издатель сертификата сервера.

TLSServerCertSubject

Разрешённый субъект сертификата сервера.

TmpDir

Временный каталог.

По умолчанию: `/tmp`

TrapperTimeout

Указывает тайм-аут в секундах для:
 - получения данных конфигурации с сервера Zabbix;
 - выполнения глобального скрипта или удалённой команды.

По умолчанию: 300
 Диапазон: 1-300

UnavailableDelay

Как часто узел сети проверяется на доступность в течение периода недоступности, в секундах.

По умолчанию: 60
 Диапазон: 1-3600

UnreachableDelay

Как часто узел сети проверяется на доступность в течение периода недоступности, в секундах.

По умолчанию: 15
 Диапазон: 1-3600

UnreachablePeriod

Через сколько секунд недоступности считать узел сети недоступным.

По умолчанию: 45
 Диапазон: 1-3600

Пользователь

Сбросить привилегии до конкретного существующего пользователя в системе.
Имеет эффект только при запуске от имени 'root' и если AllowRoot отключён.

По умолчанию: `zabbix`

Хранилище секретов

Поставщик хранилища секретов:
HashiCorp - HashiCorp KV Secrets Engine версии 2
CyberArk - CyberArk Central Credential Provider
Должен соответствовать поставщику хранилища секретов, заданному в веб-интерфейсе.

По умолчанию: `HashiCorp`

VaultDBPath

Путь или запрос Vault, в зависимости от Vault, откуда будут получены учетные данные для базы данных по ключам.

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 DBUser и DBPassword не указаны.

VaultPrefix

Пользовательский префикс для пути или запроса Vault, в зависимости от Vault. Если не указан, будут использованы наиболее подходящие значения по умолчанию.
Обратите внимание, что для HashiCorp после точки монтирования автоматически добавляется `data`, если VaultPrefix не указан.
Обратите внимание, что для HashiCorp после точки монтирования автоматически добавляется `data`, если VaultPrefix не указан.

Пример префикса для Hashicorp:

```
v1/secret/data/zabbix/
```

Пример префикса для Cyberark:

```
/AIMWebService/api/Accounts?
```

VaultTLSCertFile

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 он рекомендуется для хранилища CyberArkCCP.

VaultTLSKeyFile

Имя файла закрытого SSL-ключа, используемого для аутентификации клиента. Файл закрытого ключа должен быть в формате PEM1.
Каталог, содержащий этот файл, задаётся параметром конфигурации SSLKeyLocation.
Этот параметр можно не указывать, однако он рекомендуется для хранилища CyberArkCCP.

VaultToken

Токен аутентификации HashiCorp vault, который должен быть сгенерирован исключительно для прокси Zabbix с правами только на чтение для пути, указанного в необязательном параметре конфигурации VaultDBPath.
Одновременное определение VaultToken и переменной окружения VAULT_TOKEN является ошибкой.

Обязательно: Да, если для Vault установлено значение *HashiCorp*; в противном случае — нет

VaultURL

URL HTTP[S] сервера хранилища. Если SSLCALocation не указан, будет использоваться общесистемный каталог сертификатов CA.

По умолчанию: `https://127.0.0.1:8200`

VMwareCacheSize

Размер разделяемой памяти для хранения данных VMware.
Для мониторинга использования кэша VMware можно использовать внутреннюю проверку VMware zabbix[vmware,buffer,...] (см. Внутренние проверки).
Обратите внимание, что разделяемая память не выделяется, если не настроен запуск ни одного экземпляра сборщика vmware.

По умолчанию: `8M`
 Диапазон: 256K-2G

VMwareFrequency

Задержка в секундах между сбором данных от отдельного сервиса VMware.
Это значение следует установить равным наименьшему интервалу обновления любого элемента данных мониторинга VMware.

По умолчанию: 60
 Диапазон: 10-86400

VMwarePerfFrequency

Задержка в секундах между получением статистики счётчиков производительности от одной службы VMware.
Это значение следует установить равным наименьшему интервалу обновления любого элемента данных мониторинга VMware item, который использует счётчики производительности VMware.

По умолчанию: 60
 Диапазон: 10-86400

VMwareTimeout

Максимальное количество секунд, в течение которого сборщик vmware будет ожидать ответа от службы VMware (vCenter или гипервизора ESX).

По умолчанию: 10
 Диапазон: 1-300

WebDriverURL

HTTP[S] URL интерфейса WebDriver.

Пример (используется с автономным сервером Selenium WebDriver):

```
WebDriverURL=http://localhost:4444
```

Сноски

[1] DBSocket и DBPort являются взаимоисключающими в конфигурации прокси. Укажите только один параметр или оставьте оба неопределёнными.

3 Zabbix агент (UNIX)

Обзор

В этом разделе перечислены параметры, поддерживаемые файлом конфигурации агента Zabbix (zabbix_agentd.conf).

Параметры перечислены без дополнительной информации. Нажмите на параметр, чтобы увидеть полные сведения.

Parameter	Description
Alias	Задает псевдоним для ключа элемента данных.
AllowKey	Разрешить выполнение тех ключей элементов данных, которые соответствуют шаблону.
AllowRoot	Разрешить запуск агента от имени ’root’.
BufferSend	Не хранить данные в буфере дольше N секунд.
BufferSize	Максимальное количество значений в буфере памяти.
DebugLevel	Уровень отладки.
DenyKey	Запретить выполнение тех ключей элементов данных, которые соответствуют шаблону.
EnableRemoteCommands	Разрешены ли удаленные команды от сервера Zabbix.
HeartbeatFrequency	Частота heartbeat-сообщений в секундах.
HostInterface	Необязательный параметр, определяющий интерфейс узла сети.
HostInterfaceItem	Необязательный параметр, определяющий элемент данных, используемый для получения интерфейса узла сети.
HostMetadata	Необязательный параметр, определяющий метаданные узла сети.
HostMetadataItem	Необязательный параметр, определяющий элемент данных агента Zabbix, используемый для получения метаданных узла сети.
Hostname	Необязательный параметр, определяющий имя узла сети.
HostnameItem	Необязательный параметр, определяющий элемент данных агента Zabbix, используемый для получения имени узла сети.
Include	В файл конфигурации можно включать отдельные файлы или все файлы из каталога.
ListenBacklog	Максимальное количество ожидающих соединений в очереди TCP.
ListenIP	Список IP-адресов, разделенных запятыми, на которых агент должен прослушивать соединения.
ListenPort	Агент будет прослушивать этот порт для соединений от сервера.
LoadModule	Модуль, загружаемый при запуске агента.
LoadModulePath	Полный путь к расположению модулей агента.
LogFile	Имя файла журнала.
LogFileSize	Максимальный размер файла журнала.
LogRemoteCommands	Включить журналирование выполняемых shell-команд как предупреждений.
LogType	Тип вывода журнала.
MaxLinesPerSecond	Максимальное количество новых строк, которое агент будет отправлять в секунду на сервер Zabbix или прокси при обработке активных проверок ’log’ и ’logrt’.
PidFile	Имя PID-файла.
RefreshActiveChecks	Как часто обновляется список активных проверок.
Server	Список IP-адресов серверов Zabbix и прокси Zabbix, разделенных запятыми, при необходимости в нотации CIDR, или DNS-имен.
ServerActive	Адрес сервера Zabbix/прокси или конфигурация кластера для получения активных проверок.
SourceIP	Исходный IP-адрес.

Parameter	Description
StartAgents	Количество предварительно созданных экземпляров zabbix_agentd, обрабатывающих пассивные проверки.
Timeout	Указывает, как долго ждать (в секундах) установления соединения и обмена данными с прокси Zabbix или сервером.
TLSAccept	Какие входящие соединения принимать.
TLSCAFile	Полный путь к файлу, содержащему сертификаты корневых центров сертификации (CA) для проверки сертификата другой стороны, используемый для шифрованного взаимодействия между компонентами Zabbix.
TLSCertFile	Полный путь к файлу, содержащему сертификат агента или цепочку сертификатов, используемый для шифрованного взаимодействия между компонентами Zabbix.
TLSCipherAll	Строка приоритетов GnuTLS или строка шифров OpenSSL (TLS 1.2). Переопределяет критерии выбора набора шифров по умолчанию для шифрования на основе сертификатов и PSK.
TLSCipherAll13	Строка шифров для OpenSSL 1.1.1 или новее в TLS 1.3. Переопределяет критерии выбора набора шифров по умолчанию для шифрования на основе сертификатов и PSK.
TLSCipherCert	Строка приоритетов GnuTLS или строка шифров OpenSSL (TLS 1.2). Переопределяет критерии выбора набора шифров по умолчанию для шифрования на основе сертификатов.
TLSCipherCert13	Строка 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	Строка приоритетов GnuTLS или строка шифров OpenSSL (TLS 1.2). Переопределяет критерии выбора набора шифров по умолчанию для шифрования на основе PSK.
TLSCipherPSK13	Строка шифров для OpenSSL 1.1.1 или новее в TLS 1.3. Переопределяет критерии выбора набора шифров по умолчанию для шифрования на основе PSK.
TLSConnect	Как агент должен подключаться к серверу Zabbix или прокси.
TLSCRLFile	Полный путь к файлу, содержащему отозванные сертификаты. Этот параметр используется для шифрованного взаимодействия между компонентами Zabbix.
TLSKeyFile	Полный путь к файлу, содержащему закрытый ключ агента, используемый для шифрованного взаимодействия между компонентами Zabbix.
TLSPSKFile	Полный путь к файлу, содержащему предварительно общий ключ агента, используемый для шифрованного взаимодействия с сервером Zabbix.
TLSPSKIdentity	Строка идентификатора предварительно общего ключа, используемая для шифрованного взаимодействия с сервером Zabbix.
TLSServerCertIssuer	Разрешенный издатель сертификата сервера (прокси).
TLSServerCertSubject	Разрешенный субъект сертификата сервера (прокси).
UnsafeUserParameters	Разрешить передачу любых символов в аргументах пользовательских параметров.
User	Сбросить привилегии до конкретного существующего пользователя в системе.
UserParameter	Пользовательский параметр для мониторинга.
UserParameterDir	Путь поиска по умолчанию для команд UserParameter.

Все параметры являются необязательными, если явно не указано, что параметр обязателен.

Обратите внимание:

- Значения по умолчанию отражают значения демона по умолчанию, 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

Псевдоним

Задает псевдоним для ключа элемента данных. Его можно использовать для замены длинного и сложного ключа элемента данных более коротким и простым.
 Может присутствовать 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 ID 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 CPU 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 CPU с параметрами по умолчанию, а также **cpu.util[all, idle, avg15]** для получения конкретных данных о загрузке CPU.

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

Разрешить выполнение тех ключей элементов данных, которые соответствуют шаблону. Шаблон ключа представляет собой выражение с подстановочными знаками, поддерживающее символ "*" для сопоставления с любым количеством любых символов.
Можно определить несколько правил сопоставления ключей в сочетании с DenyKey. Параметры обрабатываются по одному в порядке их появления. См. также: Ограничение проверок агента.

AllowRoot

Разрешить агенту работать от имени 'root'. Если отключено и агент запущен пользователем 'root', агент попытается вместо этого переключиться на пользователя 'zabbix'. Не имеет эффекта, если агент запущен от имени обычного пользователя.

По умолчанию: 0
 Значения: 0 - не разрешать; 1 - разрешать

BufferSend

Не хранить данные в буфере дольше N секунд.

По умолчанию: 5
 Диапазон: 1-3600

BufferSize

Максимальное количество значений в буфере памяти. Агент отправит все собранные данные на сервер Zabbix или прокси, если буфер заполнен.

По умолчанию: 100
 Диапазон: 2-65535

DebugLevel

Укажите уровень отладки:
0 — основная 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 ещё 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 знаками, поддерживающее символ "*" для сопоставления с любым количеством любых символов.
Можно определить несколько правил сопоставления ключей в сочетании с AllowKey. Параметры обрабатываются по одному в порядке их появления. См. также: Ограничение проверок агента.

EnableRemoteCommands

Разрешены ли удалённые команды от сервера Zabbix. Этот параметр **устарел**, используйте вместо него AllowKey=system.run[*] или DenyKey=system.run[*].
Это внутренний псевдоним для параметров AllowKey/DenyKey в зависимости от значения:
0 - DenyKey=system.run[*]
1 - AllowKey=system.run[*]

По умолчанию: 0
 Значения: 0 - не разрешать, 1 - разрешать

HeartbeatFrequency

Частота сообщений heartbeat в секундах. Используется для мониторинга доступности активных проверок.
0 — сообщения heartbeat отключены.

По умолчанию: 60
 Диапазон: 0-3600

HostInterface

Необязательный параметр, который определяет интерфейс узла сети (IP-адрес или DNS-имя), используемый в процессе авторегистрации узла сети. Это значение будет использовано для заполнения интерфейса во вновь созданном узле сети

и позволяет явно настроить либо IP-адрес, либо DNS-адрес. Подробнее см. в разделе Использование DNS в качестве интерфейса по умолчанию.

Если не определено, значение будет получено из HostInterfaceItem.

Агент выдаст ошибку и не запустится, если значение превышает ограничение в 255 символов.

Диапазон: 0-255 символов

HostInterfaceItem

Необязательный параметр, который определяет элемент данных, используемый для определения интерфейса узла сети (IP-адреса или DNS-имени) в процессе авторегистрации узла сети. Это значение используется только в том случае, если HostInterface не определён. Подробнее см. в разделе Использование DNS в качестве интерфейса по умолчанию.

Во время запроса авторегистрации агент запишет предупреждающее сообщение, если значение, возвращаемое указанным элементом данных, превышает ограничение в 255 символов.

Элемент данных system.run[] поддерживается независимо от настроек AllowKey/DenyKey.

HostMetadata

Необязательный параметр, который определяет метаданные, используемые для идентификации или различения узла сети в процессе авторегистрации узла сети (активный агент). HostMetadata позволяет различать узлы сети не только по имени узла сети.

Если не определено, значение будет получено из HostMetadataItem.

Агент выдаст ошибку и не запустится, если указанное значение превышает ограничение в 2034 байта или является строкой не в UTF-8. Если параметр ожидает IP-адрес или DNS-имя, значения, которые являются допустимыми UTF-8, но не являются допустимыми IP-адресами или DNS-именами, также будут отклонены и помечены как недопустимые.

Диапазон: 0-2034 байт

HostMetadataItem

Необязательный параметр, который определяет элемент данных агента Zabbix, используемый для получения метаданных узла сети. Этот параметр используется только в том случае, если HostMetadata не задан.

Значение HostMetadataItem извлекается при каждой попытке авторегистрации и используется только в процессе авторегистрации узла сети (активный агент). HostMetadataItem позволяет различать узлы сети не только по имени узла сети.

Пользовательские параметры и псевдонимы поддерживаются. Элемент данных system.run[] поддерживается независимо от настроек AllowKey/DenyKey.

Во время запроса авторегистрации агент записывает предупреждающее сообщение в журнал, если значение, возвращаемое указанным элементом данных, превышает ограничение в 65535 кодовых точек UTF-8. Значение, возвращаемое элементом данных, должно быть строкой UTF-8, в противном случае оно будет проигнорировано. Если параметр ожидает IP-адрес или DNS-имя, значения, которые являются допустимыми UTF-8, но не являются допустимыми IP-адресами или DNS-именами, также будут отклонены и отмечены как недопустимые.

Hostname

Список уникальных, чувствительных к регистру имен узлов сети, разделенных запятыми. Обязательно для активных проверок и должно совпадать с именами узлов сети, настроенными на сервере. Если значение не задано, оно берется из HostnameItem.
Допустимые символы: буквы и цифры, '.', ' ', '_' и '-'. Максимальная длина: 128 символов для каждого имени узла сети, 2048 символов для всей строки.

По умолчанию: задается параметром HostnameItem

HostnameItem

Необязательный параметр, который определяет элемент данных агента Zabbix, используемый для получения имени узла сети. Этот параметр используется только в том случае, если Hostname не определён. Пользовательские параметры или псевдонимы не поддерживаются, но элемент данных system.run[] поддерживается независимо от значений AllowKey/DenyKey.

По умолчанию: `system.hostname`

Include

Вы можете включить отдельные файлы или все файлы в каталоге в файл конфигурации. Чтобы включать только соответствующие файлы в указанном каталоге, для сопоставления с шаблоном поддерживается символ подстановки «звёздочка».
См. специальные примечания об ограничениях.

Пример:

```
Include=/absolute/path/to/config/files/*.conf
```

ListenBacklog

Максимальное количество ожидающих соединений в очереди TCP.
Значение по умолчанию — это жестко заданная константа, которая зависит от системы.
Максимально поддерживаемое значение также зависит от системы; слишком большие значения могут быть молча усечены до «максимума, определяемого реализацией».

По умолчанию: SOMAXCONN
 Диапазон: 0 - INT_MAX

ListenIP

Список IP-адресов, разделённых запятыми, на которых агент должен прослушивать соединения.

По умолчанию: 0.0.0.0

ListenPort

Агент будет прослушивать этот порт для соединений от сервера.

По умолчанию: 10050
 Диапазон: 1024-32767

LoadModule

Модуль, загружаемый при запуске агента. Модули используются для расширения функциональности агента. Модуль должен находиться в каталоге, указанном в LoadModulePath, либо перед именем модуля должен быть указан путь. Если указанный перед именем путь является абсолютным (начинается с '/'), то LoadModulePath игнорируется.
Форматы:
LoadModu
указывать несколько параметров LoadModule.

LoadModulePath

Полный путь к расположению модулей агента. Значение по умолчанию зависит от параметров компиляции.

LogFile

Имя файла журнала.

Обязательно: Да, если для LogType установлено значение *file*; в противном случае — нет

LogFileSize

Максимальный размер файла журнала в МБ.
0 — отключить автоматическую ротацию журнала.
Примечание: Если достигнут предельный размер файла журнала и ротация файла по какой-либо причине не удалась, существующий файл журнала усекается и начинается заново.

По умолчанию: 1
 Диапазон: 0-1024

LogRemoteCommands

Включить журналирование выполняемых shell-команд как предупреждений. Команды будут заноситься в журнал, только если они выполняются удалённо. Записи журнала не будут создаваться, если system.run[] запускается локально параметрами HostMetadataItem, HostInterfaceItem или HostnameItem.

По умолчанию: 0
 Значения: 0 — отключено, 1 — включено

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

MaxLinesPerSecond

Максимальное количество новых строк, которое агент будет отправлять в секунду на сервер Zabbix или прокси при обработке активных проверок 'log' и '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 указано в *MaxLinesPerSecond*, чтобы найти требуемую строку в элементах данных журнала.

По умолчанию: 20
 Диапазон: 1-1000

PidFile

Имя PID-файла.

По умолчанию: /tmp/zabbix_agentd.pid

RefreshActiveChecks

Как часто обновляется список 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

Список IP-адресов, разделённых запятыми, при необходимости в нотации CIDR, или DNS-имён серверов Zabbix и прокси Zabbix. 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' 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 объявлены устаревшими в RFC4291. Пробелы допускаются.

Пример:

```
Server=127.0.0.1,192.168.1.0/24,::1,2001:db8::/32,zabbix.example.com
```

Обязательно: да, если StartAgents явно не установлен в 0

ServerActive

Адрес сервера/прокси Zabbix или конфигурация кластера для получения активных проверок. Адрес сервера/прокси — это IP-адрес или DNS-имя и необязательный порт, разделённые двоеточием.
Конфигурация кластера — это один или несколько адресов участников группы сервера или прокси, разделённых точкой с запятой. Можно указать несколько серверов/кластеров Zabbix и прокси Zabbix, разделённых запятыми. Если не используются группы прокси, не следует указывать более одного прокси Zabbix для каждого сервера/кластера Zabbix. Если указан прокси Zabbix, то сервер/кластер Zabbix для этого прокси указывать не следует.
Можно указать несколько адресов, разделённых 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

IP-адрес источника для:

- исходящих подключений к серверу Zabbix или прокси Zabbix;
- выполнения подключений при обработке некоторых элементов данных (web.page.get, net.tcp.port и т. д.).

StartAgents

Количество предварительно созданных экземпляров zabbix_agentd, которые обрабатывают пассивные проверки. Если установлено значение 0, пассивные проверки отключаются, и агент не будет прослушивать ни один TCP-порт.

По умолчанию: 10
 Диапазон: 0-100

Timeout

Указывает, как долго ждать (в секундах) установления соединения и обмена данными с прокси или сервером Zabbix.

Этот параметр определяет длительность различных операций обмена данными, 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 heartbeat-сообщений;
- максимальную длительность проверок `vfs.*`;
- использование модулями агента Zabbix;
- использование в качестве резервного значения в сценариях, когда сервер или прокси версии ниже 7.0 отправляет проверки без тайм-аутов.

Этот тайм-аут **не** будет использоваться для тех проверок агента, для которых настраиваются параметры тайм-аута в веб-интерфейсе (на глобальном уровне, на уровне прокси или на уровне отдельного элемента данных).

По умолчанию: 3
 Диапазон: 1-30

TLSAccept

Какие входящие соединения принимать. Используется для пассивных проверок. Можно указать несколько значений, разделенных запятой:
unencrypted — принимать соединения без шифрования (по умолчанию)
psk — принимать соединения с TLS и предварительно разделенным ключом (PSK)
cert — принимать соединения с TLS и сертификатом

Обязательно: да, если определены параметры TLS-сертификата или PSK (даже для *unencrypted* соединения); в противном случае нет

TLSCAFile

Полный путь к файлу, содержащему сертификаты корневых центров сертификации (CA) для проверки сертификата другой стороны, используемый для шифрованного обмена данными между компонентами Zabbix.

TLSCertFile

Полный путь к файлу, содержащему сертификат агента или цепочку сертификатов, используемых для шифрованного обмена данными с компонентами Zabbix.

TLSCipherAll

Строка приоритетов GnuTLS или строка 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

Строка шифров для OpenSSL 1.1.1 или новее 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

Строка приоритета GnuTLS или строка 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

Строка 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

Строка приоритетов GnuTLS или строка 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

Строка 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

Как агент должен подключаться 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

Полный путь к файлу, содержащему отозванные сертификаты. Этот параметр используется для шифрованного обмена данными между компонентами Zabbix.

TLSKeyFile

Полный путь к файлу, содержащему закрытый ключ агента, используемый для шифрованного обмена данными между компонентами Zabbix.

TLSPSKFile

Полный путь к файлу, содержащему предварительно согласованный ключ агента, используемый для шифрованного обмена данными с сервером Zabbix.

TLSPSKIdentity

Строка идентификатора предварительно разделяемого ключа, используемая для шифрованного обмена данными с сервером Zabbix.

TLSServerCertIssuer

Допустимый издатель сертификата сервера (прокси).

TLSServerCertSubject

Разрешённый subject сертификата сервера (прокси).

UnsafeUserParameters

Разрешить передачу всех символов в аргументах пользовательских параметров. Следующие символы не разрешены: \ ' " ` * ? [ ] { } ~ $! & ; ( ) < > | # @ %
Кроме того, символы новой строки не разрешены.

По умолчанию: 0
 Значения: 0 — не разрешать, 1 — разрешать

Пользователь

Сбрасывает привилегии до конкретного существующего пользователя в системе.
Имеет эффект только при запуске от имени 'root' и если AllowRoot отключён.

По умолчанию: `zabbix`

UserParameter

Пользовательский параметр для мониторинга. Может быть несколько пользовательских параметров.
Формат: UserParameter=<key>,<shell command>
Обратите внимание, что shell-команда не должна возвращать пустую строку или только символ конца строки. Shell-команды могут иметь относительные пути, если указан параметр UserParameterDir.

Пример:

```
UserParameter=system.test,who|wc -l
UserParameter=check_cpu,./custom_script.sh
```

UserParameterDir

Путь поиска по умолчанию для команд UserParameter. Если используется, агент перед выполнением команды изменит свой рабочий каталог на указанный здесь. 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

4 Zabbix агент 2 (UNIX)

Обзор

Zabbix агент 2 — это агент Zabbix нового поколения, который может использоваться вместо агента Zabbix.

Параметры, поддерживаемые файлом конфигурации Zabbix агента 2 (zabbix_agent2.conf), перечислены в этом разделе.

Параметры приведены без дополнительной информации. Нажмите на параметр, чтобы увидеть полные сведения.

Параметр	Описание
Alias	Задает псевдоним для ключа элемента данных.
AllowKey	Разрешает выполнение тех ключей элементов данных, которые соответствуют шаблону.
BufferSend	Не хранить данные в буфере дольше N секунд.
BufferSize	Максимальное количество значений в буфере памяти.
ControlSocket	Управляющий сокет, используемый для отправки команд времени выполнения с опцией '-R'.
DebugLevel	Уровень отладки.
DenyKey	Запрещает выполнение тех ключей элементов данных, которые соответствуют шаблону.
EnablePersistentBuffer	Включает использование локального постоянного хранилища для активных элементов данных.
ForceActiveChecksOnStart	Выполняет активные проверки сразу после перезапуска для первой полученной конфигурации.
HeartbeatFrequency	Частота heartbeat-сообщений в секундах.
HostInterface	Необязательный параметр, определяющий интерфейс узла сети.
HostInterfaceItem	Необязательный параметр, определяющий элемент данных, используемый для получения интерфейса узла сети.
HostMetadata	Необязательный параметр, определяющий метаданные узла сети.
HostMetadataItem	Необязательный параметр, определяющий элемент данных агента Zabbix, используемый для получения метаданных узла сети.
Hostname	Необязательный параметр, определяющий имя узла сети.
HostnameItem	Необязательный параметр, определяющий элемент данных агента Zabbix, используемый для получения имени узла сети.
Include	В файл конфигурации можно включать отдельные файлы или все файлы из каталога.
ListenIP	Список IP-адресов, разделенных запятыми, на которых агент должен прослушивать соединения.
ListenPort	Агент будет прослушивать этот порт для соединений от сервера.
LogFile	Имя файла журнала.
LogFileSize	Максимальный размер файла журнала.
LogType	Тип вывода журнала.
PersistentBufferFile	Файл, в котором Zabbix агент 2 должен хранить базу данных SQLite.
PersistentBufferPeriod	Период времени, в течение которого данные должны храниться при отсутствии соединения с сервером или прокси.
PidFile	Имя PID-файла.
Plugins.<PluginName>.System.Capacity	Емкость проверок на плагин, которые могут выполняться одновременно.
Plugins.Log.MaxLinesPerSecond	Максимальное количество новых строк в секунду, которое агент будет отправлять на Zabbix сервер или прокси при обработке активных проверок 'log' и 'logrt'.
Plugins.SystemRun.LogRemoteCommands	Включает журналирование выполняемых shell-команд как предупреждений.
PluginSocket	Путь к UNIX-сокету для взаимодействия с загружаемыми плагинами.
PluginTimeout	Тайм-аут соединений с загружаемыми плагинами в секундах.
RefreshActiveChecks	Как часто обновляется список активных проверок.

Параметр	Описание
Server	Список IP-адресов, разделенных запятыми, при необходимости в нотации CIDR, или DNS-имен серверов Zabbix и прокси Zabbix.
ServerActive	Адрес сервера/прокси Zabbix или конфигурация кластера для получения активных проверок.
SourceIP	Исходный IP-адрес.
StatusPort	Если задано, агент будет прослушивать этот порт для HTTP-запросов состояния (http://localhost:<port>/status).
Timeout	Указывает, как долго ждать (в секундах) установления соединения и обмена данными с прокси Zabbix или сервером.
TLSAccept	Какие входящие соединения принимать.
TLSCAFile	Полный путь к файлу, содержащему сертификаты корневых CA для проверки сертификата удаленной стороны, используемый для шифрованного взаимодействия между компонентами Zabbix.
TLSCertFile	Полный путь к файлу, содержащему сертификат агента или цепочку сертификатов, используемый для шифрованного взаимодействия между компонентами Zabbix.
TLSCipherAll	Строка шифров OpenSSL (TLS 1.2). Переопределяет критерии выбора набора шифров по умолчанию для шифрования на основе сертификатов и PSK.
TLSCipherAll13	Строка 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.
TLSCipherCert	Строка шифров OpenSSL (TLS 1.2). Переопределяет критерии выбора набора шифров по умолчанию для шифрования на основе сертификатов.
TLSCipherCert13	Строка шифров OpenSSL (TLS 1.3) для OpenSSL 1.1.1 или новее. Переопределяет критерии выбора набора шифров по умолчанию для шифрования на основе сертификатов.
TLSCipherPSK	Строка шифров OpenSSL (TLS 1.2). Переопределяет критерии выбора набора шифров по умолчанию для шифрования на основе PSK.
TLSCipherPSK13	Строка шифров OpenSSL (TLS 1.3) для OpenSSL 1.1.1 или новее. Переопределяет критерии выбора набора шифров по умолчанию для шифрования на основе PSK.
TLSConnect	Как агент должен подключаться к Zabbix серверу или прокси.
TLSCRLFile	Полный путь к файлу, содержащему отозванные сертификаты. Этот параметр используется для шифрованного взаимодействия между компонентами Zabbix.
TLSKeyFile	Полный путь к файлу, содержащему закрытый ключ агента, используемый для шифрованного взаимодействия между компонентами Zabbix.
TLSPSKFile	Полный путь к файлу, содержащему предварительно общий ключ агента, используемый для шифрованного взаимодействия с Zabbix сервером.
TLSPSKIdentity	Строка идентификатора предварительно общего ключа, используемая для шифрованного взаимодействия с Zabbix сервером.
TLSServerCertIssuer	Разрешенный издатель сертификата сервера (прокси).
TLSServerCertSubject	Разрешенный субъект сертификата сервера (прокси).
UnsafeUserParameters	Разрешает передавать любые символы в аргументах пользовательских параметров.
UserParameter	Пользовательский параметр для мониторинга.
UserParameterDir	Путь поиска по умолчанию для команд UserParameter.

Все параметры не являются обязательными, 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 окружения.
- Zabbix поддерживает файлы конфигурации только в кодировке UTF-8 без BOM.
- Комментарии, начинающиеся с ”#”, поддерживаются только в начале строки.

Подробности параметра

Alias

Задает 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 ID 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 утилизации CPU 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 утилизации CPU с параметрами по умолчанию, а также **cpu.util[all, idle, avg15]** для получения конкретных данных об утилизации CPU.

Пример 3: Запуск нескольких правил обнаружения низкого уровня, обрабатывающих одни и те же элементы обнаружения.

```
Alias=vfs.fs.discovery[*]:vfs.fs.discovery
```

Теперь можно настроить несколько правил обнаружения, использующих **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 представляет собой выражение с подстановочными знаками, поддерживающее символ "*" для соответствия любому количеству любых символов.
Можно определить несколько правил сопоставления ключей в сочетании с DenyKey. Параметры обрабатываются по одному в порядке их появления. См. также: Ограничение проверок агента.

BufferSend

Интервал времени в секундах, который определяет, как часто значения отправляются из буфера на сервер Zabbix. Обратите внимание, что если буфер заполнен, данные будут отправлены раньше.

По умолчанию: 5
 Диапазон: 1-3600

BufferSize

Максимальное количество значений в буфере памяти. Агент отправит все собранные данные на сервер Zabbix или прокси, если буфер заполнен. Этот параметр следует использовать только в том случае, если постоянный буфер отключен (*EnablePersistentBuffer=0*).

По умолчанию: 1000
 Диапазон: 2-65535

ControlSocket

Управляющий сокет, используемый для отправки команд времени выполнения с опцией '-R'.

По умолчанию: `/tmp/agent.sock`

DebugLevel

Укажите уровень отладки:
0 - основная информация о запуске и остановке процессов Zabbix
1 - критическая информация;
2 - информация об ошибках;
3 - предупреждения;
4 — для отладки (выдает много информации);
5 — расширенная отладка (выдает 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 знаками, поддерживающее символ "*" для сопоставления с любым количеством любых символов.
Можно определить несколько правил сопоставления ключей в сочетании с AllowKey. Параметры обрабатываются по одному в порядке их появления. См. также: Ограничение проверок агента.

EnablePersistentBuffer

Включает использование локального постоянного хранилища для активных элементов данных. Если постоянное хранилище отключено, будет использоваться буфер памяти.

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

Частота сообщений heartbeat в секундах. Используется для мониторинга доступности активных проверок.
0 — сообщения heartbeat отключены.

По умолчанию: 60
 Диапазон: 0-3600

HostInterface

Необязательный параметр, который определяет интерфейс узла сети (IP-адрес или DNS-имя), используемый в процессе авторегистрации. Это значение будет использовано для заполнения интерфейса во вновь созданном узле сети и позволяет явно настроить либо IP-, либо DNS-адрес. Подробнее см. в разделе Использование DNS в качестве интерфейса по умолчанию.

Если не задан, значение будет получено из HostInterfaceItem.

Агент выдаст ошибку и не запустится, если значение превышает ограничение в 255 символов.

Диапазон: 0-255 символов

HostInterfaceItem

Необязательный параметр, который определяет элемент данных, используемый для определения интерфейса узла сети (IP-адреса или DNS-имени) в процессе авторегистрации узла сети. Это значение используется только в том случае, если HostInterface не определён. Подробнее см. в разделе Использование DNS в качестве интерфейса по умолчанию.

Во время запроса авторегистрации агент запишет предупреждающее сообщение, если значение, возвращаемое указанным элементом данных, превышает ограничение в 255 символов.

Элемент данных system.run[] поддерживается независимо от настроек AllowKey/DenyKey.

HostMetadata

Необязательный параметр, который определяет метаданные, используемые для идентификации или различения узла сети в процессе авторегистрации узла сети. HostMetadata позволяет различать узлы сети не только по имени узла сети.

Если не задан, значение будет получено из HostMetadataItem.

Агент выдаст ошибку и не запустится, если указанное значение превышает ограничение в 2034 байта или является строкой не в формате UTF-8. Если параметр ожидает IP-адрес или DNS-имя, значения, которые являются допустимыми UTF-8, но не являются допустимыми IP-адресами или DNS-именами, также будут отклонены и помечены как недопустимые.

Диапазон: 0-2034 байт

HostMetadataItem

Необязательный параметр, который определяет элемент данных агента Zabbix, используемый для получения метаданных узла сети. Этот параметр используется только в том случае, если HostMetadata не задан.

Значение HostMetadataItem извлекается при каждой попытке авторегистрации и используется только в процессе авторегистрации узла сети. HostMetadataItem позволяет различать узлы сети не только по имени узла сети.

Пользовательские параметры и псевдонимы поддерживаются. Элемент данных system.run[] поддерживается независимо от настроек AllowKey/DenyKey.

Во время запроса авторегистрации агент записывает предупреждающее сообщение в журнал, если значение, возвращаемое указанным элементом данных, превышает ограничение в 65535 кодовых точек UTF-8. Значение, возвращаемое элементом данных, должно быть строкой UTF-8, в противном случае оно будет проигнорировано. Если параметр ожидает IP-адрес или DNS-имя, значения, которые являются допустимыми UTF-8, но не являются допустимыми IP-адресами или DNS-именами, также будут отклонены и отмечены как недопустимые.

Hostname

Список уникальных, чувствительных к регистру имен узлов сети, разделенных запятыми. Обязательно для активных проверок и должно совпадать с именами узлов сети, настроенными на сервере. Если значение не определено, оно получается из HostnameItem.
Допустимые символы: буквы и цифры, '.', ' ', '_' и '-'. Максимальная длина: 128 символов на одно имя узла сети, 2048 символов для всей строки.

По умолчанию: задается HostnameItem

HostnameItem

Необязательный параметр, который определяет элемент данных, используемый для получения имени узла сети. Этот параметр используется только в том случае, если Hostname не задан. Пользовательские параметры или псевдонимы не поддерживаются, но элемент данных system.run[] поддерживается независимо от значений AllowKey/DenyKey.

По умолчанию: `system.hostname`

Include

Вы можете включать отдельные файлы или все файлы в каталоге в файл конфигурации. Во время установки Zabbix создаст каталог include в /usr/local/etc, если этот путь не был изменён во время компиляции. Путь может быть относительным к расположению файла *zabbix_agent2.conf*.
Чтобы включать только соответствующие файлы в указанном каталоге, поддерживается символ подстановки звёздочка для сопоставления с шаблоном.
См. специальные примечания об ограничениях.

Пример:

```
Include=/absolute/path/to/config/files/*.conf
```

ListenIP

Список IP-адресов, разделённых запятыми, которые агент должен прослушивать. Первый IP-адрес отправляется на сервер Zabbix при подключении к нему для получения списка активных проверок.

По умолчанию: `0.0.0.0`

ListenPort

Агент будет прослушивать этот порт для подключений от сервера.

По умолчанию: 10050
 Диапазон: 1024-32767

LogFile

Имя файла журнала.

По умолчанию: `/tmp/zabbix_agent2.log`
 Обязательно: Да, если LogType установлен в *file*; в противном случае — нет

LogFileSize

Максимальный размер файла журнала в МБ.
0 — отключить автоматическую ротацию журнала.
Примечание: Если достигнут предельный размер файла журнала и ротация файла по какой-либо причине завершается неудачей, существующий файл журнала усекается и начинается 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`

PersistentBufferFile

Файл, в котором Zabbix агент 2 должен хранить базу данных SQLite. Должен содержать полное имя файла. Этот параметр используется только в том случае, если включен постоянный буфер (*EnablePersistentBuffer=1*).

PersistentBufferPeriod

Период времени, в течение которого данные должны храниться при отсутствии соединения с сервером или прокси. Более старые данные будут потеряны. Данные журнала будут сохранены. Этот параметр используется только в том случае, если включен постоянный буфер (*EnablePersistentBuffer=1*).

По умолчанию: `1h`
 Диапазон: 1m-365d

PidFile

Имя PID-файла.

По умолчанию: `/tmp/zabbix_agent2.pid`

Plugins.<PluginName>.System.Capacity

Предел количества проверок для плагина <PluginName>, которые могут выполняться одновременно.

По умолчанию: 1000 Диапазон: 1-1000

Plugins.Log.MaxLinesPerSecond

Максимальное количество новых строк, которое агент будет отправлять в секунду на сервер Zabbix или прокси при обработке активных проверок 'log' и '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 журналирование выполняемых shell-команд как предупреждений. Команды будут заноситься в журнал только в том случае, если они выполняются удалённо. Записи журнала не будут создаваться, если system.run[] запускается локально параметрами HostMetadataItem, HostInterfaceItem или HostnameItem.

По умолчанию: 0
 Значения: 0 — отключено, 1 — включено

PluginSocket

Путь к UNIX-сокету для обмена данными с загружаемыми плагинами.

По умолчанию: `/tmp/agent.plugin.sock`

PluginTimeout

Тайм-аут для соединений с загружаемыми плагинами, в секундах.

По умолчанию: `Timeout`
 Диапазон: 1-30

RefreshActiveChecks

Как часто обновляется список 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

Список IP-адресов, разделённых запятыми, при необходимости в нотации CIDR, или DNS-имён серверов Zabbix или прокси Zabbix. 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' разрешит 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 отключаются, 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

Адрес сервера/прокси Zabbix или конфигурация кластера для получения активных проверок. Адрес сервера/прокси — это IP-адрес или DNS-имя и необязательный порт, разделённые двоеточием.
Конфигурация кластера представляет собой один или несколько адресов участников группы сервера или прокси, разделённых точкой с запятой. Можно указать несколько серверов/кластеров Zabbix и прокси Zabbix, разделённых запятой. Если не используются группы прокси, не следует указывать более одного прокси Zabbix для каждого сервера/кластера Zabbix. Если указан прокси Zabbix, то сервер/кластер Zabbix для этого прокси указывать не следует.
Можно указать несколько адресов, разделённых запятыми, чтобы использовать несколько независимых серверов Zabbix параллельно. Пробелы допускаются.
Если порт не указан, используется порт по умолчанию.
Адреса IPv6 должны быть заключены в квадратные скобки, если для этого узла сети указан порт. Если порт не указан, квадратные скобки для адресов IPv6 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

IP-адрес источника для:

- исходящих соединений с сервером Zabbix или прокси Zabbix.
- установки соединений при выполнении некоторых элементов данных (web.page.get, net.tcp.port и т. д.).

StatusPort

Если задано, агент будет прослушивать этот порт для HTTP-запросов состояния (http://localhost:<port>/status).

Диапазон: 1024-32767

Timeout

Указывает, как долго ждать (в секундах) установления соединения и обмена данными с прокси или сервером Zabbix.

Этот параметр определяет длительность различных операций обмена данными, 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 heartbeat-сообщений;
- максимальную длительность проверок `vfs.*`;
- использование в качестве резервного значения в сценариях, когда сервер или прокси версии ниже 7.0 отправляет проверки без тайм-аутов.

Этот тайм-аут **не** будет использоваться для тех проверок агента, для которых в веб-интерфейсе можно настроить параметры тайм-аута (на глобальном уровне, уровне прокси или отдельного элемента данных).

Значение по умолчанию: 3
 Диапазон: 1-30

TLSAccept

Входящие соединения, которые следует принимать. Используется для пассивных проверок. Можно указать несколько значений, разделенных запятой:
unencrypted — принимать соединения без шифрования (по умолчанию)
psk — принимать соединения с TLS и предварительно общим ключом (PSK)
cert — принимать соединения с TLS и сертификатом

Обязательно: да, если определены параметры TLS-сертификата или PSK (даже для соединения *unencrypted*); в противном случае нет

TLSCAFile

Полный путь к файлу, содержащему сертификаты корневых центров сертификации (CA) для проверки сертификата удалённой стороны, используемый для шифрованного обмена данными между компонентами Zabbix.

TLSCertFile

Полный путь к файлу, содержащему сертификат агента или цепочку сертификатов, используемых для шифрованного обмена данными с компонентами Zabbix.

TLSCipherAll

Строка 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

Строка 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

Строка 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 соединения по сертификату (TLSConnect=cert) используйте вместо этого TLSCipherAll.

TLSCipherCert13

Строка шифров OpenSSL (TLS 1.3) для OpenSSL 1.1.1 или новее. 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 соединения по сертификату (TLSConnect=cert) используйте вместо этого TLSCipherAll13.

TLSCipherPSK

Строка шифров OpenSSL (TLS 1.2). Переопределяет критерии выбора набора шифров по умолчанию для шифрования на основе PSK.

Пример:

```
kECDHEPSK+AES128:kPSK+AES128
```

TLSCipherPSK13

Строка шифров OpenSSL (TLS 1.3) для OpenSSL 1.1.1 или новее. Переопределяет критерии выбора набора шифров по умолчанию для шифрования на основе PSK.

Пример:

```
TLS_CHACHA20_POLY1305_SHA256:TLS_AES_128_GCM_SHA256
```

TLSConnect

Как агент должен подключаться 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

Полный путь к файлу, содержащему отозванные сертификаты. Этот параметр используется для шифрованного обмена данными между компонентами Zabbix.

TLSKeyFile

Полный путь к файлу, содержащему закрытый ключ агента, используемый для шифрованного обмена данными между компонентами Zabbix.

TLSPSKFile

Полный путь к файлу, содержащему предварительно разделяемый ключ агента, используемый для шифрованного обмена данными с сервером Zabbix.

TLSPSKIdentity

Строка идентификатора предварительно общего ключа, используемая для шифрованного обмена данными с сервером Zabbix.

TLSServerCertIssuer

Допустимый издатель сертификата сервера (прокси).

TLSServerCertSubject

Разрешённый субъект сертификата сервера (прокси).

UnsafeUserParameters

Разрешить передачу любых символов в аргументах пользовательских параметров. Следующие символы не допускаются: \ ' " ` * ? [ ] { } ~ $! & ; ( ) < > | # @ %
Кроме того, не допускаются символы новой строки.

По умолчанию: 0
 Значения: 0 — не разрешать, 1 — разрешать

UserParameter

Пользовательский параметр для мониторинга. Может быть несколько пользовательских параметров.
Формат: UserParameter=<key>,<shell command>
Обратите внимание, что shell-команда не должна возвращать пустую строку или только символ конца строки. Shell-команды могут иметь относительные пути, если указан параметр UserParameterDir.

Пример:

```
UserParameter=system.test,who|wc -l
UserParameter=check_cpu,./custom_script.sh
```

UserParameterDir

Путь поиска по умолчанию для команд UserParameter. Если используется, агент перед выполнением команды изменит свой рабочий каталог на указанный здесь. Таким образом, команды UserParameter могут использовать относительный префикс `./` вместо полного пути.
Допускается только одна запись.

Пример:

```
UserParameterDir=/opt/myscripts
```

5 Zabbix агент (Windows)

Обзор

В этом разделе перечислены параметры, поддерживаемые файлом конфигурации Windows агента Zabbix (zabbix_agentd.conf).

Параметры перечислены без дополнительной информации. Нажмите на параметр, чтобы увидеть полные сведения.

Parameter	Description
Alias	Задает псевдоним для ключа элемента данных.
AllowKey	Разрешает выполнение тех ключей элементов данных, которые соответствуют шаблону.
BufferSend	Не хранить данные в буфере дольше N секунд.
BufferSize	Максимальное количество значений в буфере памяти.
DebugLevel	Уровень отладки.
DenyKey	Запрещает выполнение тех ключей элементов данных, которые соответствуют шаблону.
EnableRemoteCommands	Разрешены ли удаленные команды от сервера Zabbix.
HeartbeatFrequency	Частота heartbeat-сообщений в секундах.
HostInterface	Необязательный параметр, определяющий интерфейс узла сети.
HostInterfaceItem	Необязательный параметр, определяющий элемент данных, используемый для получения интерфейса узла сети.
HostMetadata	Необязательный параметр, определяющий метаданные узла сети.
HostMetadataItem	Необязательный параметр, определяющий элемент данных агента Zabbix, используемый для получения метаданных узла сети.
Hostname	Необязательный параметр, определяющий имя узла сети.
HostnameItem	Необязательный параметр, определяющий элемент данных агента Zabbix, используемый для получения имени узла сети.
Include	В файл конфигурации можно включать отдельные файлы или все файлы из каталога.
ListenBacklog	Максимальное количество ожидающих соединений в очереди TCP.
ListenIP	Список IP-адресов, разделенных запятыми, на которых агент должен прослушивать соединения.
ListenPort	Агент будет прослушивать этот порт для соединений от сервера.
LogFile	Имя файла журнала.
LogFileSize	Максимальный размер файла журнала.
LogRemoteCommands	Включить журналирование выполняемых shell-команд как предупреждений.
LogType	Тип вывода журнала.
MaxLinesPerSecond	Максимальное количество новых строк, которое агент будет отправлять в секунду на сервер Zabbix или прокси при обработке активных проверок 'log' и 'logrt'.
PerfCounter	Определяет новый параметр <parameter_name>, который является средним значением системного счетчика производительности <perf_counter_path> за указанный период времени <period> (в секундах).
PerfCounterEn	Определяет новый параметр <parameter_name>, который является средним значением системного счетчика производительности <perf_counter_path> за указанный период времени <period> (в секундах). По сравнению с PerfCounter, пути perfcounter должны быть на английском языке.
RefreshActiveChecks	Как часто обновляется список активных проверок.
Server	Список IP-адресов, разделенных запятыми, при необходимости в нотации CIDR, или DNS-имен серверов Zabbix и прокси Zabbix.
ServerActive	Адрес сервера/прокси Zabbix или конфигурация кластера для получения активных проверок.

Parameter	Description
SourceIP	Исходный IP-адрес.
StartAgents	Количество предварительно созданных экземпляров zabbix_agentd, обрабатывающих пассивные проверки.
Timeout	Указывает, как долго ждать (в секундах) установления соединения и обмена данными с прокси Zabbix или сервером.
TLSAccept	Какие входящие соединения принимать.
TLSCAFile	Полный путь к файлу, содержащему сертификаты корневых CA для проверки сертификата другой стороны, используемый для шифрованного взаимодействия между компонентами Zabbix.
TLSCertFile	Полный путь к файлу, содержащему сертификат агента или цепочку сертификатов, используемый для шифрованного взаимодействия между компонентами Zabbix.
TLSConnect	Как агент должен подключаться к серверу Zabbix или прокси.
TLSCRLFile	Полный путь к файлу, содержащему отозванные сертификаты. Этот параметр используется для шифрованного взаимодействия между компонентами Zabbix.
TLSKeyFile	Полный путь к файлу, содержащему закрытый ключ агента, используемый для шифрованного взаимодействия между компонентами Zabbix.
TLSPSKFile	Полный путь к файлу, содержащему предварительно разделяемый ключ агента, используемый для шифрованного взаимодействия с сервером Zabbix.
TLSPSKIdentity	Строка идентификатора предварительно разделяемого ключа, используемая для шифрованного взаимодействия с сервером Zabbix.
TLSServerCertIssuer	Допустимый издатель сертификата сервера (прокси).
TLSServerCertSubject	Допустимый субъект сертификата сервера (прокси).
UnsafeUserParameters	Разрешить передачу любых символов в аргументах пользовательских параметров.
UserParameter	Пользовательский параметр для мониторинга.
UserParameterDir	Путь поиска по умолчанию для команд UserParameter.

Все параметры являются необязательными, если явно не указано, что параметр обязателен.

Обратите внимание:

- Значения по умолчанию отражают значения демона по умолчанию, 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 параметров

Псевдоним

Задает псевдоним для ключа элемента данных. Его можно использовать для замены длинного и сложного ключа элемента данных более коротким и простым.
 Может присутствовать 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 параметрах *HostnameItem* или *PerfCounter*.

Пример 1: Получение использования 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 CPU 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 CPU с параметрами по умолчанию, а также **cpu.load[percpu,avg15]** для получения конкретных данных о загрузке CPU.

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, использующих **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 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. Параметры обрабатываются по одному в порядке их появления. См. также: Ограничение проверок агента.

BufferSend

Не хранить данные в буфере дольше N секунд.

По умолчанию: 5
 Диапазон: 1-3600

BufferSize

Максимальное количество значений в буфере памяти. Агент отправит все собранные данные на сервер Zabbix или прокси, если буфер заполнен.

По умолчанию: 100
 Диапазон: 2-65535

DebugLevel

Укажите уровень отладки:
0 — основная 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 ещё 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 знаками, поддерживающее символ "*" для сопоставления с любым количеством любых символов.
Можно определить несколько правил сопоставления ключей в сочетании с AllowKey. Параметры обрабатываются по одному в порядке их появления. См. также: Ограничение проверок агента.

EnableRemoteCommands

Разрешены ли удалённые команды от сервера Zabbix. Этот параметр **устарел**; вместо него используйте AllowKey=system.run[*] или DenyKey=system.run[*].
Это внутренний псевдоним для параметров AllowKey/DenyKey в зависимости от значения:
0 - DenyKey=system.run[*]
1 - AllowKey=system.run[*]

По умолчанию: 0
 Значения: 0 — не разрешать, 1 — разрешать

HeartbeatFrequency

Частота сообщений heartbeat в секундах. Используется для мониторинга доступности активных проверок.
0 — сообщения heartbeat отключены.

По умолчанию: 60
 Диапазон: 0-3600

HostInterface

Необязательный параметр, который определяет интерфейс узла сети (IP-адрес или DNS-имя), используемый в процессе авторегистрации. Это значение будет использовано для заполнения интерфейса во вновь созданном узле сети и позволяет явно настроить либо IP-адрес, либо DNS-адрес. Подробнее см. в разделе Использование DNS в качестве интерфейса по умолчанию.

Если не определено, значение будет получено из HostInterfaceItem.

Агент выдаст ошибку и не запустится, если значение превышает ограничение в 255 символов.

Диапазон: 0-255 символов

HostInterfaceItem

Необязательный параметр, который определяет элемент данных, используемый для определения интерфейса узла сети (IP-адреса или DNS-имени) в процессе авторегистрации узла сети. Это значение используется только в том случае, если HostInterface не определён. Подробнее см. в разделе Использование DNS в качестве интерфейса по умолчанию.

Во время запроса авторегистрации агент запишет предупреждающее сообщение, если значение, возвращаемое указанным элементом данных, превышает ограничение в 255 символов.

Элемент данных system.run[] поддерживается независимо от настроек AllowKey/DenyKey.

HostMetadata

Необязательный параметр, который определяет метаданные, используемые для идентификации или различения узла сети в процессе авторегистрации 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 является строкой не в кодировке UTF-8. Если параметр ожидает IP-адрес или DNS-имя, значения, которые являются корректными UTF-8, но не являются корректными IP-адресами или DNS-именами, также будут отклонены и помечены как недопустимые.

Многострочные метаданные не поддерживаются — вывод будет обрезан на первом разрыве строки.

Диапазон: 0-2034 байт

HostMetadataItem

Необязательный параметр, который определяет элемент данных агента Zabbix, используемый для получения метаданных узла сети. Этот параметр используется только в том случае, если HostMetadata не задан.

Значение HostMetadataItem извлекается при каждой попытке авторегистрации и используется только в процессе авторегистрации узла сети (активный агент). HostMetadataItem позволяет различать узлы сети не только по имени узла сети.

Поддерживаются пользовательские параметры, счётчики производительности и псевдонимы. Элемент данных system.run[] поддерживается независимо от настроек AllowKey/DenyKey.

Во время запроса авторегистрации агент будет записывать предупреждающее сообщение в журнал, если значение, возвращаемое указанным элементом данных, превышает ограничение в 65535 кодовых точек UTF-8. Значение, возвращаемое элементом данных, должно быть строкой UTF-8, в противном случае оно будет проигнорировано. Если параметр ожидает IP-адрес или DNS-имя, значения, которые являются допустимыми UTF-8, но не являются допустимыми IP-адресами или DNS-именами, также будут отклонены и отмечены как недопустимые.

Многострочные метаданные не поддерживаются — вывод будет обрезан на первом разрыве строки.

Hostname

Список уникальных, чувствительных к регистру имен узлов сети, разделенных запятыми. Обязательно для активных проверок и должно совпадать с именами узлов сети, настроенными на сервере. Если значение не задано, оно получается из HostnameItem.
Допустимые символы: буквы и цифры, '.', ' ', '_' и '-'. Максимальная длина: 128 символов для каждого имени узла сети, 2048 символов для всей строки.

По умолчанию: задается параметром HostnameItem

HostnameItem

Необязательный параметр, который определяет элемент данных агента Zabbix, используемый для получения имени узла сети. Этот параметр используется только в том случае, если не определён Hostname. Пользовательские параметры, счётчики производительности или псевдонимы не поддерживаются, однако элемент данных system.run[] поддерживается независимо от значений AllowKey/DenyKey.
См. также более подробное описание.

По умолчанию: `system.hostname`

Include

Вы можете включать отдельные файлы или все файлы в каталоге в файл конфигурации (по умолчанию он расположен в `C:\Program Files\Zabbix Agent`, если Zabbix агент установлен с использованием пакетов установщика Windows MSI; расположен в папке, указанной во время установки, если Zabbix агент установлен как zip-архив). Все включаемые файлы должны иметь корректный синтаксис, в противном случае агент не запустится.
Чтобы включать только нужные файлы из указанного каталога, для сопоставления с шаблоном поддерживается символ подстановки звёздочка.
См. специальные примечания об ограничениях.

Пример:

```
Include=C:\Program Files\Zabbix Agent\zabbix_agentd.d\*.conf
```

ListenBacklog

Максимальное количество ожидающих соединений в очереди TCP.
Значение по умолчанию — это жестко заданная константа, которая зависит от системы.
Максимально поддерживаемое значение также зависит от системы; слишком большие значения могут быть молча усечены до «максимума, определяемого реализацией».

По умолчанию: `SOMAXCONN`
 Диапазон: 0 - INT_MAX

ListenIP

Список IP-адресов, разделённых запятыми, на которых агент должен прослушивать соединения.

По умолчанию: `0.0.0.0`

ListenPort

Агент будет прослушивать этот порт для соединений от сервера.

По умолчанию: 10050
 Диапазон: 1024-32767

LogFile

Имя файла журнала агента.

По умолчанию: `c:\zabbix_agentd.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 достигнут предельный размер файла журнала и ротация файла по какой-либо причине не удалась, существующий файл журнала усекается и начинается заново.

По умолчанию: 1
 Диапазон: 0-1024

LogRemoteCommands

Включает журналирование выполняемых shell-команд как предупреждений. Команды будут записываться в журнал, только если они выполняются удалённо. Записи журнала не будут создаваться, если system.run[] запускается локально параметрами HostMetadataItem, HostInterfaceItem или HostnameItem.

По умолчанию: 0
 Значения: 0 — отключено, 1 — включено

LogType

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

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 указано в *MaxLinesPerSecond*, чтобы найти требуемую строку в элементах данных журнала.

По умолчанию: 20
 Диапазон: 1-1000

PerfCounter

Определяет новый параметр <parameter_name>, который является средним значением системного счётчика производительности <perf_counter_path> за указанный период времени <period> (в секундах).
Синтаксис: <parameter_name>,"<perf_counter_path>",

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 счётчику производительности. Имя параметра (interrupts) следует использовать как ключ элемента данных при создании элемента данных. Выборки для вычисления среднего значения будут выполняться каждую секунду.
Вы можете выполнить команду "typeperf -qx", чтобы получить список всех счётчиков производительности, доступных в Windows.

PerfCounterEn

Определяет новый параметр <parameter_name>, который является средним значением системного счётчика производительности <perf_counter_path> за указанный период времени <period> (в секундах). По сравнению с PerfCounter, пути счётчиков производительности в 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 количество 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 счётчика производительности. Имя параметра (interrupts) следует использовать 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

Как часто обновляется список 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

Список IP-адресов, разделённых запятыми, при необходимости в нотации CIDR, или DNS-имён серверов Zabbix или прокси Zabbix. 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' 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 объявлены устаревшими в RFC4291. Пробелы допускаются.

Пример:

```
Server=127.0.0.1,192.168.1.0/24,::1,2001:db8::/32,zabbix.example.com
```

Обязательный: да, если StartAgents явно не установлен в 0

ServerActive

Адрес сервера/прокси Zabbix или конфигурация кластера, откуда получать активные проверки. Адрес сервера/прокси — это IP-адрес или DNS-имя и необязательный порт, разделённые двоеточием.
Конфигурация кластера представляет собой один или несколько адресов участников группы сервера или прокси, разделённых точкой с запятой. Можно указать несколько серверов/кластеров Zabbix и прокси Zabbix, разделив их запятыми. Если не используются группы прокси, не следует указывать более одного прокси Zabbix для каждого сервера/кластера Zabbix. Если указан прокси Zabbix, то сервер/кластер Zabbix для этого прокси указывать не следует.
Для параллельного использования нескольких независимых серверов Zabbix можно указать несколько адресов, разделённых запятыми. Пробелы допускаются.
Если порт не указан, используется порт по умолчанию.
Адреса IPv6 должны быть заключены в квадратные скобки, если для этого узла сети указан порт. Если порт не указан, квадратные скобки для адресов IPv6 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: (*)

SourceIP

IP-адрес источника для:

- исходящих соединений с сервером Zabbix или прокси Zabbix;
- установления соединений при выполнении некоторых элементов данных (web.page.get, net.tcp.port и т. д.).

StartAgents

Количество предварительно созданных экземпляров zabbix_agentd, которые обрабатывают пассивные проверки. Если установлено значение 0, пассивные проверки отключаются, и агент не будет прослушивать ни один TCP-порт.

По умолчанию: 10
 Диапазон: 0-100 (*)

Timeout

Указывает, как долго ждать (в секундах) установления соединения и обмена данными с Zabbix прокси или сервером.

Этот параметр определяет длительность различных операций обмена данными, включая:

- ожидание ответа от Zabbix сервера;
- отправку запросов на Zabbix сервер, включая запросы конфигурации элементов данных и данные элементов данных в active checks;
- получение данных журналов при мониторинге logfile или журнала событий Windows;
- отправку heartbeat-сообщений;
- максимальную длительность проверок `vfs.*`;
- использование модулями Zabbix агента;
- использование в качестве резервного значения в сценариях, когда сервер или прокси старше версии 7.0 отправляет проверки без тайм-аутов.

Этот тайм-аут **не** будет использоваться для тех проверок агента, для которых в веб-интерфейсе можно настроить параметры тайм-аута (на глобальном уровне, на уровне прокси или отдельного элемента данных).

По умолчанию: 3
 Диапазон: 1-30

TLSAccept

Входящие соединения, которые следует принимать. Используется для пассивных проверок. Можно указать несколько значений, разделённых запятой:
unencrypted — принимать соединения без шифрования (по умолчанию)
psk — принимать соединения с TLS и предварительно разделённым ключом (PSK)
cert — принимать соединения с TLS и сертификатом

Обязательно: да, если определены параметры TLS-сертификата или PSK (даже для *unencrypted* соединения); в противном случае — нет

TLSCAFile

Полный путь к файлу, содержащему сертификаты корневых центров сертификации (CA) для проверки сертификата другой стороны, используемый для шифрованного обмена данными между компонентами Zabbix.

TLSCertFile

Полный путь к файлу, содержащему сертификат агента или цепочку сертификатов, используемому для шифрованного обмена данными с компонентами Zabbix.

TLSConnect

Как агент должен подключаться 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

Полный путь к файлу, содержащему отозванные сертификаты. Этот параметр используется для шифрованного обмена данными между компонентами Zabbix.

TLSKeyFile

Полный путь к файлу, содержащему закрытый ключ агента, используемый для шифрованного обмена данными между компонентами Zabbix.

TLSPSKFile

Полный путь к файлу, содержащему предварительно разделяемый ключ агента, используемый для зашифрованного обмена данными с сервером Zabbix.

TLSPSKIdentity

Строка идентификатора предварительно общего ключа, используемая для шифрованного обмена данными с сервером Zabbix.

TLSServerCertIssuer

Разрешённый издатель сертификата сервера (прокси).

TLSServerCertSubject

Разрешённый субъект сертификата сервера (прокси).

UnsafeUserParameters

Разрешить передачу всех символов в аргументах пользовательских параметров. Следующие символы не допускаются: \ ' " ` * ? [ ] { } ~ $! & ; ( ) < > | # @ %
Кроме того, не допускаются символы новой строки.

По умолчанию: 0
 Значения: 0 — не разрешать, 1 — разрешать

UserParameter

Определяемый пользователем параметр для мониторинга. Таких параметров, определяемых пользователем, может быть несколько.
Формат: UserParameter=<key>,<shell command>
Обратите внимание, что shell-команда не должна возвращать пустую строку или только символ конца строки. Shell-команды могут использовать относительные пути, если указан параметр UserParameterDir.

Пример:

```
UserParameter=system.test,dir /b | find /c /v ""
UserParameter=check_cpu,.\custom\_script.bat
```

UserParameterDir

Путь поиска по умолчанию для команд UserParameter. Если 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
```

Note:
(*) Количество активных серверов, перечисленных в ServerActive, плюс количество заранее созданных экземпляров для пассивных проверок, указанных в StartAgents, должно быть меньше 64.

6 Zabbix агент 2 (Windows)

Обзор

Zabbix агент 2 — это новое поколение Zabbix агента, которое может использоваться вместо Zabbix агента.

Параметры, поддерживаемые файлом конфигурации Windows Zabbix агента 2 (zabbix_agent2.conf), перечислены в этом разделе.

Параметры перечислены без дополнительной информации. Нажмите на параметр, чтобы увидеть полные сведения.

Parameter	Description
Alias	Задает псевдоним для ключа элемента данных.
AllowKey	Разрешает выполнение тех ключей элементов данных, которые соответствуют шаблону.
BufferSend	Не хранить данные в буфере дольше N секунд.
BufferSize	Максимальное количество значений в буфере памяти.
ControlSocket	Управляющий сокет, используемый для отправки команд времени выполнения с опцией '-R'.
DebugLevel	Уровень отладки.
DenyKey	Запрещает выполнение тех ключей элементов данных, которые соответствуют шаблону.
EnablePersistentBuffer	Включает использование локального постоянного хранилища для активных элементов данных.
ForceActiveChecksOnStart	Выполняет активные проверки сразу после перезапуска для первой полученной конфигурации.
HeartbeatFrequency	Частота heartbeat-сообщений в секундах.
HostInterface	Необязательный параметр, определяющий интерфейс узла сети.
HostInterfaceItem	Необязательный параметр, определяющий элемент данных, используемый для получения интерфейса узла сети.
HostMetadata	Необязательный параметр, определяющий метаданные узла сети.
HostMetadataItem	Необязательный параметр, определяющий элемент данных Zabbix агента, используемый для получения метаданных узла сети.
Hostname	Необязательный параметр, определяющий имя узла сети.
HostnameItem	Необязательный параметр, определяющий элемент данных Zabbix агента, используемый для получения имени узла сети.
Include	В файл конфигурации можно включать отдельные файлы или все файлы из каталога.

Parameter	Description
ListenIP	Список IP-адресов, разделенных запятыми, на которых агент должен прослушивать соединения.
ListenPort	Агент будет прослушивать этот порт для соединений от сервера.
LogFile	Имя файла журнала.
LogFileSize	Максимальный размер файла журнала.
LogType	Тип вывода журнала.
PersistentBufferFile	Файл, в котором Zabbix агент 2 должен хранить базу данных SQLite.
PersistentBufferPeriod	Период времени, в течение которого данные должны храниться при отсутствии соединения с сервером или прокси.
Plugins.<PluginName>.System.Capacity	Предел количества проверок на плагин, которые могут выполняться одновременно.
Plugins.Log.MaxLinesPerSecond	Максимальное количество новых строк, которое агент будет отправлять в секунду на Zabbix сервер или прокси при обработке активных проверок ’log’ и ’logrt’.
Plugins.SystemRun.LogRemoteCommands	Включает журналирование выполняемых shell-команд как предупреждений.
PluginSocket	Путь к UNIX-сокету для взаимодействия с загружаемыми плагинами.
PluginTimeout	Тайм-аут соединений с загружаемыми плагинами в секундах.
PerfCounter	Определяет новый параметр <parameter_name>, который является средним значением системного счетчика производительности <perf_counter_path> за указанный период времени <period> (в секундах).
PerfCounterEn	Определяет новый параметр <parameter_name>, который является средним значением системного счетчика производительности <perf_counter_path> за указанный период времени <period> (в секундах). По сравнению с PerfCounter, пути perfcounter должны быть на английском языке.
RefreshActiveChecks	Как часто обновляется список активных проверок.
Server	Список IP-адресов, разделенных запятыми, при необходимости в нотации CIDR, или DNS-имен Zabbix серверов и Zabbix прокси.
ServerActive	Адрес Zabbix сервера/прокси или конфигурация кластера для получения активных проверок.
SourceIP	Исходный IP-адрес.
StatusPort	Если задан, агент будет прослушивать этот порт для HTTP-запросов состояния (http://localhost:<port>/status).
Timeout	Указывает, как долго ждать (в секундах) установления соединения и обмена данными с Zabbix прокси или сервером.
TLSAccept	Какие входящие соединения принимать.
TLSCAFile	Полный путь к файлу, содержащему сертификаты корневых CA для проверки сертификата удаленной стороны, используемый для шифрованного взаимодействия между компонентами Zabbix.
TLSCertFile	Полный путь к файлу, содержащему сертификат агента или цепочку сертификатов, используемый для шифрованного взаимодействия между компонентами Zabbix.
TLSCipherAll13	Строка шифров OpenSSL (TLS 1.3). Переопределяет критерии выбора набора шифров по умолчанию для шифрования на основе сертификатов и PSK.
TLSCipherCert13	Строка шифров OpenSSL (TLS 1.3). Переопределяет критерии выбора набора шифров по умолчанию для шифрования на основе сертификатов.
TLSCipherPSK13	Строка шифров OpenSSL (TLS 1.3). Переопределяет критерии выбора набора шифров по умолчанию для шифрования на основе PSK.
TLSConnect	Как агент должен подключаться к Zabbix серверу или прокси.
TLSCRLFile	Полный путь к файлу, содержащему отозванные сертификаты. Этот параметр используется для шифрованного взаимодействия между компонентами Zabbix.
TLSKeyFile	Полный путь к файлу, содержащему закрытый ключ агента, используемый для шифрованного взаимодействия между компонентами Zabbix.
TLSPSKFile	Полный путь к файлу, содержащему предварительно общий ключ агента, используемый для шифрованного взаимодействия с Zabbix сервером.
TLSPSKIdentity	Строка идентификатора предварительно общего ключа, используемая для шифрованного взаимодействия с Zabbix сервером.
TLSServerCertIssuer	Разрешенный издатель сертификата сервера (прокси).
TLSServerCertSubject	Разрешенный субъект сертификата сервера (прокси).
UnsafeUserParameters	Разрешает передавать любые символы в аргументах пользовательских параметров.
UserParameter	Пользовательский параметр для мониторинга.
UserParameterDir	Путь поиска по умолчанию для команд UserParameter.

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 окружения;
- Zabbix поддерживает файлы конфигурации только в кодировке UTF-8 без BOM;
- Комментарии, начинающиеся с ”#”, поддерживаются только в начале строки.

Подробности параметра

Псевдоним

Задает псевдоним для ключа элемента данных. Его можно использовать для замены длинного и сложного ключа элемента данных более коротким и простым.
 Может присутствовать 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 процента использования файла подкачки 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 CPU 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 CPU с параметрами по умолчанию, а также использовать **cpu.load[percpu,avg15]** для получения конкретных данных о загрузке CPU.

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

Разрешить выполнение тех ключей элементов данных, которые соответствуют шаблону. Шаблон ключа представляет собой выражение с подстановочными знаками, поддерживающее символ ”*” для сопоставления с любым количеством любых символов.
Можно определить несколько правил сопоставления ключей в сочетании с DenyKey. Параметры обрабатываются по одному в порядке их появления. См. также: Ограничение проверок агента.

BufferSend

Временной интервал в секундах, который определяет, как часто значения отправляются из буфера на сервер Zabbix.
Обратите внимание, что если буфер заполнен, данные будут отправлены раньше.

По умолчанию: 5
 Диапазон: 1-3600

BufferSize

Максимальное количество значений в буфере памяти. Агент отправит все собранные данные на сервер Zabbix или прокси, если буфер заполнен.
Этот параметр следует использовать только в том случае, если постоянный буфер отключен (*EnablePersistentBuffer=0*).

По умолчанию: 1000
 Диапазон: 2-65535

ControlSocket

Управляющий сокет, используемый для отправки команд времени выполнения с опцией ’-R’.

По умолчанию: `\\.\pipe\agent.sock`

DebugLevel

Укажите уровень отладки:
0 - основная информация о запуске и остановке процессов Zabbix
1 - критическая информация;
2 - информация об ошибках;
3 - предупреждения;
4 — для отладки (выдает много информации);
5 — расширенная отладка (выдает 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 знаками, которое поддерживает символ "*" для соответствия любому количеству любых символов.
Можно определить несколько правил сопоставления ключей в сочетании с AllowKey. Параметры обрабатываются по одному в порядке их появления. См. также: Ограничение проверок агента.

EnablePersistentBuffer

Включает использование локального постоянного хранилища для активных элементов данных. Если постоянное хранилище отключено, будет использоваться буфер памяти.

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

Частота сообщений heartbeat в секундах. Используется для мониторинга доступности активных проверок.
0 — сообщения heartbeat отключены.

По умолчанию: 60
 Диапазон: 0-3600

HostInterface

Необязательный параметр, который определяет интерфейс узла сети (IP-адрес или DNS-имя), используемый в процессе авторегистрации. Это значение будет использовано для заполнения интерфейса во вновь созданном узле сети и позволяет явно настроить либо IP-адрес, либо DNS-адрес. Подробнее см. в разделе Использование DNS в качестве интерфейса по умолчанию.

Если параметр не задан, значение будет получено из HostInterfaceItem.

Агент выдаст ошибку и не запустится, если значение превышает ограничение в 255 символов.

Диапазон: 0-255 символов

HostInterfaceItem

Необязательный параметр, который определяет элемент данных, используемый для определения интерфейса узла сети (IP-адреса или DNS-имени) в процессе авторегистрации узла сети. Это значение используется только в том случае, если HostInterface не определён. Подробнее см. в разделе Использование DNS в качестве интерфейса по умолчанию.

Во время запроса авторегистрации агент запишет предупреждающее сообщение, если значение, возвращаемое указанным элементом данных, превышает ограничение в 255 символов.

Элемент данных system.run[] поддерживается независимо от настроек AllowKey/DenyKey.

HostMetadata

Необязательный параметр, который определяет метаданные, используемые для идентификации или различения узла сети в процессе авторегистрации узла сети (активный агент). HostMetadata позволяет различать узлы сети не только по имени узла сети.

Если не определено, значение будет получено из HostMetadataItem.

Агент выдаст ошибку и не запустится, если указанное значение превышает ограничение в 2034 байта или является строкой не в UTF-8. Если параметр ожидает IP-адрес или DNS-имя, значения, которые являются допустимым UTF-8, но не являются допустимыми IP-адресами или DNS-именами, также будут отклонены и помечены как недопустимые.

Многострочные метаданные не поддерживаются — вывод будет обрезан на первом разрыве строки.

Диапазон: 0-2034 байт

HostMetadataItem

Необязательный параметр, который определяет элемент данных агента Zabbix, используемый для получения метаданных узла сети. Этот параметр используется только в том случае, если HostMetadata не задан.

Значение HostMetadataItem извлекается при каждой попытке авторегистрации и используется только в процессе авторегистрации узла сети (активный агент). HostMetadataItem позволяет различать узлы сети не только по имени узла сети.

Пользовательские параметры и псевдонимы поддерживаются. Элемент данных system.run[] поддерживается независимо от настроек AllowKey/DenyKey.

Во время запроса авторегистрации агент записывает предупреждающее сообщение в журнал, если значение, возвращаемое указанным элементом данных, превышает ограничение в 65535 кодовых точек UTF-8. Значение, возвращаемое элементом данных, должно быть строкой UTF-8, в противном случае оно будет проигнорировано. Если параметр ожидает IP-адрес или DNS-имя, значения, которые являются допустимыми UTF-8, но не являются допустимыми IP-адресами или DNS-именами, также будут отклонены и отмечены как недопустимые.

Многострочные метаданные не поддерживаются — вывод будет обрезан на первом разрыве строки.

Hostname

Список уникальных, чувствительных к регистру имен узлов сети, разделенных запятыми. Обязательно для активных проверок и должно совпадать с именами узлов сети, настроенными на сервере. Если значение не определено, оно берется из HostnameItem.
Допустимые символы: буквы и цифры, '.', ' ', '_' и '-'. Максимальная длина: 128 символов для каждого имени узла сети, 2048 символов для всей строки.

По умолчанию: задается параметром HostnameItem

HostnameItem

Необязательный параметр, который определяет элемент данных, используемый для получения имени узла сети. Этот параметр используется только в том случае, если Hostname не задан. Пользовательские параметры или псевдонимы не поддерживаются, однако элемент данных system.run[] поддерживается независимо от значений AllowKey/DenyKey.

По умолчанию: `system.hostname`

Include

Вы можете включать отдельные файлы или все файлы в каталоге в файл конфигурации (по умолчанию он расположен в `C:\Program Files\Zabbix Agent 2`, если агент Zabbix установлен с помощью пакетов установщика Windows MSI; расположен в папке, указанной во время установки, если агент Zabbix установлен как zip-архив). Все включаемые файлы должны иметь корректный синтаксис, в противном случае агент не запустится. Путь может быть относительным по отношению к расположению файла *zabbix_agent2.conf* (например, `Include=.\zabbix_agent2.d\plugins.d\*.conf`).
Чтобы включать только подходящие файлы в указанном каталоге, для сопоставления с шаблоном поддерживается символ подстановки звёздочка.
См. специальные примечания об ограничениях.

Пример:

```
Include=C:\Program Files\Zabbix Agent2\zabbix_agent2.d\*.conf
```

ListenIP

Список IP-адресов, разделённых запятыми, которые агент должен прослушивать. Первый IP-адрес отправляется на сервер Zabbix при подключении к нему для получения списка активных проверок.

По умолчанию: `0.0.0.0`

ListenPort

Агент будет прослушивать этот порт на предмет соединений с сервером.

По умолчанию: 10050
 Диапазон: 1024-32767

LogFile

Имя файла журнала агента.

По умолчанию: `c:\zabbix_agent2.log`
 Обязательно: Да, если LogType установлен в *file*; в противном случае нет

LogFileSize

Максимальный размер файла журнала в МБ.
0 — отключить автоматическую ротацию журналов.
Примечание: если достигнут предел размера файла журнала и ротация файла по какой-либо причине не удалась, существующий файл журнала усекается и запускается заново.

По умолчанию: 1
 Диапазон: 0–1024

LogType

Тип вывода журнала:
file - запись журнала в файл, указанный параметром LogFile;
console - запись журнала в стандартный вывод.

По умолчанию:

PersistentBufferFile

Файл, в котором Zabbix агент 2 должен хранить базу данных SQLite. Должен быть указан полный путь к файлу. Этот параметр используется только в том случае, если включен постоянный буфер (*EnablePersistentBuffer=1*).

PersistentBufferPeriod

Период времени, в течение которого данные должны храниться при отсутствии соединения с сервером или прокси. Более старые данные будут потеряны. Данные журнала будут сохранены. Этот параметр используется только в том случае, если включен постоянный буфер (*EnablePersistentBuffer=1*).

По умолчанию: 1h
 Диапазон: 1m-365d

Plugins.<Имя плагина>.System.Capacity

Предел проверок для каждого плагина < >, который может выполняться одновременно.

По умолчанию: 1000 Диапазон: 1-1000

Plugins.Log.MaxLinesPerSecond

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 указано в *MaxLinesPerSecond*, чтобы найти требуемую строку в элементах данных журнала.

По умолчанию: 20
 Диапазон: 1-1000

Plugins.SystemRun.LogRemoteCommands

Включить журналирование выполняемых команд оболочки как предупреждений. Команды будут записываться в журнал только в случае удалённого выполнения. Записи журнала не будут создаваться, если system.run[] запускается локально параметрами HostMetadataItem, HostInterfaceItem или HostnameItem.

По умолчанию: 0
 Значения: 0 — отключено, 1 — включено

PluginSocket

Путь к сокету UNIX для связи с загружаемым плагином.

По умолчанию: `\\.\pipe\agent.plugin.sock`

PluginTimeout

Таймаут соединений с загружаемыми плагинами, в секундах.

По умолчанию: `Timeout`
 Диапазон: 1-30

PerfCounter

Определяет новый параметр <parameter_name>, который является средним значением системного счётчика производительности <perf_counter_path> за указанный период времени <period> (в секундах).
Синтаксис: <parameter_name>,"<perf_counter_path>",

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 счётчику производительности. Имя параметра (interrupts) должно использоваться как ключ элемента данных при создании элемента данных. Выборки для вычисления среднего значения будут выполняться каждую секунду.
Вы можете выполнить команду "typeperf -qx", чтобы получить список всех счётчиков производительности, доступных в Windows.

PerfCounterEn

Определяет новый параметр <parameter_name>, который является средним значением системного счётчика производительности <perf_counter_path> за указанный период времени <period> (в секундах). По сравнению с PerfCounter, пути perf-counter 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 количество 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 счётчику производительности. Имя параметра (interrupts) следует использовать как ключ элемента данных при создании элемента данных. Выборки для вычисления среднего значения будут выполняться каждую секунду.
Список английских строк можно найти, просмотрев следующий раздел реестра: `HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows NT\CurrentVersion\Perflib\009`.

RefreshActiveChecks

Как часто обновляется список 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

Список IP-адресов, разделённых запятыми, при необходимости в нотации CIDR, или DNS-имён серверов Zabbix или прокси Zabbix. 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' разрешит 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 отключаются, 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

Адрес сервера/прокси Zabbix или конфигурация кластера для получения активных проверок. Адрес сервера/прокси — это IP-адрес или DNS-имя и необязательный порт, разделённые двоеточием.
Конфигурация кластера — это один или несколько адресов участников группы сервера или прокси, разделённых точкой с запятой. Можно указать несколько серверов/кластеров Zabbix и прокси Zabbix, разделённых запятой. Если не используются группы прокси, не следует указывать более одного прокси Zabbix для каждого сервера/кластера Zabbix. Если указан прокси Zabbix, то сервер/кластер Zabbix для этого прокси указывать не следует.
Можно указать несколько адресов, разделённых запятыми, чтобы использовать несколько независимых серверов Zabbix параллельно. Пробелы допускаются.
Если порт не указан, используется порт по умолчанию.
Адреса IPv6 должны быть заключены в квадратные скобки, если для этого узла сети указан порт. Если порт не указан, квадратные скобки для адресов IPv6 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

IP-адрес источника для:

- исходящих соединений с сервером Zabbix или прокси Zabbix;
- установления соединений при выполнении некоторых элементов данных (web.page.get, net.tcp.port и т. д.).

StatusPort

Если установлено, агент будет прослушивать этот порт на предмет запросов статуса HTTP (http://localhost:<port>/status).

Диапазон: 1024-32767

Timeout

Указывает, как долго ждать (в секундах) установления соединения и обмена данными с прокси или сервером Zabbix.

Этот параметр определяет длительность различных операций обмена данными, 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 журналов при мониторинге logfile или журнала событий Windows;
- отправку heartbeat-сообщений;
- максимальную длительность проверок `vfs.*`;

- использование в качестве резервного значения в сценариях, когда сервер или прокси версии ниже 7.0 отправляет проверки без тайм-аутов.

Этот тайм-аут **не** будет использоваться для тех проверок агента, для которых настраиваются параметры тайм-аута в веб-интерфейсе (на глобальном уровне, на уровне прокси или отдельного элемента данных).

По умолчанию: 3
 Диапазон: 1-30

TLSAccept

Входящие соединения, которые следует принимать. Используется для пассивных проверок. Можно указать несколько значений, разделённых запятой:
unencrypted — принимать соединения без шифрования (по умолчанию)
psk — принимать соединения с TLS и предварительно общим ключом (PSK)
cert — принимать соединения с TLS и сертификатом

Обязательно: да, если определены параметры TLS-сертификата или PSK (даже для соединения *unencrypted*); в противном случае — нет

######TLSCAFile

Полный путь к файлу, содержащему сертификаты ЦС верхнего уровня для проверки одноранговых сертификатов, используемый для зашифрованной связи между компонентами Zabbix.

TLSCertFile

Полный путь к файлу, содержащему сертификат агента или цепочку сертификатов, используемый для зашифрованной связи с компонентами Zabbix.

TLSCipherAll13

Строка 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

Строка 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 соединения по сертификату (`TLSConnect=cert`) используйте вместо этого TLSCipherAll13.

TLSCipherPSK13

Строка 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

Как агент должен подключаться 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

Полный путь к файлу, содержащему отозванные сертификаты. Этот параметр используется для шифрованного обмена данными между компонентами Zabbix.

TLSKeyFile

Полный путь к файлу, содержащему закрытый ключ агента, используемый для зашифрованной связи между компонентами Zabbix.

######TLSPSKFile

Полный путь к файлу, содержащему предварительный ключ агента, используемый для зашифрованной связи с сервером Zabbix.

TLSPSKIdentity

Строка идентификации предварительного общего ключа, используемая для зашифрованной связи с сервером Zabbix.

TLSServerCertIssuer

Разрешенный эмитент сертификата сервера (прокси).

TLSServerCertSubject

Разрешенный субъект сертификата сервера (прокси).

UnsafeUserParameters

Разрешить передачу всех символов в аргументах пользовательских параметров. Следующие символы не разрешены: \ ' " ` * ? [ ] { } ~ $! & ; ( ) < > | # @ %
Кроме того, символы новой строки не разрешены.

По умолчанию: 0
 Значения: 0 — не разрешать, 1 — разрешать

UserParameter

Определяемый пользователем параметр для мониторинга. Может быть несколько определяемых пользователем параметров.
Формат: UserParameter=<key>,<shell command>
Обратите внимание, что shell-команда не должна возвращать пустую строку или только EOL. Shell-команды могут иметь относительные пути, если указан параметр UserParameterDir.

Пример:

```
UserParameter=system.test,dir /b | find /c /v ""
UserParameter=check_cpu,.\custom\_script.bat
```

UserParameterDir

Путь поиска по умолчанию для команд UserParameter. Если используется, агент изменит свой рабочий каталог на указанный здесь перед выполнением команды. Таким образом, команды UserParameter могут иметь относительный префикс .\ вместо полного пути.
Допускается только одна запись.

Пример:

```
UserParameterDir=C:\opt\myscripts
```

7 Configuration parameters of plugins

See configuration parameters for Zabbix agent 2 plugins:

- Ceph plugin
- Docker plugin
- Ember+ plugin
- Memcached plugin
- Modbus plugin
- MongoDB plugin
- MQTT plugin
- MSSQL plugin
- MySQL plugin
- NVIDIA GPU plugin
- Oracle plugin
- PostgreSQL plugin
- Redis plugin
- S.M.A.R.T. plugin

Plugin configuration principles

This page describes plugin configuration principles and best practices.

All plugins are configured using *Plugins.** parameter, which can either be part of the Zabbix agent 2 configuration file or a plugin's own configuration file. If a plugin uses a separate configuration file, path to this file should be specified in the Include parameter of Zabbix agent 2 configuration file.

A typical plugin parameter has the following structure:

Plugins.<PluginName>.<Parameter>=<Value>

Additionally, there are two specific groups of parameters:

- *Plugins.<PluginName>.Default.<Parameter>=<Value>* used for defining default parameter values.
- *Plugins.<PluginName>.<SessionName>.<Parameter>=<Value>* used for defining separate sets of parameters for different monitoring targets via named sessions.

All parameter names should adhere to the following requirements:

- it is recommended to capitalize the names of your plugins
- the parameter should be capitalized
- special characters are not allowed
- nesting isn't limited by a maximum level
- the number of parameters is not limited

For example, to perform active checks that do not have *Scheduling* update interval immediately after the agent restart only for the Uptime plugin, set `Plugins.Uptime.System.ForceActiveChecksOnStart=1` in the configuration file. Similarly, to set custom limit for concurrent checks for the CPU plugin, set the `Plugins.CPU.System.Capacity=N` in the configuration file.

Default values

You can set default values for the connection-related parameters (URI, username, password, etc.) in the configuration file in the format:

```
Plugins.<PluginName>.Default.<Parameter>=<Value>
```

For example, `Plugins.Mysql.Default.Username=zabbix`, `Plugins.MongoDB.Default.Uri=tcp://127.0.0.1:27017`, etc.

If a value for such parameter is not provided in an item key or in the named session parameters, the plugin will use the default value. If a default parameter is also undefined, hardcoded defaults will be used.

> **Note:**
> If an item key does not have any parameters, Zabbix agent 2 will attempt to collect the metric using values defined in the default parameters section.

Named sessions

Named sessions represent an additional level of plugin parameters and can be used to specify separate sets of authentication parameters for each of the instances being monitored. Each named session parameter should have the following structure:

```
Plugins.<PluginName>.Sessions.<SessionName>.<Parameter>=<Value>
```

A session name can be used as a connString item key parameter instead of specifying a URI, username, and/or password separately.

In item keys, the first parameter can be either a connString or a URI. If the first key parameter doesn't match any session name, it will be treated as a URI. Note that passing embedded URI credentials in the item key is not supported, use named session parameters instead.

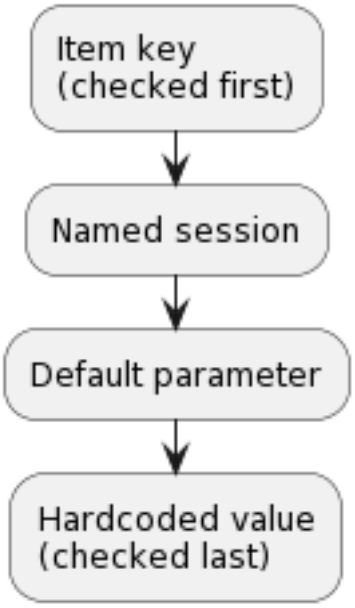
The list of available named session parameters depends on the plugin.

It is possible to override session parameters by specifying new values in the item key parameters (see example).

If a parameter is not defined for the named session, Zabbix agent 2 will use the value defined in the default plugin parameter.

Parameter priority

Zabbix agent 2 plugins search for connection-related parameter values in the following order:



1. The first item key parameter is compared to session names. If no match is found, it is treated as an actual value; in this case, step 3 will be skipped. If a match is found, the parameter value (usually, a URI) must be defined in the named session.
2. Other parameters will be taken from the item key if defined.
3. If an item key parameter (for example, password) is empty, plugin will look for the corresponding named session parameter.
4. If the session parameter is also not specified, the value defined in the corresponding default parameter will be used.
5. If all else fails, the plugin will use the hardcoded default value.

Example 1

Monitoring of two instances “MySQL1” and “MySQL2”.

Configuration parameters:

```
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://192.0.2.0:3306
Plugins.Mysql.Sessions.MySQL2.User=mysql2_user
Plugins.Mysql.Sessions.MySQL2.Password=different_password
```

As a result of this configuration, each session name may be used as a connString in an item key, e.g., `mysql.ping[MySQL1]` or `mysql.ping[MySQL2]`.

Example 2

Providing some of the parameters in the item key.

Configuration parameters:

```
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
```

Item key: `pgsql.ping[session1,new_username,,postgres]`

As a result of this configuration, the agent will connect to PostgreSQL using the following parameters:

- URI from session parameter: `192.0.2.234:5432`
- Username from the item key: `new_username`
- Password from session parameter (since it is omitted in the item key): `session_password`
- Database name from the item key: `postgres`

Example 3

Collecting a metric using default configuration parameters.

Configuration parameters:

```
Plugins.PostgreSQL.Default.Uri=tcp://192.0.2.234:5432
Plugins.PostgreSQL.Default.User=zabbix
Plugins.PostgreSQL.Default.Password=password
```

Item key: `pgsql.ping[,,,postgres]`

As a result of this configuration, the agent will connect to PostgreSQL using the parameters:

- Default URI: `192.0.2.234:5432`
- Default username: `zabbix`
- Default password: `password`
- Database name from the item key: `postgres`

Connections

Some plugins support gathering metrics from multiple instances simultaneously. Both local and remote instances can be monitored. TCP and Unix-socket connections are supported.

It is recommended to configure plugins to keep connections to instances in an open state. The benefits are reduced network congestion, latency, and CPU and memory usage due to the lower number of connections. The client library takes care of this.

> **Note:**
> Time period for which unused connections should remain open can be determined by *Plugins.<PluginName>.KeepAlive* parameter. Example: *Plugins.Memcached.KeepAlive*

1 Ceph plugin

Overview

The configuration file of Zabbix agent 2 is used to configure plugins. These Zabbix agent 2 configuration parameters are supported for operating the Ceph plugin.

It is recommended to specify them in their own configuration file (e.g. `ceph.conf`) and then use the Include directive for adding this file to the Zabbix agent 2 configuration.

Note that:

- The default values reflect process defaults, not the values in the shipped configuration files;
- Values support environment variables;
- Zabbix supports configuration files only in UTF-8 encoding without BOM;
- Comments starting with "#" are only supported at the beginning of the line.

Parameters

Parameter	Mandatory	Range	Default	Description
Plugins.Ceph.Default.ApiKey				Default API key for connecting to Ceph; used if no value is specified in an item key or named session.
Plugins.Ceph.Default.Mode		native / restful	restful (deprecated)	Default mode for connecting to Ceph. The native mode is only supported on Linux and starting with Ceph 16. The restful mode will not work with Ceph version 20 or newer. Note that the user credential set differs for each mode and they are not compatible with each other.
Plugins.Ceph.Default.User				Default username for connecting to Ceph; used if no value is specified in an item key or named session.
Plugins.Ceph.Default.Uri			https://localhost:8003	Default URI for connecting to Ceph; used if no value is specified in an item key or named session. Should not include embedded credentials (they will be ignored). Must match the URI format. Only `https` scheme is supported; a scheme can be omitted. A port can be omitted (default=8003). Examples: `https://127.0.0.1:8003` `localhost`

Parameter	Mandatory	Range	Default	Description
Plugins.Ceph.InsecureSkipVerify		false / true	false	Determines whether an http client should verify the server's certificate chain and host name. If *true*, TLS accepts any certificate presented by the server and any host name in that certificate. In this mode, TLS is susceptible to man-in-the-middle attacks (should be used only for testing).
Plugins.Ceph.KeepAlive		60-900	300	The maximum time of waiting (in seconds) before unused plugin connections are closed.
Plugins.Ceph.Sessions.<SessionName>.ApiKey				Named session API key. **<SessionName>** - define name of a session for using in item keys.
Plugins.Ceph.Sessions.<SessionName>.Mode		native / restful	restful (deprecated)	Named session mode for connecting to Ceph. The native mode is only supported on Linux and starting with Ceph 16. The restful mode will not work with Ceph version 20 or newer. Note that the user credential set differs for each mode and they are not compatible with each other.
Plugins.Ceph.Sessions.<SessionName>.User				Named session username. **<SessionName>** - define name of a session for using in item keys.
Plugins.Ceph.Sessions.<SessionName>.Uri				Connection string of a named session. **<SessionName>** - define name of a session for using in item keys. Should not include embedded credentials (they will be ignored). Must match the URI format. Only `https` scheme is supported; a scheme can be omitted. A port can be omitted (default=8003). Examples: `https://127.0.0.1:8003` `localhost`
Plugins.Ceph.Timeout		1-30	global timeout	Request execution timeout (the duration, in seconds, to wait for a request to complete before shutting it down).

See also:

- Description of general Zabbix agent 2 configuration parameters: Zabbix agent 2 (UNIX) / Zabbix agent 2 (Windows)
- Instructions for configuring plugins

2 Docker plugin

Overview

The configuration file of Zabbix agent 2 is used to configure plugins. These Zabbix agent 2 configuration parameters are supported for operating the Docker plugin.

It is recommended to specify them in their own configuration file (e.g. `docker.conf`) and then use the Include directive for adding this file to the Zabbix agent 2 configuration.

Note that:

- The default values reflect process defaults, not the values in the shipped configuration files;
- Values support environment variables;
- Zabbix supports configuration files only in UTF-8 encoding without BOM;
- Comments starting with "#" are only supported at the beginning of the line.

Parameters

Parameter	Mandatory	Range	Default	Description
Plugins.Docker.Endpoint			unix:///var/run/docker.sock	Docker daemon unix-socket location. Must contain a scheme (only `unix://` is supported).

Parameter	Mandatory	Range	Default	Description
Plugins.Docker.Timeout		1-30	global timeout	Request execution timeout (the duration, in seconds, to wait for a request to complete before shutting it down).

See also:

- Description of general Zabbix agent 2 configuration parameters: Zabbix agent 2 (UNIX) / Zabbix agent 2 (Windows)
- Instructions for configuring plugins

3 Ember+ plugin

Overview

The configuration file of Zabbix agent 2 is used to configure plugins. These Zabbix agent 2 configuration parameters are supported for operating the Ember+ plugin.

It is recommended to specify them in their own configuration file (e.g. `emberplus.conf`) and then use the Include directive for adding this file to the Zabbix agent 2 configuration.

The Ember+ plugin is a loadable plugin and is available and fully described in the Ember+ plugin repository.

This plugin is currently only available to be built from the source (for both Unix and Windows).

Note that:

- The default values reflect process defaults, not the values in the shipped configuration files.
- Values support environment variables;
- Zabbix supports configuration files only in UTF-8 encoding without BOM.
- Comments starting with "#" are only supported at the beginning of the line.

Options

Parameter	Description
-V --version	Print the plugin version and license information.
-h --help	Print help information (shorthand).
-t, --test <item key>	Launch plugin for testing (plugin config ignored).

Parameters

Parameter	Mandatory	Range	Default	Description
Plugins.EmberPlus.Default.Uri			tcp://localhost:9998	The default URI to connect. The only supported schema is `tcp://`. A schema can be omitted. Embedded credentials will be ignored.
Plugins.EmberPlus.KeepAlive		60-900	300	The maximum time of waiting (in seconds) before unused plugin connections are closed.
Plugins.EmberPlus.Sessions.<SessionName>.Uri			tcp://localhost:9998	The URI to connect, for the named session. The only supported schema is `tcp://`. A schema can be omitted. Embedded credentials will be ignored. **<SessionName>** - define name of a session for using in item keys.
Plugins.EmberPlus.System.Path				Path to the Ember+ plugin executable. Example usage: `Plugins.EmberPlus.System.Path=/usr/libexec/zabbix/zabbix`
Plugins.EmberPlus.Timeout		1-30	global timeout	The duration, in seconds, to wait for a server to respond when first connecting and on follow-up operations in the session.

See also:

- Description of general Zabbix agent 2 configuration parameters: Zabbix agent 2 (UNIX) / Zabbix agent 2 (Windows)
- Instructions for configuring plugins

4 Memcached plugin

Overview

The configuration file of Zabbix agent 2 is used to configure plugins. These Zabbix agent 2 configuration parameters are supported for operating the Memcached plugin.

It is recommended to specify them in their own configuration file (e.g. `memcached.conf`) and then use the Include directive for adding this file to the Zabbix agent 2 configuration.

Note that:

- The default values reflect process defaults, not the values in the shipped configuration files;
- Values support environment variables;
- Zabbix supports configuration files only in UTF-8 encoding without BOM;
- Comments starting with "#" are only supported at the beginning of the line.

Parameters

Parameter	Mandatory	Range	Default	Description
Plugins.Memcached.Default.Password				Default password for connecting to Memcached; used if no value is specified in an item key or named session.

Plugins.Memcached.Default.Uri			tcp://localhost:11211	Default URI for connecting to Memcached; used if no value is specified in an item key or named session.
				
				Should not include embedded credentials (they will be ignored).
				Must match the URI format.
				Supported schemes: `tcp`, `unix`; a scheme can be omitted.
				A port can be omitted (default=11211).
				Examples: `tcp://localhost:11211`
				`localhost`
				`unix:/var/run/memcached.sock`

Plugins.Memcached.Default.User				Default username for connecting to Memcached; used if no value is specified in an item key or named session.
Plugins.Memcached.KeepAlive		60-900	300	The maximum time of waiting (in seconds) before unused plugin connections are closed.
Plugins.Memcached.Sessions.<SessionName>.Password				Named session password. **<SessionName>** - define name of a session for using in item keys.

Plugins.Memcached.Sessions.<SessionName>.Uri				Connection string of a named session.
				<SessionName> - define name of a session for using in item keys.
				
				Should not include embedded credentials (they will be ignored).
				Must match the URI format.
				Supported schemes: `tcp`, `unix`; a scheme can be omitted.
				A port can be omitted (default=11211).
				Examples: `tcp://localhost:11211`
				`localhost`
				`unix:/var/run/memcached.sock`

Plugins.Memcached.Sessions.<SessionName>.User				Named session username. **<SessionName>** - define name of a session for using in item keys.
Plugins.Memcached.Timeout		1-30	global timeout	Request execution timeout (the duration, in seconds, to wait for a request to complete before shutting it down).

See also:

- Description of general Zabbix agent 2 configuration parameters: Zabbix agent 2 (UNIX) / Zabbix agent 2 (Windows)
- Instructions for configuring plugins

5 Modbus plugin

Overview

The configuration file of Zabbix agent 2 is used to configure plugins. These Zabbix agent 2 configuration parameters are supported for operating the Modbus plugin.

It is recommended to specify them in their own configuration file (e.g. `modbus.conf`) and then use the Include directive for adding this file to the Zabbix agent 2 configuration.

Note that:

- The default values reflect process defaults, not the values in the shipped configuration files;
- Values support environment variables;
- Zabbix supports configuration files only in UTF-8 encoding without BOM;
- Comments starting with "#" are only supported at the beginning of the line.

Parameters

Parameter	Mandatory	Range	Default	Description
Plugins.Modbus.Sessions.<SessionName>.Endpoint				Endpoint is a connection string consisting of a protocol scheme, a host address and a port or serial port name and attributes. **<SessionName>** - define name of a session for using in item keys.
Plugins.Modbus.Sessions.<SessionName>.SlaveID				Slave ID of a named session. **<SessionName>** - define name of a session for using in item keys. Example: `Plugins.Modbus.Sessions.MB1.SlaveID=20` *Note* that this named session parameter is checked only if the value provided in the item key slave ID parameter is empty.
Plugins.Modbus.Sessions.<SessionName>.Timeout				Timeout of a named session in seconds. **<SessionName>** - define name of a session for using in item keys. Example: `Plugins.Modbus.Sessions.MB1.Timeout=2`

If you need to set the request execution timeout (how long to wait for a request to complete before shutting it down), use the item configuration form.

See also:

- Description of general Zabbix agent 2 configuration parameters: Zabbix agent 2 (UNIX) / Zabbix agent 2 (Windows)
- Instructions for configuring plugins

6 MongoDB plugin

Overview

The configuration file of Zabbix agent 2 is used to configure plugins. These Zabbix agent 2 configuration parameters are supported for operating the MongoDB plugin.

It is recommended to specify them in their own configuration file (e.g., `mongo.conf`) and then use the Include directive for adding this file to the Zabbix agent 2 configuration.

The MongoDB plugin is a loadable plugin and is available and fully described in the MongoDB plugin repository.

Note that:

- The default values reflect process defaults, not the values in the shipped configuration files.
- Values support environment variables.
- Zabbix supports configuration files only in UTF-8 encoding without BOM.
- Comments starting with "#" are only supported at the beginning of the line.

Options

Parameter	Description
-V --version	Print the plugin version and license information.
-h --help	Print help information (shorthand).
-t, --test <item key>	Launch plugin for testing (plugin config ignored).

Parameters

When connecting to MongoDB, the plugin checks its configuration in a specific order, which determines which value to use. The order, from highest to lowest priority, is as follows:

1. Parameters specified in the connection URI (e.g., query parameters defined after ? in `Plugins.MongoDB.Sessions.<session_name` have the highest priority).
2. Parameters in the Zabbix agent 2 item key (e.g., mongodb.collection.stats).
3. Parameters defined in a named session (e.g., `Plugins.MongoDB.Sessions.<SessionName>.User`, `Plugins.MongoDB.Sessions`
4. Default values (e.g., `Plugins.MongoDB.Default.User`, `Plugins.MongoDB.Default.Password`).

For example:

- If you specify the database for authenticating a connection to the MongoDB server in the `*.<SessionName>.Uri` parameter (e.g., `Plugins.MongoDB.Sessions.<session_name>.Uri=mongodb://user:password@127.0.0.1:27017/?authSource=a` then the plugin will consider the `*.<SessionName>.AuthSource` parameter to be set to `admin`, even if the `*.<SessionName>.AuthSource` parameter is set to a different value.
- If you specify the URI in the `*.<SessionName>.Uri` parameter (e.g., `Plugins.MongoDB.Sessions.<session_name>.Uri=mongo` but provide the username and the password in the Zabbix agent 2 item key, the plugin will use the username and password from the item key, even if `Plugins.MongoDB.Sessions.<SessionName>.User` and `Plugins.MongoDB.Sessions.<SessionNam` are configured.

Attention:
If the `mongodb+srv://` scheme is used, the URI may retrieve parameters from a DNS TXT record. Parameters obtained from this TXT record override all other configurations, including URI parameters, Zabbix Agent 2 item key parameters, and session parameters.
 Only one TXT record may exist for the hostname used in the connection string.

Parameter	Mandatory	Range	Default	Description
Plugins.MongoDB.System.Path	No			Path to the MongoDB plugin executable. Example usage: `Plugins.MongoDB.System.Path=/usr/libexec/zabbix/zabbix-ag`
Plugins.MongoDB.Timeout	No	1-30	global timeout	Request execution timeout (the duration, in seconds, to wait for a request to complete before shutting it down).
Plugins.MongoDB.KeepAlive	No	60-900	300	The maximum time of waiting (in seconds) before unused plugin connections are closed.
Plugins.MongoDB.Sessions.<SessionName>.Uri	No		`tcp://127.0.0.1:27017`	Connection string of a named session. **<SessionName>** - define name of a session for using in item keys. Must match the URI format. Supported schemes: `mongodb://` and `mongodb+srv://`; the `tcp://` scheme is deprecated and kept for backward compatibility with existing configurations. A port can be omitted (default=27017). Examples: `mongodb://127.0.0.1:27017`, `mongodb+srv://example.com`, `localhost`.
Plugins.MongoDB.Sessions.<SessionName>.User	No			Named session username. **<SessionName>** - define name of a session for using in item keys.
Plugins.MongoDB.Sessions.<SessionName>.Password	No			Named session password. **<SessionName>** - define name of a session for using in item keys.

Parameter	Mandatory	Range	Default	Description
Plugins.MongoDB.Sessions.<SessionName>.Direct			false	Connection method used to determine whether automatic discovery of MongoDB replica set nodes should be enabled or disabled. **<SessionName>** - define name of a session for using in item keys. Supported values: `true` - the plugin will connect only to the specified server, ignoring other members of the replica set; `false` - the plugin will automatically discover all nodes in the replica set. If the `tcp://` (deprecated) scheme is used, the parameter is set to `true` by default.
Plugins.MongoDB.Sessions.<SessionName>.AuthMechanism				Authentication mechanism for connecting to MongoDB server. **<SessionName>** - define name of a session for using in item keys. Supported values: `SCRAM-SHA-1`, `SCRAM-SHA-256`, and `MONGODB-X509`.
Plugins.MongoDB.Sessions.<SessionName>.AuthSource			admin	Database for authenticating connection to MongoDB server. **<SessionName>** - define name of a session for using in item keys. If the `MONGODB-X509` authentication mechanism is used, this parameter is automatically set to `$external`; modifying it to any other value will cause an error when launching.
Plugins.MongoDB.Sessions.<SessionName>.ReplicaSet				Replica set name, used for automatic discovery of MongoDB replica set nodes. **<SessionName>** - define name of a session for using in item keys. If this parameter is used with the `mongodb+srv://` scheme, it will behave as if multiple URLs have been provided; a DNS server may have a TXT record that can override this (or any other) parameter. Unavailable if the `Plugins.MongoDB.Sessions.<SessionName>.Direct` parameter is set to `true`, the agent will fail to start.
Plugins.MongoDB.Sessions.<SessionName>.ReadPreference				Order for connecting to MongoDB replica set nodes or fallback. **<SessionName>** - define name of a session for using in item keys. Supported values: `PrimaryMode`, `PrimaryPreferredMode`, `SecondaryMode`, `SecondaryPreferredMode`, and `NearestMode`. Unavailable if the `Plugins.MongoDB.Sessions.<SessionName>.Direct` parameter is set to `true`, the agent will fail to start.
Plugins.MongoDB.Sessions.<SessionName>.TLSConnect				Encryption type for communications between Zabbix agent 2 and monitored databases. **<SessionName>** - define name of a session for using in item keys. Supported values: `required` - require TLS connection; `verify\_ca` - verify certificates; `verify\_full` - verify certificates and IP address. Supported since plugin version 1.2.1.

Parameter	Mandatory	Range	Default	Description
Plugins.MongoDB.Sessions.<SessionName>.TLSCAFile	Yes, if Plugins.MongoDB.Sessions.<SessionName>.TLSConnect is set to one of: verify_ca, verify_full			Full pathname of a file containing the top-level CA(s) certificates for peer certificate verification, used for encrypted communications between Zabbix agent 2 and monitored databases. **<SessionName>** - define name of a session for using in item keys.
Plugins.MongoDB.Sessions.<SessionName>.TLSCertFile	Yes, if Plugins.MongoDB.Sessions.<SessionName>.TLSKeyFile is specified			Full pathname of a file containing the agent certificate or certificate chain, used for encrypted communications between Zabbix agent 2 and monitored databases. **<SessionName>** - define name of a session for using in item keys.
Plugins.MongoDB.Sessions.<SessionName>.TLSKeyFile	Yes, if Plugins.MongoDB.Sessions.<SessionName>.TLSCertFile is specified			Full pathname of a file containing the database private key used for encrypted communications between Zabbix agent 2 and monitored databases. **<SessionName>** - define name of a session for using in item keys.
Plugins.MongoDB.Default.Uri			`tcp://127.0.0.1:27017`	Default URI for connecting to MongoDB. Must match the URI format. Supported schemes: `mongodb://` and `mongodb+srv://`. Deprecated scheme: `tcp://` (used as the default for backward compatibility with existing configurations). A port can be omitted (default=27017). Examples: `mongodb://127.0.0.1:27017`, `mongodb+srv://example.com`, `localhost`.
Plugins.MongoDB.Default.User				Default username for connecting to MongoDB.
Plugins.MongoDB.Default.Password				Default password for connecting to MongoDB.
Plugins.MongoDB.Default.Direct			`false`	Default connection method used to determine whether automatic discovery of MongoDB replica set nodes should be enabled or disabled. Supported values: `true` - the plugin will connect only to the specified server, ignoring other members of the replica set; `false` - the plugin will automatically discover all nodes in the replica set. If the `tcp://` (deprecated) scheme is used, the parameter is set to `true` by default.
Plugins.MongoDB.Default.AuthMechanism				Default authentication mechanism for connecting to MongoDB server. Supported values: `SCRAM-SHA-1`, `SCRAM-SHA-256`, and `MONGODB-X509`. If no value is specified either here, in an item key, in a named session, or in the URI, a SCRAM-based mechanism is used (if a username is provided; otherwise, no authentication will be used).
Plugins.MongoDB.Default.AuthSource			`admin`	Default database for authenticating connection to MongoDB server. If the `MONGODB-X509` authentication mechanism is used, this parameter is automatically set to `$external`; modifying it to any other value will cause an error when launching.
Plugins.MongoDB.Default.ReplicaSet				Default replica set name; used for automatic discovery of MongoDB replica set nodes. If this parameter is used with the `mongodb+srv://` scheme, it will behave as if multiple URLs have been provided; a DNS server may have a TXT record that can override this (or any other) parameter. Unavailable if the `Plugins.MongoDB.Default.Direct` parameter is set to `true`, the agent will fail to start.

Parameter	Mandatory	Range	Default	Description
Plugins.MongoDB.Default.ReadPreference				Default order for connecting to MongoDB replica set nodes or fallback. Supported values: `PrimaryMode`, `PrimaryPreferredMode`, `SecondaryMode`, `SecondaryPreferredMode`, and `NearestMode`. Unavailable if the `Plugins.MongoDB.Default.Direct` parameter is set to `true`, the agent will fail to start.
Plugins.MongoDB.Default.TLSConnect				Default encryption type for communications between Zabbix agent 2 and monitored databases. Supported values: `required` - requires TLS connection. `verify\_ca` - verifies certificates. `verify\_full` - verifies certificates and IP address. If no value is specified either here, in an item key, in a named session, or in the URI, a TLS connection will not be established.
Plugins.MongoDB.Default.TLSCAFile				Default pathname of a file containing the top-level CA(s) certificates for peer certificate verification. Used for encrypted communications between Zabbix agent 2 and monitored databases.
Plugins.MongoDB.Default.TLSCertFile				Default pathname of a file containing the MongoDB client certificate or certificate chain. Used for encrypted communications between Zabbix agent 2 and monitored databases.
Plugins.MongoDB.Default.TLSKeyFile				Default pathname of a file containing the MongoDB client private key used for encrypted communications between Zabbix agent 2 and monitored databases.

See also:

- Description of general Zabbix agent 2 configuration parameters: Zabbix agent 2 (UNIX) / Zabbix agent 2 (Windows)
- Instructions for configuring plugins

7 MQTT plugin

Overview

The configuration file of Zabbix agent 2 is used to configure plugins. These Zabbix agent 2 configuration parameters are supported for operating the MQTT plugin.

It is recommended to specify them in their own configuration file (e.g. `mqtt.conf`) and then use the Include directive for adding this file to the Zabbix agent 2 configuration.

Note that:

- The default values reflect process defaults, not the values in the shipped configuration files;
- Values support environment variables;
- Zabbix supports configuration files only in UTF-8 encoding without BOM;
- Comments starting with "#" are only supported at the beginning of the line.

Parameters

Parameter	Mandatory	Range	Default	Description
Plugins.MQTT.Default.Password				Default password for connecting to MQTT; used if no value is specified in an item key or named session.
Plugins.MQTT.Default.TLSCAFile				Full pathname of a file containing the top-level CA(s) certificates for peer certificate verification for encrypted communications between Zabbix agent 2 and MQTT broker; used if no value is specified in a named session.

Parameter	Mandatory	Range	Default	Description
Plugins.MQTT.Default.TLSCertFile				Full pathname of a file containing the agent certificate or certificate chain for encrypted communications between Zabbix agent 2 and MQTT broker; used if no value is specified in a named session.
Plugins.MQTT.Default.TLSKeyFile				Full pathname of a file containing the MQTT private key for encrypted communications between Zabbix agent 2 and MQTT broker; used if no value is specified in a named session.

Plugins.MQTT.Default.Topic				Default topic for MQTT subscription; used if no value is specified in an item key or named session.
				
				The topic may contain wildcards (”+”,”#”)
				Examples: `path/to/file`
				`path/to/#`
				`path/+/topic`

Plugins.MQTT.Default.Url			tcp://localhost:1883	Default MQTT broker connection string; used if no value is specified in an item key or named session.
				
				Should not include query parameters.
				Must match the URL format.
				Supported schemes: `tcp` (default), `ws`, `tls`; a scheme can be omitted.
				A port can be omitted (default=1883).
				Examples: `tcp://host:1883`
				`localhost`
				`ws://host:8080`

Plugins.MQTT.Default.User				Default username for connecting to MQTT; used if no value is specified in an item key or named session.
Plugins.MQTT.Sessions.<SessionName>.Password				Named session password. **<SessionName>** - define name of a session for using in item keys.
Plugins.MQTT.Sessions.<SessionName>.TLSCAFile				Full pathname of a file containing the top-level CA(s) certificates for peer certificate verification, used for encrypted communications between Zabbix agent 2 and MQTT broker. **<SessionName>** - define name of a session for using in item keys.
Plugins.MQTT.Sessions.<SessionName>.TLSCertFile				Full pathname of a file containing the agent certificate or certificate chain, used for encrypted communications between Zabbix agent 2 and MQTT broker. **<SessionName>** - define name of a session for using in item keys.
Plugins.MQTT.Sessions.<SessionName>.TLSKeyFile				Full pathname of a file containing the MQTT private key used for encrypted communications between Zabbix agent 2 and MQTT broker. **<SessionName>** - define name of a session for using in item keys.

Plugins.MQTT.Sessions.<SessionName>.Topic				Named session topic for MQTT subscription.
				<SessionName> - define name of a session for using in item keys.
				
				The topic may contain wildcards (”+”,”#”)
				Examples: `path/to/file`
				`path/to/#`
				`path/+/topic`

Parameter	Mandatory	Range	Default	Description

Plugins.MQTT.Sessions.<SessionName>.Url				Connection string of a named session.
				<SessionName> - define name of a session for using in item keys.
				
				Should not include query parameters.
				Must match the URL format.
				Supported schemes: `tcp` (default), `ws`, `tls`; a scheme can be omitted.
				A port can be omitted (default=1883).
				Examples: `tcp://host:1883`
				`localhost`
				`ws://host:8080`

Plugins.MQTT.Sessions.<SessionName>.User				Named session username. **<SessionName>** - define name of a session for using in item keys.

If you need to set the request execution timeout (how long to wait for a request to complete before shutting it down), use the item configuration form.

See also:

- Description of general Zabbix agent 2 configuration parameters: Zabbix agent 2 (UNIX) / Zabbix agent 2 (Windows)
- Instructions for configuring plugins

8 MSSQL plugin

Overview

The configuration file of Zabbix agent 2 is used to configure plugins. These Zabbix agent 2 configuration parameters are supported for operating the MSSQL plugin.

It is recommended to specify them in their own configuration file (e.g. `mssql.conf`) and then use the Include directive for adding this file to the Zabbix agent 2 configuration.

The MSSQL plugin is a loadable plugin and is available and fully described in the MSSQL plugin repository.

Note that:

- The default values reflect process defaults, not the values in the shipped configuration files.
- Values support environment variables;
- Zabbix supports configuration files only in UTF-8 encoding without BOM.
- Comments starting with "#" are only supported at the beginning of the line.

Options

Parameter	Description
-V --version	Print the plugin version and license information.
-h --help	Print help information (shorthand).
-t, --test <item key>	Launch plugin for testing (plugin config ignored).

Parameters

Parameter	Mandatory	Range	Default	Description
Plugins.MSSQL.CustomQueriesDir			`/usr/local/share/zabbix/custom-queries/mssql` for Unix systems `*:\Program Files\Zabbix Agent 2\Custom Queries\MSSQL` for Windows systems, where `*` is the drive name from the `ProgramFiles` environment variable	Specifies the path to a directory containing user-defined .sql files with custom queries that the plugin can execute. The plugin loads all available .sql files in the configured directory at startup. This means that any changes to the custom query files will not be reflected until the plugin is restarted. The plugin is started and stopped together with Zabbix agent 2.
Plugins.MSSQL.CustomQueriesEnabled			false	If set, enables the execution of the `mssql.custom.query` item key. If disabled, no queries will be loaded from the custom query directory path.
Plugins.MSSQL.Default.CACertPath				The default file path to the public key certificate of the certificate authority (CA) that issued the certificate of the MSSQL server. The certificate must be in PEM format.
Plugins.MSSQL.Default.Database				The default database name to connect to.
Plugins.MSSQL.Default.Encrypt				Specifies the default connection encryption type. Possible values are: *true* - data sending between plugin and server is encrypted; *false* - data sending between plugin and server is not encrypted beyond the login packet; *strict* - data sending between plugin and server is encrypted E2E using TDS8; *disable* - data sending between plugin and server is not encrypted.
Plugins.MSSQL.Default.HostNameInCertificate				The common name (CN) of the certificate of the MSSQL server by default.
Plugins.MSSQL.Default.Password				The password to be sent to a protected MSSQL server by default.
Plugins.MSSQL.Default.TLSMinVersion				The minimum TLS version to use by default. Possible values are: `1.0`, `1.1`, `1.2`, `1.3`.
Plugins.MSSQL.Default.TrustServerCertificate				Whether the plugin should trust the server certificate without validating it by default. Possible values: `true`, `false`.
Plugins.MSSQL.Default.Uri			sqlserver://localhost:1433	The default URI to connect. The only supported schema is `sqlserver://`. A schema can be omitted. Embedded credentials will be ignored.
Plugins.MSSQL.Default.User				The default username to be sent to a protected MSSQL server.
Plugins.MSSQL.KeepAlive		60-900	300	The maximum time of waiting (in seconds) before unused plugin connections are closed.
Plugins.MSSQL.Sessions.<SessionName>.CACertPath				The file path to the public key certificate of the certificate authority (CA) that issued the certificate of the MSSQL server for the named session. The certificate must be in PEM format. **<SessionName>** - define name of a session for using in item keys.
Plugins.MSSQL.Sessions.<SessionName>.Database				The database name to connect to for the named session. **<SessionName>** - define name of a session for using in item keys.

Parameter	Mandatory	Range	Default	Description
Plugins.MSSQL.Sessions.<SessionName>.Encrypt				Specifies the connection encryption type for the named session. Possible values are: *true* - data sending between plugin and server is encrypted; *false* - data sending between plugin and server is not encrypted beyond the login packet; *strict* - data sending between plugin and server is encrypted E2E using TDS8; *disable* - data sending between plugin and server is not encrypted. **<SessionName>** - define name of a session for using in item keys.
Plugins.MSSQL.Sessions.<SessionName>.HostNameInCertificate				The common name (CN) of the certificate of the MSSQL server for the named session. **<SessionName>** - define name of a session for using in item keys.
Plugins.MSSQL.Sessions.<SessionName>.Password				The password to be sent to a protected MSSQL server for the named session. **<SessionName>** - define name of a session for using in item keys.
Plugins.MSSQL.Sessions.<SessionName>.TLSMinVersion				The minimum TLS version to use for the named session. Possible values are: `1.0`, `1.1`, `1.2`, `1.3`. **<SessionName>** - define name of a session for using in item keys.
Plugins.MSSQL.Sessions.<SessionName>.TrustServerCertificate				Whether the plugin should trust the server certificate without validating it for the named session. Possible values: `true`, `false`. **<SessionName>** - define name of a session for using in item keys.
Plugins.MSSQL.Sessions.<SessionName>.Uri			sqlserver://localhost:1433	The URI to connect, for the named session. The only supported schema is `sqlserver://`. A schema can be omitted. Embedded credentials will be ignored. **<SessionName>** - define name of a session for using in item keys.
Plugins.MSSQL.Sessions.<SessionName>.User				The username to be sent to a protected MSSQL server for the named session. **<SessionName>** - define name of a session for using in item keys.
Plugins.MSSQL.System.Path				Path to the MSSQL plugin executable. Global setting for the MSSQL plugin. Applied to all connections. Example usage: `Plugins.MSSQL.System.Path=/usr/libexec/zabbix/zabbix-agen`
Plugins.MSSQL.Timeout		1-30	global timeout	The duration, in seconds, to wait for a server to respond when first connecting and on follow-up operations in the session.

See also:

- Description of general Zabbix agent 2 configuration parameters: Zabbix agent 2 (UNIX) / Zabbix agent 2 (Windows)
- Instructions for configuring plugins

9 MySQL plugin

Overview

The configuration file of Zabbix agent 2 is used to configure plugins. These Zabbix agent 2 configuration parameters are supported for operating the MySQL plugin.

It is recommended to specify them in their own configuration file (e.g. `mysql.conf`) and then use the Include directive for adding this file to the Zabbix agent 2 configuration.

Note that:

- The default values reflect process defaults, not the values in the shipped configuration files;

- Values support environment variables;
- Zabbix supports configuration files only in UTF-8 encoding without BOM;
- Comments starting with "#" are only supported at the beginning of the line.

Parameters

Parameter	Mandatory	Range	Default	Description
Plugins.Mysql.CallTimeout	no	1-30	global timeout	The maximum amount of time in seconds to wait for a request to be done.
Plugins.Mysql.CustomQueriesPath	no		`/usr/local/share/zabbix/custom-queries/mysql` for Unix systems `*:\Program Files\Zabbix Agent 2\Custom Queries\Mysql` for Windows systems, where `*` is the drive name from the `ProgramFiles` environment variable	Full pathname of a directory containing `.sql` files with custom queries.
Plugins.Mysql.CustomQueriesEnabled	no		false	If set, enables the execution of the `mysql.custom.query` item key. If disabled, no queries will be loaded from the custom query directory path.
Plugins.Mysql.Default.Password	no			Default password for connecting to MySQL; used if no value is specified in an item key or named session.
Plugins.Mysql.Default.TLSCAFile	no (yes, if Plugins.Mysql.Default.TLSConnect is set to *verify_ca* or *verify_full*)			Full pathname of a file containing the top-level CA(s) certificates for peer certificate verification for encrypted communications between Zabbix agent 2 and monitored databases; used if no value is specified in a named session.
Plugins.Mysql.Default.TLSCertFile	no (yes, if Plugins.Mysql.Default.TLSConnect is set to *verify_ca* or *verify_full*)			Full pathname of a file containing the agent certificate or certificate chain for encrypted communications between Zabbix agent 2 and monitored databases; used if no value is specified in a named session.
Plugins.Mysql.Default.TLSConnect	no			Encryption type for communications between Zabbix agent 2 and monitored databases; used if no value is specified in a named session. Supported values: *required* - require TLS connection; *verify_ca* - verify certificates; *verify_full* - verify certificates and IP address.
Plugins.Mysql.Default.TLSKeyFile	no (yes, if Plugins.Mysql.Default.TLSConnect is set to *verify_ca* or *verify_full*)			Full pathname of a file containing the database private key for encrypted communications between Zabbix agent 2 and monitored databases; used if no value is specified in a named session.

Parameter	Mandatory	Range	Default	Description

Plugins.Mysql.Default.Uri			tcp://localhost:3306	Default URI for connecting to MySQL; used if no value is specified in an item key or named session.
				
				Should not include embedded credentials (they will be ignored).
				Must match the URI format.
				Supported schemes: `tcp`, `unix`; a scheme can be omitted.
				A port can be omitted (default=3306).
				Examples: `tcp://localhost:3306`
				`localhost`
				`unix:/var/run/mysql.sock`

Plugins.Mysql.Default.User				Default username for connecting to MySQL; used if no value is specified in an item key or named session.
Plugins.Mysql.KeepAlive		60-900	300	The maximum time of waiting (in seconds) before unused plugin connections are closed.
Plugins.Mysql.Sessions.<SessionName>.Password				Named session password. **<SessionName>** - define name of a session for using in item keys.
Plugins.Mysql.Sessions.<SessionName>.TLSCAFile	no (yes, if Plugins.Mysql.Sessions.<SessionName>.TLSConnect is set to *verify_ca* or *verify_full*)			Full pathname of a file containing the top-level CA(s) certificates for peer certificate verification, used for encrypted communications between Zabbix agent 2 and monitored databases. **<SessionName>** - define name of a session for using in item keys.
Plugins.Mysql.Sessions.<SessionName>.TLSCertFile	yes, if Plugins.Mysql.Sessions.<SessionName>.TLSKeyFile is specified			Full pathname of a file containing the agent certificate or certificate chain, used for encrypted communications between Zabbix agent 2 and monitored databases. **<SessionName>** - define name of a session for using in item keys.
Plugins.Mysql.Sessions.<SessionName>.TLSConnect				Encryption type for communications between Zabbix agent 2 and monitored databases. **<SessionName>** - define name of a session for using in item keys. Supported values: *required* - require TLS connection; *verify_ca* - verify certificates; *verify_full* - verify certificates and IP address.
Plugins.Mysql.Sessions.<SessionName>.TLSKeyFile	yes, if Plugins.Mysql.Sessions.<SessionName>.TLSCertFile is specified			Full pathname of a file containing the database private key used for encrypted communications between Zabbix agent 2 and monitored databases. **<SessionName>** - define name of a session for using in item keys.

Plugins.Mysql.Sessions.<SessionName>.Uri				Connection string of a named session.
				<SessionName> - define name of a session for using in item keys.
				
				Should not include embedded credentials (they will be ignored).
				Must match the URI format.
				Supported schemes: `tcp`, `unix`; a scheme can be omitted.
				A port can be omitted (default=3306).
				Examples: `tcp://localhost:3306`
				`localhost`
				`unix:/var/run/mysql.sock`

Plugins.Mysql.Sessions.<SessionName>.User				Named session username. **<SessionName>** - define name of a session for using in item keys.
Plugins.Mysql.Timeout	no	1-30	global timeout	The maximum time in seconds for waiting when a connection has to be established.

See also:

- Description of general Zabbix agent 2 configuration parameters: Zabbix agent 2 (UNIX) / Zabbix agent 2 (Windows)
- Instructions for configuring plugins

10 NVIDIA GPU plugin

Overview

The configuration file of Zabbix agent 2 is used to configure plugins. These Zabbix agent 2 configuration parameters are supported for operating the NVIDIA GPU plugin.

It is recommended to specify them in their own configuration file (e.g. `nvidiagpu.conf`) and then use the Include directive for adding this file to the Zabbix agent 2 configuration.

The NVIDIA GPU plugin is a loadable plugin and is available and fully described in the NVIDIA GPU plugin repository.

Note that:

- The default values reflect process defaults, not the values in the shipped configuration files;
- Values support environment variables;
- Zabbix supports configuration files only in UTF-8 encoding without BOM;
- Comments starting with "#" are only supported at the beginning of the line.

Parameters

Parameter	Mandatory	Range	Default	Description
Plugins.NVIDIA.System.Path				Path to the NVIDIA GPU plugin executable. Example usage: `Plugins.NVIDIA.System.Path=/path/to/executable/nvidia`
Plugins.NVIDIA.Timeout		1-30	global timeout	Specifies the maximum time (in seconds) to wait for a server response during connection attempts and subsequent operations in the session. The global item-type timeout or individual item timeout will override this value if greater.

See also:

- Description of general Zabbix agent 2 configuration parameters: Zabbix agent 2 (UNIX) / Zabbix agent 2 (Windows)
- Instructions for configuring plugins

11 Oracle plugin

Overview

The configuration file of Zabbix agent 2 is used to configure plugins. These Zabbix agent 2 configuration parameters are supported for operating the Oracle plugin.

It is recommended to specify them in their own configuration file (e.g. `oracle.conf`) and then use the Include directive for adding this file to the Zabbix agent 2 configuration.

Note that:

- The default values reflect process defaults, not the values in the shipped configuration files.
- Values support environment variables.
- Zabbix supports configuration files only in UTF-8 encoding without BOM.
- Comments starting with "#" are only supported at the beginning of the line.

Parameters

Parameter	Mandatory	Range	Default	Description
Plugins.Oracle.CallTimeout		1-30	global timeout	The maximum wait time in seconds for a request to be completed.
Plugins.Oracle.ConnectTimeout		1-30	global timeout	The maximum wait time in seconds for a connection to be established.

Parameter	Mandatory	Range	Default	Description
Plugins.Oracle.CustomQueriesPath			/usr/local/share/zabbix/custom_queries/oracle for Unix systems *:\Program Files\Zabbix Agent 2\Custom Queries\Oracle for Windows systems, where * is the drive name from the ProgramFiles environment variable	Full pathname of a directory containing *.sql files with custom queries. Example: /etc/zabbix/oracle/sql
Plugins.Oracle.CustomQueriesEnabled			false	If set, enables the execution of the oracle.custom.query item key. If disabled, no queries will be loaded from the custom query directory path.
Plugins.Oracle.Default.Password				Default password for connecting to Oracle; used if no value is specified in an item key or named session.
Plugins.Oracle.Default.Service				Default service name for connecting to Oracle (SID is not supported); used if no value is specified in an item key or named session.
Plugins.Oracle.Default.Uri				Default URI for connecting to Oracle; used if no value is specified in an item key or named session. Should not include embedded credentials (they will be ignored). Must match the URI format. Only tcp and tcps schemes are supported; a scheme can be omitted. A port can be omitted (default=1521). It is also possible to specify the TNS key or TNS value as the connection string. The TNS value must be composed without whitespaces. Examples: tcp://127.0.0.1:1521 localhost zbx_tns_example (TNS key) (DESCRIPTION=(ADDRESS=(PROTOCOL=TCP)(HOST=localhost)(POR (TNS value)
Plugins.Oracle.Default.User				Default username for connecting to Oracle; used if no value is specified in an item key or named session.
Plugins.Oracle.KeepAlive		60-900	300	The maximum time of waiting (in seconds) before unused plugin connections are closed.
Plugins.Oracle.ResolveTNS		true / false	true	The option specifies how to interpret the connection string (ConnString) for connecting to the Oracle server. If it is set to false the default scheme and port will be added to ConnString (if absent). If the option is set to true, the default scheme and port will be omitted (unless explicitly specified in ConnString), and ConnString will be passed to the Oracle client as is. If the Oracle client finds ConnString in the tnsnames.ora file, the connection description found will be used to connect to the Oracle server.
Plugins.Oracle.Sessions.<SessionName>.Password				Named session password. **<SessionName>** - define name of a session for using in item keys.

Parameter	Mandatory	Range	Default	Description
Plugins.Oracle.Sessions.<SessionName>.Service				Named session service name to be used for connection (SID is not supported). **<SessionName>** - define name of a session for using in item keys.
Plugins.Oracle.Sessions.<SessionName>.Uri				Named session connection string for Oracle. **<SessionName>** - define name of a session for using in item keys. Should not include embedded credentials (they will be ignored). Must match the URI format. Only `tcp` and `tcps` schemes are supported; a scheme can be omitted. A port can be omitted (default=1521). It is also possible to specify the TNS key or TNS value as the connection string. The TNS value must be composed without whitespaces. Examples: `tcp://127.0.0.1:1521` `localhost` `zbx_tns_example` (TNS key) `(DESCRIPTION=(ADDRESS=(PROTOCOL=TCP)(HOST=localhost)(POR` (TNS value)
Plugins.Oracle.Sessions.<SessionName>.User				Named session username. **<SessionName>** - define name of a session for using in item keys.

See also:

- Description of general Zabbix agent 2 configuration parameters: Zabbix agent 2 (UNIX) / Zabbix agent 2 (Windows)
- Instructions for configuring plugins

12 PostgreSQL plugin

Overview

The configuration file of Zabbix agent 2 is used to configure plugins. These Zabbix agent 2 configuration parameters are supported for operating the PostgreSQL plugin.

It is recommended to specify them in their own configuration file (e.g. `postgresql.conf`) and then use the Include directive for adding this file to the Zabbix agent 2 configuration.

The PostgreSQL plugin is a loadable plugin and is available and fully described in the PostgreSQL plugin repository.

Note that:

- The default values reflect process defaults, not the values in the shipped configuration files.
- Values support environment variables;
- Zabbix supports configuration files only in UTF-8 encoding without BOM.
- Comments starting with "#" are only supported at the beginning of the line.

Options

Parameter	Description
-V --version	Print the plugin version and license information.
-h --help	Print help information (shorthand).
-t, --test <item key>	Launch plugin for testing (plugin config ignored).

Parameters

Parameter	Mandatory	Range	Default	Description
Plugins.PostgreSQL.Default.CacheMode			prepare	Cache mode for the PostgreSQL connection. Supported values: *prepare* (default) - will create prepared statements on the PostgreSQL server; *describe* - will use the anonymous prepared statement to describe a statement without creating a statement on the server. Note that "describe" is primarily useful when the environment does not allow prepared statements such as when running a connection pooler like PgBouncer.
Plugins.PostgreSQL.CallTimeout		1-30	global timeout	Maximum wait time (in seconds) for a request to be completed.
Plugins.PostgreSQL.CustomQueriesPath			`/usr/local/share/zabbix/custom_queries/postgresql` for Unix systems `*:\Program Files\Zabbix Agent 2\Custom Queries\PostgreSQL` for Windows systems, where `*` is the drive name from the `ProgramFiles` environment variable	Full pathname of a directory containing files with custom queries.
Plugins.PostgreSQL.CustomQueriesEnabled			false	If set, enables the execution of the `postgresql.custom.query` item key. If disabled, no queries will be loaded from the custom query directory path.
Plugins.PostgreSQL.Default.Database				Default database for connecting to PostgreSQL; used if no value is specified in an item key or named session.
Plugins.PostgreSQL.Default.Password				Default password for connecting to PostgreSQL; used if no value is specified in an item key or named session.
Plugins.PostgreSQL.Default.TLSCAFile	(yes, if Plugins.PostgreSQL.Default.TLSConnect is set to *verify_ca* or *verify_full*)			Full pathname of a file containing the top-level CA(s) certificate for peer certificate verification for encrypted communications between Zabbix agent 2 and monitored databases; used if no value is specified in a named session.
Plugins.PostgreSQL.Default.TLSCertFile	(yes, if Plugins.PostgreSQL.Default.TLSConnect is set to *verify_ca* or *verify_full*)			Full pathname of a file containing the PostgreSQL certificate or certificate chain for encrypted communications between Zabbix agent 2 and monitored databases; used if no value is specified in a named session.
Plugins.PostgreSQL.Default.TLSConnect				Encryption type for communications between Zabbix agent 2 and monitored databases; used if no value is specified in a named session. Supported values: *required* - connect using TLS as transport mode without identity checks; *verify_ca* - connect using TLS and verify certificate; *verify_full* - connect using TLS, verify certificate and verify that database identity (CN) specified by DBHost matches its certificate. Undefined encryption type means unencrypted connection.

Parameter	Mandatory	Range	Default	Description
Plugins.PostgreSQL.Default.TLSKeyFile	No (yes, if Plugins.PostgreSQL.Default.TLSConnect is set to *verify_ca* or *verify_full*)			Full pathname of a file containing the PostgreSQL private key for encrypted communications between Zabbix agent 2 and monitored databases; used if no value is specified in a named session.

Plugins.PostgreSQL.Default.Uri	No			Default URI for connecting to PostgreSQL; used if no value is specified in an item key or named session.
				
				Should not include embedded credentials (they will be ignored).
				Must match the URI format.
				Supported schemes: `tcp`, `unix`.
				Examples: `tcp://127.0.0.1:5432`
				`tcp://localhost`
				`unix:/var/run/postgresql/.s.PGSQL.5432`

Plugins.PostgreSQL.Default.User	No			Default username for connecting to PostgreSQL; used if no value is specified in an item key or named session.
Plugins.PostgreSQL.KeepAlive	No	60-900	300	Maximum time of waiting (in seconds) before unused plugin connections are closed.
Plugins.PostgreSQL.Sessions.<SessionName>.CachePrepareMode	No		prepare	Cache mode for the PostgreSQL connection. **<SessionName>** - define name of a session for using in item keys. Supported values: *prepare* (default) - will create prepared statements on the PostgreSQL server; *describe* - will use the anonymous prepared statement to describe a statement without creating a statement on the server. Note that ”describe” is primarily useful when the environment does not allow prepared statements such as when running a connection pooler like PgBouncer.
Plugins.PostgreSQL.Sessions.<SessionName>.Database	No			Database for session connection. **<SessionName>** - define name of a session for using in item keys.
Plugins.PostgreSQL.Sessions.<SessionName>.Password	No	Must match the password format.		Password for session connection. **<SessionName>** - define name of a session for using in item keys.
Plugins.PostgreSQL.Sessions.<SessionName>.TLSCAFile	No (yes, if Plugins.PostgreSQL.Sessions.<SessionName>.TLSConnect is set to *verify_ca* or *verify_full*)			Full pathname of a file containing the top-level CA(s) certificate peer certificate verification. **<SessionName>** - define name of a session for using in item keys.
Plugins.PostgreSQL.Sessions.<SessionName>.TLSCertFile	Yes, if Plugins.PostgreSQL.Sessions.<SessionName>.TLSKeyFile is specified			Full pathname of a file containing the PostgreSQL certificate or certificate chain. **<SessionName>** - define name of a session for using in item keys.
Plugins.PostgreSQL.Sessions.<SessionName>.TLSConnect	No			Encryption type for PostgreSQL connection. **<SessionName>** - define name of a session for using in item keys. Supported values: *required* - connect using TLS as transport mode without identity checks; *verify_ca* - connect using TLS and verify certificate; *verify_full* - connect using TLS, verify certificate and verify that database identity (CN) specified by DBHost matches its certificate. Undefined encryption type means unencrypted connection.

Parameter	Mandatory	Range	Default	Description
Plugins.PostgreSQL.Sessions.<SessionName>.TLSKeyFile	yes, if Plugins.PostgreSQL.Sessions.<SessionName>.TLSCertFile is specified			Full pathname of a file containing the PostgreSQL private key. **<SessionName>** - define name of a session for using in item keys.

Plugins.PostgreSQL.Sessions.<SessionName>.Uri				Connection string of a named session.
				<SessionName> - define name of a session for using in item keys.
				
				Should not include embedded credentials (they will be ignored).
				Must match the URI format.
				Supported schemes: `tcp`, `unix`.
				Examples: `tcp://127.0.0.1:5432`
				`tcp://localhost`
				`unix:/var/run/postgresql/.s.PGSQL.5432`

Plugins.PostgreSQL.Sessions.<SessionName>.User				Named session username. **<SessionName>** - define name of a session for using in item keys.
Plugins.PostgreSQL.System.Path	yes			Path to the PostgreSQL plugin executable. Example usage: `Plugins.PostgreSQL.System.Path=/usr/libexec/zabbix/zabbi`
Plugins.PostgreSQL.Timeout		1-30	global timeout	Request execution timeout (the duration, in seconds, to wait for a request to complete before shutting it down).

See also:

- Description of general Zabbix agent 2 configuration parameters: Zabbix agent 2 (UNIX) / Zabbix agent 2 (Windows)
- Instructions for configuring plugins

13 Redis plugin

Overview

The configuration file of Zabbix agent 2 is used to configure plugins. These Zabbix agent 2 configuration parameters are supported for operating the Redis plugin.

It is recommended to specify them in their own configuration file (e.g. `redis.conf`) and then use the Include directive for adding this file to the Zabbix agent 2 configuration.

Note that:

- The default values reflect process defaults, not the values in the shipped configuration files.
- Values support environment variables.
- Zabbix supports configuration files only in UTF-8 encoding without BOM.
- Comments starting with ”#” are only supported at the beginning of the line.

Parameters

Parameter	Mandatory	Range	Default	Description
Plugins.Redis.Default.Password				Default password for connecting to Redis; used if no value is specified in an item key or named session.
Plugins.Redis.Default.TLSConnect				Client (Zabbix agent 2) TLS verification requirement. Supported values: `required` - encrypted, unverified (only for testing) `verify_ca` - encrypted, server certificate `verify_full` - encrypted, server certificate and server name verified with SAN An empty or unset value is treated as no tls.
Plugins.Redis.Default.TLSCAFile				Default full pathname of a file containing the top-level CA(s) certificates.
Plugins.Redis.Default.TLSCertFile				Default full pathname of a file containing the agent’s certificate.

Parameter	Mandatory	Range	Default	Description
Plugins.Redis.Default.TLSKeyFile				Default full pathname of a file containing the agent's private key.

Plugins.Redis.Default.Uri			tcp://localhost:6379	Default URI for connecting to Redis; used if no value is specified in an item key or named session.
				
				Should not include embedded credentials (they will be ignored).
				Must match the URI format.
				Supported schemes: `tcp`, `unix`; a scheme can be omitted.
				A port can be omitted (default=6379).
				Examples: `tcp://localhost:6379`
				`localhost`
				`unix:/var/run/redis.sock`

Plugins.Redis.Default.User			default	Default user to send to the protected Redis server; used if no value is specified in an item key or named session.
Plugins.Redis.KeepAlive		60-900	300	The maximum time of waiting (in seconds) before unused plugin connections are closed.
Plugins.Redis.Sessions.<SessionName>.Password				Named session password. **<SessionName>** - define the session name to be used in item keys.
Plugins.Redis.Sessions.<SessionName>.TLSConnect				Client (Zabbix agent 2) TLS verification requirement. **<SessionName>** - define the session name to be used in item keys. Supported values: `required` - encrypted, unverified (only for testing) `verify_ca` - encrypted, server certificate `verify_full` - encrypted, server certificate and server name verified with SAN An empty or unset value is treated as no tls.
Plugins.Redis.Sessions.<SessionName>.TLSCAFile	mandatory (yes, if TLSConnect is `verify_ca` or `verify_full`)			Full pathname of a file containing the top-level CA(s) certificates for Redis server certificate verification. **<SessionName>** - define the session name to be used in item keys.
Plugins.Redis.Sessions.<SessionName>.TLSCertFile				Full pathname of a file containing the agent's certificate for client authentication. **<SessionName>** - define the session name to be used in item keys.
Plugins.Redis.Sessions.<SessionName>.TLSKeyFile				Full pathname of a file containing the agent's private key for client authentication. **<SessionName>** - define the session name to be used in item keys.

Plugins.Redis.Sessions.<SessionName>.Uri			localhost:6379	Connection string of a named session.
				<SessionName> - define the session name to be used in item keys.
				
				Should not include embedded credentials (they will be ignored).
				Must match the URI format.
				Supported schemes: `tcp`, `unix`; a scheme can be omitted.
				A port can be omitted (default=6379).
				Examples: `tcp://localhost:6379`
				`localhost`
				`unix:/var/run/redis.sock`

Plugins.Redis.Sessions.<SessionName>.User			default	User to send to the protected Redis server. **<SessionName>** - define the session name to be used in item keys.

Parameter	Mandatory	Range	Default	Description
Plugins.Redis.Timeout	no	1-30	global timeout	Request execution timeout (the duration, in seconds, to wait for a request to complete before shutting it down).

See also:

- Description of general Zabbix agent 2 configuration parameters: Zabbix agent 2 (UNIX) / Zabbix agent 2 (Windows)
- Instructions for configuring plugins

14 SMART plugin

Overview

The configuration file of Zabbix agent 2 is used to configure plugins. These Zabbix agent 2 configuration parameters are supported for operating the SMART plugin.

It is recommended to specify them in their own configuration file (e.g. `smart.conf`) and then use the Include directive for adding this file to the Zabbix agent 2 configuration.

Note that:

- The default values reflect process defaults, not the values in the shipped configuration files;
- Values support environment variables;
- The path to the `smartctl` executable must be set either by adding its directory to the system's `PATH` environment variable or by configuring `Plugins.Smart.Path`; this applies to both Linux and Windows;
- Zabbix supports configuration files only in UTF-8 encoding without BOM;
- Comments starting with "#" are only supported at the beginning of the line.

Parameters

Parameter	Mandatory	Range	Default	Description
Plugins.Smart.Path	no		smartctl	Path to the smartctl executable.
Plugins.Smart.Timeout	no	1-30	global timeout	Request execution timeout (the duration, in seconds, to wait for a request to complete before shutting it down).

See also:

- Description of general Zabbix agent 2 configuration parameters: Zabbix agent 2 (UNIX) / Zabbix agent 2 (Windows)
- Instructions for configuring plugins

8 Zabbix Java gateway

Если вы используете скрипты `startup.sh` и `shutdown.sh` для запуска Zabbix Java gateway, то можете указать необходимые параметры конфигурации в файле `settings.sh`. Скрипты запуска и остановки подключают файл настроек и выполняют преобразование переменных shell (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, на котором выполняется прослушивание.
LISTEN_PORT	zabbix.listenPort	no	1024-32767	10052	Порт, на котором выполняется прослушивание.

Variable	Property	Mandatory	Range	Default	Description
PID_FILE	zabbix.pidFile	no		/tmp/zabbix_java.pid	Имя PID-файла. Если параметр не указан, Zabbix Java gateway запускается как консольное приложение.
PROPERTIES_FILE	zabbix.propertiesFile	no			Имя файла свойств. Может использоваться для задания дополнительных свойств в формате ключ-значение, чтобы они не были видны в командной строке, или для перезаписи существующих свойств. Например: ”javax.net.ssl.trustStorePassword=<p
START_POLLERS	zabbix.startPollers	no	1-1000	5	Количество запускаемых рабочих потоков.
TIMEOUT	zabbix.timeout	no	1-30	3	Как долго ожидать завершения сетевых операций (в секундах).

Warning:
Порт 10052 не зарегистрирован в IANA.

9 Веб-служба Zabbix

Обзор

Веб-служба Zabbix — это процесс, используемый для взаимодействия с внешними веб-службами.

В этом разделе перечислены параметры, поддерживаемые файлом конфигурации веб-службы Zabbix (zabbix_web_service.conf).

Параметры перечислены без дополнительной информации. Нажмите на параметр, чтобы увидеть полное описание.

Parameter	Description
AllowedIP	Список IP-адресов, разделённых запятыми, при необходимости в нотации CIDR, или DNS-имён серверов Zabbix и прокси Zabbix.
DebugLevel	Уровень отладки.
IgnoreURLCertErrors	Определяет обработку ошибок проверки TLS-сертификата при обращении к URL веб-интерфейса.
Include	В файл конфигурации можно включать отдельные файлы или все файлы из каталога.
ListenPort	Служба будет прослушивать этот порт для соединений от сервера.
LogFile	Имя файла журнала.
LogFileSize	Максимальный размер файла журнала.
LogType	Тип вывода журнала.
Timeout	Максимальное время (в секундах), затрачиваемое на формирование PDF отчёта панели мониторинга.
TLSAccept	Какие входящие соединения принимать.
TLSCAFile	Полный путь к файлу, содержащему сертификаты корневых центров сертификации (CA) для проверки сертификата удалённой стороны, используемый для шифрованного взаимодействия между компонентами Zabbix.
TLSCertFile	Полный путь к файлу, содержащему сертификат службы или цепочку сертификатов, используемый для шифрованного взаимодействия между компонентами Zabbix.

Parameter	Description
TLSKeyFile	Полный путь к файлу, содержащему закрытый ключ службы, используемый для шифрованного взаимодействия 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 окружения;
- Zabbix поддерживает файлы конфигурации только в кодировке UTF-8 без BOM;
- Комментарии, начинающиеся с "#", поддерживаются только в начале строки.

Подробные сведения о параметрах

AllowedIP

Список IP-адресов, разделённых запятыми, при необходимости в нотации CIDR, или DNS-имён серверов Zabbix и прокси Zabbix. Входящие соединения будут приниматься только от узлов сети, перечисленных здесь.
Если поддержка IPv6 включена, тогда `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:

```
127.0.0.1,192.168.1.0/24,::1,2001:db8::/32,zabbix.example.com
```

Обязательный: да

DebugLevel

Укажите уровень отладки:
0 — основная 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 ещё 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

Вы можете включать отдельные файлы или все файлы в каталоге в файл конфигурации. Во время установки Zabbix создаст каталог include в /usr/local/etc, если этот путь не был изменён во время компиляции. Путь может быть относительным к расположению файла *zabbix_web_service.conf*.
Чтобы включать только соответствующие файлы в указанном каталоге, для сопоставления с шаблоном поддерживается символ подстановки звёздочка.
См. специальные примечания об ограничениях.

Пример:

```
Include=/absolute/path/to/config/files/*.conf
```

ListenPort

Служба будет прослушивать этот порт для соединений от сервера.

По умолчанию: 10053
 Диапазон: 1024-32767

LogFile

Имя файла журнала.

Пример:

```
/tmp/zabbix_web_service.log
```

Обязательно: Да, если для LogType установлено значение *file*; в противном случае нет

LogFileSize

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 удалась, существующий файл журнала будет усечён и начат 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`

Timeout

Максимальное время (в секундах), затрачиваемое на форматирование PDF отчёта панели.

По умолчанию: 10
 Диапазон: 1-30

TLSAccept

Какие входящие соединения принимать:
unencrypted — принимать соединения без шифрования (по умолчанию)
cert — принимать соединения с TLS и сертификатом

По умолчанию: `unencrypted`

TLSCAFile

Полный путь к файлу, содержащему сертификаты корневых центров сертификации (CA) для проверки сертификата удалённой стороны, используемый для шифрованного обмена данными между компонентами Zabbix.

TLSCertFile

Полный путь к файлу, содержащему сертификат службы или цепочку сертификатов, используемый для шифрованного обмена данными с компонентами Zabbix.

TLSKeyFile

Полный путь к файлу, содержащему закрытый ключ сервиса, используемый для шифрованного обмена данными между компонентами Zabbix.

10 Переменные окружения

Обзор

Переменные окружения позволяют настраивать компоненты Zabbix без жёсткого задания значений в конфигурационных файлах. Это упрощает управление конфигурациями в динамических средах, таких как Docker, где переменные могут передаваться во время выполнения для адаптации к различным вариантам настройки.

В простейшем случае, задав значение параметра конфигурации сервера Zabbix DebugLevel через переменную окружения, вы затем можете использовать её для настройки сервера при запуске:

```
####                    Zabbix:
DebugLevel=${NEW_DEBUG_LEVEL}

####           Zabbix:
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 параметра -c, --config)

Важные примечания

- Если параметр конфигурации задан через переменную окружения, которая не указана при запуске компонента, используется значение параметра по умолчанию.

- При использовании команд времени выполнения (например, для увеличения уровня журналирования агента) необходимо указывать все ранее использованные переменные окружения. Это связано с тем, что компоненты Zabbix используют свой файл конфигурации для выполнения команд времени выполнения; если переменные окружения не указаны, будут использованы значения параметров конфигурации по умолчанию. См. Примеры.
- Команда времени выполнения `userparameter_reload` не поддерживает перезагрузку переменных окружения. Во время перезагрузки переменные игнорируются, и перезагружаются только параметры с обычными значениями.
- Текущие переменные окружения процесса, которые использовались в файлах конфигурации, очищаются после запуска компонента Zabbix. Это гарантирует, что дочерние процессы (например, удалённые скрипты, выполняемые Zabbix) не смогут получить доступ к этим переменным. Однако обратите внимание, что исходные переменные процесса всё ещё могут быть получены (например, через файл `/proc/<PID>/environ`).

Синтаксис

Переменные окружения должны использовать следующий синтаксис: `${alphanumerics/underscores}`.

Имя переменной может включать 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.

Корректный синтаксис переменных:

```
DebugLevel=${NEW_DEBUG_LEVEL}
Hostname=${ZBX_HOSTNAME}
LogFile=${LogFile_001}
```

Некорректный синтаксис переменных:

```
DebugLevel=${5_DebugLevel}
Hostname=${ZBX.HOSTNAME 1}
LogFile=/${HOME}/zabbix/zabbix_server.log
```

> **Note:**
> В Windows имена переменных окружения нечувствительны к регистру.

Примеры

Следующие примеры показывают, как настраивать и использовать переменные окружения с компонентами Zabbix.

Пример 1: Настройка и тестирование агента Zabbix

1. Задайте переменные окружения в файле конфигурации агента:

```
Hostname=${ZBX_HOSTNAME}
ServerActive=${ServerActive}
```

2. Проверьте 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 окружения:

```
ZBX_HOSTNAME="New Zabbix agent" ServerActive=127.0.0.1 /usr/sbin/zabbix_agentd -c /etc/zabbix/zabbix_agent
```

При использовании команд времени выполнения (например, для увеличения уровня логирования агента) необходимо указывать все ранее использованные переменные окружения:

```
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 времени выполнения; если переменные окружения не указаны, будут использованы значения параметров конфигурации по умолчанию.

В качестве альтернативы, после задания переменных окружения в файле конфигурации агента, вы можете сделать их доступными для процессов (например, с помощью команды `export`). Это снижает риск неожиданного поведения из-за отсутствующих или неправильно заданных 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 создаёте и настраиваете собственный пользовательский образ для компонентов Zabbix (например, агента Zabbix), вы можете задать параметры конфигурации с помощью переменных окружения, а затем запустить контейнер с этими переменными.

1. При подготовке образа задайте переменные окружения 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 окружения:

```
docker run --name my-zabbix-agent -e ZBX_HOSTNAME="new-hostname" -e BUFSZ=1000 -e LISTENPORT=20050 -e _Usr
```

3. При использовании команд времени выполнения (например, для увеличения уровня журналирования агента) откройте оболочку контейнера и выполните команду времени выполнения:

```
docker exec -it <containerid> sh
/usr/sbin/zabbix_agentd -R log_level_increase
```

Attention:
Команда времени выполнения `userparameter_reload` не поддерживает перезагрузку переменных окружения. Во время перезагрузки переменные игнорируются, и повторно загружаются только параметры с обычными значениями.

11 Включение

Обзор

Дополнительные файлы или каталоги могут быть включены в конфигурацию сервера, прокси, агента и веб-службы с помощью параметра `Include`.

Примечания по включению

Если параметр `Include` используется для включения файла, файл должен быть доступен для чтения.

Если параметр `Include` используется для включения каталога:

- Все файлы в каталоге должны быть доступны для чтения.
- Не следует предполагать какой-либо определенный порядок включения (например, файлы не включаются в алфавитном порядке). Поэтому не определяйте один параметр в нескольких файлах "Include" (например, чтобы переопределить общее значение более специфичным).
- Все файлы в каталоге включаются в конфигурацию.
- Обратите внимание на резервные копии файлов, автоматически создаваемые некоторыми текстовыми редакторами. Например, если при редактировании файла "include/my_specific.conf" создается резервная копия "include/my_specific_conf.BAK", то будут включены оба файла. Переместите "include/my_specific.conf.BAK" из каталога "Include". В Linux содержимое каталога "Include" можно проверить командой "ls -al" на наличие ненужных файлов.

Если параметр `Include` используется для включения файлов с использованием шаблона:

- Все файлы, соответствующие шаблону, должны быть доступны для чтения.
- Не следует предполагать какой-либо определенный порядок включения (например, файлы не включаются в алфавитном порядке). Поэтому не определяйте один параметр в нескольких файлах "Include" (например, чтобы переопределить общее значение более специфичным).

3 Протоколы

Пожалуйста, используйте боковую панель для доступа к содержимому этого раздела.

1 Протокол обмена данными между сервером и прокси

Обзор

Обмен данными сервер - прокси основан на формате JSON.

Сообщения запроса и ответа должны начинаться с заголовка и длины данных.

Пассивный прокси

Запрос конфигурации

Сервер сначала отправляет пустой запрос `proxy config`. Этот запрос отправляется каждые `ProxyConfigFrequency` секунд (параметр конфигурации сервера).

Прокси отвечает текущей версией прокси, токеном сессии и ревизией 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 сессии конфигурации прокси.
config_revision	*number*	Ревизия конфигурации прокси.
server→proxy:		
full_sync	*number*	1 — если отправлены полные данные конфигурации; отсутствует — в противном случае (необязательно).
data	*array*	Объект с данными таблиц. Отсутствует, если конфигурация не изменялась (необязательно).
<table>	*object*	Один или несколько объектов с данными <table> (необязательно, в зависимости от изменений).
fields	*array*	Массив имен полей.
-	*string*	Имя поля.
data	*array*	Массив строк.
-	*array*	Массив столбцов.
-	*string*,*number*	Значение столбца, тип зависит от типа столбца в схеме базы данных.
macro.secrets	*object*	Информация о секретных макросах; отсутствует, если в макросах хранилища секретов нет изменений (необязательно).
config_revision	*number*	Ревизия кэша конфигурации — отправляется вместе с данными конфигурации (необязательно).
del_hostids	*array*	Массив удалённых hostid (необязательно).
-	*number*	Идентификатор узла сети.
del_macro_hostids	*array*	Массив hostid, для которых удалены все макросы (необязательно).
-	*number*	Идентификатор узла сети.
proxy→server:		
response	*string*	Информация об успешности запроса ('success' или 'failed').
version	*string*	Версия прокси (<major>.<minor>.<build>).

Пример:

server→proxy:

```
{
  "request":"proxy config"
}
```

proxy→server:

```
{
  "version": "8.0.0",
  "session": "0033124949800811e5686dbfd9bcea98",
  "config_revision": 0
}
```


server→proxy:


```
{
"full_sync": 1,
"data": {
"hosts": {
"fields": ["hostid", "host", "status", "ipmi_authtype", "ipmi_privilege", "ipmi_username", "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, "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "5
]
},
"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
}
```


proxy→server:


```
{
  "response": "success",
  "version": "8.0.0"
}
```


Запрос данных

Запрос `proxy data` используется для получения от прокси данных о доступности интерфейсов узла сети, исторических данных, данных обнаружения и авторегистрации. Этот запрос отправляется каждые `ProxyDataFrequency` секунд (параметр конфигурации сервера).

name	value type	description
server→proxy:		
request	*string*	'proxy data'
proxy→server:		
session	*string*	Токен сеанса данных.
interface avail-abil-ity	*array*	*(optional)* Массив объектов данных о доступности интерфейсов.
interfaceid	*number*	Идентификатор интерфейса.

name	value type	description
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).
value	*string*	*(optional)* Значение элемента данных.
id	*number*	Идентификатор значения (возрастающий счётчик, уникальный в пределах одного сеанса данных).
timestamp	*number*	*(optional)* Метка времени элементов данных типа log.
source	*string*	*(optional)* Значение source элемента данных eventlog.
severity	*number*	*(optional)* Значение severity 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 сервиса SSH **1**, *SVC_LDAP* - проверка сервиса LDAP **2**, *SVC_SMTP* - проверка сервиса SMTP **3**, *SVC_FTP* - проверка сервиса FTP **4**, *SVC_HTTP* - проверка сервиса HTTP **5**, *SVC_POP* - проверка сервиса POP **6**, *SVC_NNTP* - проверка сервиса NNTP **7**, *SVC_IMAP* - проверка сервиса 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 сервиса HTTPS **15**, *SVC_TELNET* - проверка доступности Telnet
ip	*string*	IP-адрес узла сети.
dns	*string*	DNS-имя узла сети.
port	*number*	*(optional)* Номер порта сервиса.
key_	*string*	*(optional)* Ключ элемента данных для проверки обнаружения типа **9** *SVC_AGENT*
value	*string*	*(optional)* Значение, полученное от сервиса; для большинства сервисов может быть пустым.
status	*number*	*(optional)* Состояние сервиса: **0**, *DOBJECT_STATUS_UP* - Сервис UP **1**, *DOBJECT_STATUS_DOWN* - Сервис DOWN

name	value type	description
auto reg- is- tra- tion	*array*	*(optional)* Массив объектов данных авторегистрации.
clock	*number*	Метка времени данных авторегистрации.
host	*string*	Имя узла сети.
ip	*string*	*(optional)* IP-адрес узла сети.
dns	*string*	*(optional)* DNS-имя, разрешённое из IP-адреса.
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 удалённой команды
status	*number*	Состояние выполнения удалённой команды: **0**, *ZBX_TM_REMOTE_COMMAND_COMPLETED* - удалённая команда успешно выполнена **1**, *ZBX_TM_REMOTE_COMMAND_FAILED* - ошибка выполнения удалённой команды
error	*string*	*(optional)* Сообщение об ошибке.
parent_taskid	*number*	ID родительской задачи.
more	*number*	*(optional)* 1 - есть дополнительные исторические данные 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’).
tasks	*array*	*(optional)* Массив задач.
type	*number*	Тип задачи: **1**, *ZBX_TM_TASK_PROCESS_REMOTE_COMMAND* - удалённая команда
clock	*number*	Время создания задачи.
ttl	*number*	Время в секундах, по истечении которого задача становится недействительной.
commandtype	*number*	Тип удалённой 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*	Удалённая 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.

name	value type	description
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 доставку и подтверждение задач удалённых 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*	*(необязательно)* Временная метка передачи данных (секунды).
ns	*number*	*(необязательно)* Временная метка передачи данных (наносекунды).
server→proxy:		
response	*string*	Информация об успешности запроса ('success' или '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` (параметр конфигурации прокси) секунд.

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 (необязательно).
data	*array*	Объект табличных данных. Отсутствует, если конфигурация не изменялась (необязательно).
<table>	*object*	Один или несколько объектов с данными <table> (необязательно, в зависимости от изменений).
fields	*array*	Массив имен полей.
-	*string*	Имя поля.
data	*array*	Массив строк.
-	*array*	Массив столбцов.
-	*string,number*	Значение столбца, тип которого зависит от типа столбца в схеме базы данных.
macro.secrets	*object*	Информация о секретных макросах; отсутствует, если в макросах хранилища секретов нет изменений (необязательно).
proxy_group	*string*	Имя группы прокси, к которой принадлежит прокси.
config_revision	*number*	Ревизия кэша конфигурации — отправляется вместе с данными конфигурации (необязательно).
del_hostids	*array*	Массив удаленных hostid (необязательно).
-	*number*	Идентификатор узла сети.
del_macro_hostids	*array*	Массив hostid, у которых удалены все макросы (необязательно).
-	*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, "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "5
]
},
"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

Запрос `proxy data` отправляется прокси для передачи данных о доступности интерфейсов узла сети, истории, обнаружении и авторегистрации. Этот запрос отправляется каждые `DataSenderFrequency` секунд (параметр конфигурации прокси). Обратите внимание, что активный прокси по-прежнему будет опрашивать сервер Zabbix каждую секунду на предмет задач удалённых 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 счётчик, уникальный в пределах одного сеанса данных).
timestamp	*number*	*(optional)* Метка времени элементов данных типа журнал.
source	*string*	*(optional)* Значение source элемента данных eventlog.
severity	*number*	*(optional)* Значение severity 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 журнал.
mtime	*number*	*(optional)* Время изменения для элементов данных типа журнал.
discovery data	*array*	*(optional)* Массив объектов данных обнаружения.

name	value type	description
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 сервиса SSH **1**, *SVC_LDAP* - проверка сервиса LDAP **2**, *SVC_SMTP* - проверка сервиса SMTP **3**, *SVC_FTP* - проверка сервиса FTP **4**, *SVC_HTTP* - проверка сервиса HTTP **5**, *SVC_POP* - проверка сервиса POP **6**, *SVC_NNTP* - проверка сервиса NNTP **7**, *SVC_IMAP* - проверка сервиса 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 сервиса HTTPS **15**, *SVC_TELNET* - проверка доступности Telnet
ip	*string*	IP-адрес узла сети.
dns	*string*	DNS-имя узла сети.
port	*number*	*(optional)* Номер порта сервиса.
key_	*string*	*(optional)* Ключ элемента данных для проверки обнаружения типа **9** *SVC_AGENT*
value	*string*	*(optional)* Значение, полученное от сервиса; для большинства сервисов может быть пустым.
status	*number*	*(optional)* Состояние сервиса: **0**, *DOBJECT_STATUS_UP* - сервис UP **1**, *DOBJECT_STATUS_DOWN* - сервис DOWN
autoregistration	*array*	*(optional)* Массив объектов данных авторегистрации.
clock	*number*	Метка времени данных авторегистрации.
host	*string*	Имя узла сети.
ip	*string*	*(optional)* IP-адрес узла сети.
dns	*string*	*(optional)* DNS-имя, разрешённое из IP-адреса.
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 удалённой команды
status	*number*	Состояние выполнения удалённой команды: **0**, *ZBX_TM_REMOTE_COMMAND_COMPLETED* - удалённая команда успешно выполнена **1**, *ZBX_TM_REMOTE_COMMAND_FAILED* - ошибка выполнения удалённой команды
error	*string*	*(optional)* Сообщение об ошибке.
parent_taskid	*number*	ID родительской задачи.
more	*number*	*(optional)* 1 - есть дополнительные 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 (история, авторегистрация, доступность узла сети, сетевое обнаружение). Возможные значения: **enabled** - нормальная работа **disabled** - сервер не принимает данные (возможно, из-за превышения лимита внутреннего кэша)
tasks	*array*	*(optional)* Массив задач.
type	*number*	Тип задачи: **1**, *ZBX_TM_TASK_PROCESS_REMOTE_COMMAND* - удалённая команда
clock	*number*	Время создания задачи.
ttl	*number*	Время в секундах, по истечении которого задача становится недействительной.
commandtype	*number*	Тип удалённой 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*	Удалённая 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
}]
}
```

2 Протокол Zabbix агент/агент2

Пожалуйста, обратитесь к странице пассивные и активные проверки агента для получения более подробной информации о протоколах агента Zabbix и агента Zabbix 2.

4 Zabbix agent 2 plugin protocol

Zabbix agent 2 protocol is based on code, size and data model.

Код

Type	Size	Comments
Байт	4	Тип полезной нагрузки, в настоящее время поддерживается только JSON.

Размер

Тип	Размер	Комментарии
Байт	4	Размер текущей полезной нагрузки в байтах.

Данные полезной нагрузки

Тип	Размер	Комментарии
Байт	Определяется полем *Размер*	Данные в формате JSON.

Определение данных полезной нагрузки

Общие данные

Эти параметры присутствуют во всех запросах/ответах:

Name	Type	Comments
id	uint32	Для запросов — увеличивающийся идентификатор, используемый для связывания запросов с ответами. Уникален в пределах одного направления запросов (то есть от агент к плагину или от плагина к агент). Для ответов — ID соответствующего запроса.
type	uint32	Тип запроса.

Запрос журнала

Запрос, отправляемый плагином для записи сообщения журнала в файл журнала агента.

direction	plugin → agent
response	no

Параметры, специфичные для запросов журнала:

Name	Type	Comments
severity	uint32	Уровень важности сообщения (уровень журнала).
message	string	Сообщение для записи в журнал.

Пример:

```
{"id":0,"type":1,"severity":3,"message":"message"}
```

Запрос регистрации

Запрос, отправляемый агентом на этапе запуска агента для получения предоставляемых метрик с целью регистрации плагина.

direction	агент → plugin
response	да

Параметры, специфичные для запросов регистрации:

Name	Type	Comments
version	string	Версия протокола <major>.<minor>

Пример:

```
{"id":1,"type":2,"version":"1.0"}
```

Ответ на регистрацию

Ответ плагина на запрос регистрации.

direction	плагин → агент
response	n/a

Параметры, специфичные для ответов на регистрацию:

Name	Type	Comments
name	string	Имя плагина.
metrics	array of strings (optional)	Метрики с описаниями, используемые в плагине. Возвращает RegisterMetrics(). Отсутствует, если возвращена ошибка.
interfaces	uint32 (optional)	Битовая маска поддерживаемых плагином интерфейсов. Отсутствует, если возвращена ошибка.
error	string (optional)	Сообщение об ошибке, возвращаемое, если плагин не может быть запущен. Отсутствует, если возвращены 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 имеет специальных параметров, он содержит только параметры общих данных.

Пример:

```
{"id":3,"type":4}
```

Завершение запроса

Запрос, отправляемый агентом для завершения работы плагина.

direction	агент → plugin
response	no

У запроса нет 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 plugin.
parameters	array of strings (optional)	Параметры для функции Export.

Пример:

```
{"id":4,"type":6,"key":"test.key","parameters":["foo","bar"]}
```

Ответ экспорта

Ответ функции Export интерфейса Exporter.

direction	plugin → агент
response	n/a

Параметры, специфичные для ответов экспорта:

Name	Type	Comments
value	string (optional)	Значение ответа 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	Объект JSON, содержащий глобальные параметры конфигурации агента.
private_options	JSON object (optional)	Объект JSON, содержащий частные параметры конфигурации плагина, 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	да

Параметры, специфичные для запросов *Validate*:

Name	Type	Comments
private_options	JSON object (optional)	Объект JSON, содержащий параметры приватной конфигурации плагина, если они указаны.

Пример:

```
{"id":7,"type":9,"private_options":{...}}
```

Проверка ответа

Ответ от функции *Validate* интерфейса *Configurator*.

направление	плагин → агент
ответ	н/д

Параметры, специфичные для ответов *Validate*:

Имя	Тип	Комментарии
error	string (optional)	Сообщение об ошибке, возвращаемое, если функция Validate не была выполнена успешно. Отсутствует при успешном выполнении.

Пример:

```
{"id":8,"type":10}
```

или

```
{"id":8,"type":10,"error":"error message"}
```

5 Протокол Zabbix sender

Обзор

Сервер Zabbix и прокси-сервер Zabbix используют протокол связи на основе JSON для получения данных от отправителя Zabbix. Данные можно получить с помощью траппер элемента данных или элемента данных HTTP-агента с включенным перехватом.

Сообщения запроса и ответа должны начинаться с длины заголовка и данных.

Запрос Zabbix sender

```
{
   "request": "sender data",
   "data": [
   {
   "host": "<        >",
   "key": "trap",
   "value": "          "
    }
    ]
}
```

Ответ сервера Zabbix

```
{
    "response": "success",
    "info": "      : 1;       : 0;    : 1;          : 0.060753"
}
```

Запрос Zabbix sender с временной меткой

В качестве альтернативы 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"
}
```

6 Заголовок

Обзор

Заголовок присутствует во всех сообщениях запросов и ответов между компонентами Zabbix. Он необходим для определения длины сообщения, сжато оно или нет, является ли пакет большим или нет.

Протокол обмена данными Zabbix имеет ограничение размера пакета в 1 ГБ на одно соединение. Ограничение в 1 ГБ применяется как к длине данных полученного пакета, так и к длине данных в несжатом виде.

При отправке конфигурации на Zabbix прокси ограничение размера пакета увеличивается до 4 ГБ, чтобы обеспечить синхронизацию больших конфигураций. Когда длина данных до сжатия превышает 4 ГБ, Zabbix сервер автоматически начинает использовать формат большого пакета (флаг 0x04), который увеличивает ограничение размера пакета до 16 ГБ.

Обратите внимание, что хотя формат большого пакета может использоваться для отправки любых данных, в настоящее время только синхронизатор конфигурации Zabbix прокси может обрабатывать пакеты размером более 1 ГБ.

Структура

Заголовок состоит из четырёх полей. Все числа в заголовке отформатированы в формате little-endian.

Поле	Размер	Размер (большой пакет)	Описание
<PROTOCOL>	4	4	"ZBXD" или 5A 42 58 44
<FLAGS>	1	1	Флаги протокола: 0x01 — протокол обмена данными Zabbix 0x02 — сжатие 0x04 — большой пакет
<DATALEN>	4	8	Длина данных.
<RESERVED>	4	8	При использовании сжатия (флаг 0x02) — длина несжатых данных Если сжатие не используется — 00 00 00 00

Примеры

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. Эти фрагменты кода предполагают, 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"
```

7 Протокол экспорта в формате JSON с разделением по строкам

В этом разделе представлены сведения о протоколе экспорта в формате JSON с разделением по строкам, который используется в:

- экспорте данных в файлы
- потоковой передаче во внешние системы

Можно экспортировать следующее:

- события триггеров

- значения элементов данных
- тренды (только экспорт в файлы)

Все файлы имеют расширение .ndjson. Каждая строка файла экспорта представляет собой объект JSON.

События триггеров

Для события проблемы экспортируется следующая информация:

Field		Type	Description
clock		number	Количество секунд с начала эпохи до момента обнаружения проблемы (целая часть).
ns		number	Количество наносекунд, которое необходимо 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	Важность события проблемы (0 - Не классифицировано, 1 - Информация, 2 - Предупреждение, 3 - Средняя, 4 - Высокая, 5 - Катастрофа).
hosts		array	Список узлов сети, задействованных в выражении триггера; массив должен содержать 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, задействованных в выражении триггера; массив должен содержать 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 до момента устранения проблемы (целая часть).
ns	number	Количество наносекунд, которое необходимо 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

Проблема:

```
{"clock":1519304285,"ns":123456789,"value":1,"name":"       Zabbix               Host B,
```

Восстановление:

```
{"clock":1519304345,"ns":987654321,"value":0,"eventid":43,"p_eventid":42}
```

Проблема (генерация нескольких событий проблем):

```
{"clock":1519304286,"ns":123456789,"value":1,"eventid":43,"name":"       Zabbix               Host B,
```

```
{"clock":1519304286,"ns":123456789,"value":1,"eventid":43,"name":"       Zabbix               Host B,
```

Восстановление:

{"clock":1519304346,"ns":987654321,"value":0,"eventid":44,"p_eventid":43}

{"clock":1519304346,"ns":987654321,"value":0,"eventid":44,"p_eventid":42}

Значения элементов данных

Следующая информация экспортируется для собранного значения элемента данных:

Поле	Тип	Описание
host	object	Имя узла сети элемента данных.
host	string	Имя узла сети.
name	string	Видимое имя узла сети.
groups	array	Список групп узлов сети узла 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	Видимое имя элемента данных.
clock	number	Количество секунд с начала эпохи до момента, когда значение было собрано (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 — числовое с плавающей точкой, 1 — символьное, 2 — журнал, 3 — числовое без знака, 4 — текст, 5 — двоичное, 6 — JSON

Примеры

Числовое (беззнаковое) значение:

{"host":{"host":"Host B","name":"Host B visible"},"groups":["Group X","Group Y","Group Z"],"item_tags":[{"

Числовое значение (с плавающей точкой):

{"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":[{"

Тренды

Для вычисленного значения тренда экспортируется следующая информация:

Поле	Тип	Описание
host	object	Имя узла сети элемента данных.
host	string	Имя узла сети.
name	string	Видимое имя узла сети.
groups	array	Список групп узла сети элемента данных; в массиве должен быть как минимум один элемент.
-	string	Имя группы узла 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	Видимое имя элемента данных.
clock	number	Количество секунд с начала эпохи до момента, когда значение было собрано (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.
max	number	Максимальное значение элемента данных за данный час.
type	number	Тип значения: 0 - числовой с плавающей точкой, 3 - числовой без знака

Примеры

Числовое (беззнаковое) значение:

```
{"host":{"host":"Host B","name":"Host B visible"},"groups":["Group X","Group Y","Group Z"],"item_tags":[{"
```

Числовое значение с плавающей точкой:

```
{"host":{"host":"Host B","name":"Host B visible"},"groups":["Group X","Group Y","Group Z"],"item_tags":[{"
```

4 Элементы данных

Пожалуйста, используйте боковую панель для доступа к содержимому этого раздела.

1 Параметры vm.memory.size

Обзор

В этом разделе приведены некоторые сведения о параметрах элемента данных агента vm.memory.size[<mode>].

Параметры

Для этого элемента данных доступны следующие параметры:

- **active** — память, используемая в данный момент или использовавшаяся совсем недавно, поэтому она находится в RAM
- **anon** — память, не связанная с файлом (не может быть повторно считана из него)
- **available** — доступная память, вычисляется по-разному в зависимости от платформы (см. таблицу ниже)
- **buffers** — кэш для таких вещей, как метаданные файловой системы
- **cached** — кэш для различных данных
- **exec** — исполняемый код, обычно из файла (программы)
- **file** — кэш содержимого недавно использовавшихся файлов
- **free** — память, которая немедленно доступна любому объекту, запрашивающему память
- **inactive** — память, помеченная как неиспользуемая
- **pavailable** — память 'available' в процентах от 'total' (вычисляется как `available/total*100`)
- **pinned** — то же, что и 'wired'
- **pused** — память 'used' в процентах от 'total' (вычисляется как `used/total*100`)
- **shared** — память, к которой могут одновременно обращаться несколько процессов
- **slab** — общий объём памяти, используемой ядром для кэширования структур данных для собственных нужд
- **total** — общий объём доступной физической памяти
- **used** — используемая память, вычисляется по-разному в зависимости от платформы (см. таблицу ниже)

- **wired** — память, помеченная как всегда остающаяся в RAM. Она никогда не перемещается на диск.

Warning:
Некоторые из этих параметров зависят от платформы и могут быть недоступны на вашей платформе. Подробности см. в элементах данных агента Zabbix.

Специфичный для платформы расчёт **available** и **used**:

Platform	**"available"**	**"used"**
AIX	free + cached	используемая реальная память
FreeBSD	inactive + cached + free	active + wired + cached
HP UX	free	total - free
Linux<3.14	free + buffers + cached	total - free
Linux 3.14+ (также портировано в 3.10 на RHEL 7)	/proc/meminfo, подробности см. в "MemAvailable" в documentation ядра Linux. Обратите внимание, что free + buffers + cached больше не равно 'available', поскольку не весь кэш страниц может быть освобождён, и в расчёте используется нижняя граница 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 total. Например, в FreeBSD:
* Память active, inactive, wired, cached считается используемой, потому что в ней хранится некоторая полезная информация.
* В то же время память inactive, cached, free считается доступной, потому что эти виды памяти могут быть немедленно предоставлены процессам, запрашивающим больше памяти.

Таким образом, память inactive одновременно является и используемой, и доступной. Поэтому элемент данных *vm.memory.size[used]* предназначен только для информационных целей, тогда как *vm.memory.size[available]* предназначен для использования в триггерах.

См. также

1. Дополнительные сведения о расчете памяти в различных ОС

2. Пассивные и активные проверки агента

Обзор

Этот раздел подробно описывает пассивные и активные проверки, которые выполняются Zabbix агентом.

Для взаимодействия с агентами Zabbix использует коммуникационный протокол на основе JSON.

Протоколы Zabbix агента и Zabbix агента 2 были унифицированы, начиная с версии Zabbix 7.0. Отличие между запросами/ответами Zabbix агента и Zabbix агента 2 выражается значением тега «variant».

Пассивные проверки

Пассивная проверка — это простой запрос данных. Zabbix сервер или прокси запрашивает некоторые данные (например, загрузку CPU), а Zabbix агент отправляет результат обратно на сервер.

Пассивные проверки выполняются асинхронно — не требуется получать ответ на один запрос до запуска других проверок. Разрешение DNS также выполняется асинхронно.

Опросчик агента попытается подключиться ко всем адресам, возвращённым при DNS-поиске. Это гарантирует, что если один IP-адрес недоступен, опросчик попробует следующий доступный адрес, повышая вероятность успешного подключения. Это улучшение применяется как к Zabbix серверу, так и к прокси.

Максимальная параллельность асинхронных проверок составляет 1000 (определяется параметром MaxConcurrentChecksPer-Poller).

Количество асинхронных опросчиков агента определяется параметром StartAgentPollers.

Запрос сервера

Описание заголовка и длины данных см. в разделе подробности протокола.

```
{
  "request": "passive checks",
  "data": [
    {
      "key": "agent.version",
      "timeout": 3
    }
  ]
}
```

Поле		Тип	Обязательное	Значение
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	Тайм-аут соединения.

Ответ агента

```
{
  "version": "8.0.0",
  "variant": 2,
  "data": [
    {
      "value": "8.0.0"
    }
  ]
}
```

Поле		Тип	Обязательное	Значение
version		*string*	yes	Номер версии агента.
variant		*number*	yes	Вариант агента (*1* - Zabbix агент, *2* - Zabbix агент 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 завершилась 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 «не поддерживается» с указанным сообщением об ошибке
5. TCP-соединение закрывается

Переключение на старый протокол

Чтобы сервер Zabbix или прокси могли работать с агентами версий до 7.2, в которых используется открытый текстовый протокол, реализовано переключение на старый протокол.

Пассивные проверки выполняются с использованием протокола JSON (7.0 и новее) после перезапуска или при изменении конфигурации интерфейса. Если в ответ не получен корректный JSON (агент отправил "ZBX_NOTSUPPORTED"), Zabbix сохранит интерфейс в кэше как использующий старый протокол и **повторит** проверку, отправив только ключ элемента данных.

Обратите внимание, что каждый час сервер Zabbix/прокси будет снова пытаться работать по новому протоколу со всеми интерфейсами, при необходимости возвращаясь к старому протоколу.

Активные проверки

Активные проверки требуют более сложной обработки. Агент сначала должен получить с сервера/прокси список элементов данных и/или удалённых команд для независимой обработки.

Сервера/прокси для получения активных проверок перечислены в параметре «ServerActive» файла конфигурации агента. Частота запросов данных проверок настраивается параметром «RefreshActiveChecks» в этом же файле конфигурации. Однако, если обновление активных проверок завершится с ошибкой, запрос повторится через (жёстко заданные в коде) 60 секунд.

Note:
Начиная с версии 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 уменьшить сетевой трафик и использование ресурсов, каждые 5 секунд (по умолчанию) выполняется инкрементная синхронизация конфигурации, после чего сервер/прокси-сервер предоставляет полную копию конфигурации **только**, если агент ещё не получил её либо что-то изменилось в конфигурации узла сети, глобальных макросах или глобальных регулярных выражениях.

Затем агент периодически отправляет новые значения на сервер(ы). Если агент получил какие-либо удалённые команды для выполнения, также будет отправлен результат выполнения. Обратите внимание, что удалённое выполнение команд на активном агенте поддерживается, начиная с версии Zabbix агента 7.0.

Note:
Если агент находится за брандмауэром, вы можете рассмотреть возможность использования исключительно Активных проверок, т.к. в этом случае вам не понадобится менять настройки брандмауэра для разрешения начальных входящих соединений.

Получение списка элементов данных

Запрос агента

Запрос активных проверок 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 параметра конфигурации HostMetadata или метрики HostMetadataItem.
interface	*string*	no	Значение параметра конфигурации HostInterface или метрики HostInterfaceItem.
ip	*string*	no	Первый IP из параметра конфигурации ListenIP, если задан.
port	*number*	no	Значение параметра конфигурации ListenPort, если задано и не равно порту прослушивания агента по умолчанию.
version	*string*	yes	Номер версии агента.

Field	Type	Mandatory	Value
variant	*number*	yes	Вариант агента (*1* - Zabbix агент, *2* - Zabbix агент 2).
config_revision	*number*	no	Идентификатор конфигурации для инкрементальной синхронизации конфигурации.
session	*string*	no	Идентификатор сессии для инкрементальной синхронизации конфигурации.

Ответ сервера

Ответ активных проверок отправляется сервером обратно агенту после обработки запроса активных проверок.

```
{
  "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. Опускается, 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 агентом, так и Zabbix агентом 2, начиная с Zabbix 7.0.
lastlogsize	*number*	no	Значение lastlogsize элемента данных.
mtime	*number*	no	Значение 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.

Field	Type	Mandatory	Value
case_sensitive	*number*	no	Настройка чувствительности к регистру для глобального регулярного выражения.
commands	*array of objects*	no	Удалённые команды для выполнения. Включаются, если выполнение удалённой команды было инициировано операцией действия или ручным выполнением скрипта. Обратите внимание, что выполнение удалённых команд на активном агенте поддерживается начиная с Zabbix агент 7.0. Более старые активные агенты будут игнорировать любые удалённые команды, включённые в ответ сервера активных проверок.
command	*string*	no	Удалённая команда.
id	*number*	no	Идентификатор удалённой команды.
wait	*number*	no	Режим выполнения удалённой команды (”0” (nowait) для команд из операций действия; ”1” (wait) для команд из ручного выполнения скрипта).
config_revision	*number*	no	Идентификатор конфигурации для инкрементальной синхронизации конфигурации. Опускается, если конфигурация узла сети не изменилась. Увеличивается, если конфигурация узла сети изменена.

Сервер должен ответить сообщением об успешном выполнении.

Например:

1. Агент открывает TCP-соединение
2. Агент запрашивает список проверок
3. Сервер отвечает списком элементов данных и удалённых команд для выполнения
4. Агент разбирает ответ
5. TCP-соединение закрывается
6. Агент начинает периодический сбор данных и выполняет удалённые команды (поддерживается начиная с Zabbix агент 7.0)

Attention:
Обратите внимание, что при использовании активной проверки (чувствительные) данные конфигурации могут стать доступны сторонам, имеющим доступ к порту trapper сервера Zabbix. Это возможно, потому что любой может выдать себя за активный агент и запросить данные конфигурации элементов данных; аутентификация не выполняется, если не используются параметры шифрования.

Отправка собранных данных

Агент отправляет

Запрос данных агента содержит собранные значения элементов данных и значения для выполненных удалённых команд (если 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
}
```

Field	Type	Mandatory	Value
request	*string*	yes	`agent data`
data	*array of objects*	yes	Значения элементов данных.
id	*number*	yes	Идентификатор значения (инкрементный счётчик, 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 журнала событий значения.
eventid	*number*	no	eventid журнала событий значения.
severity	*number*	no	severity журнала событий значения.
timestamp	*number*	no	timestamp 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 удалённых команд. Обратите внимание, что выполнение удалённых команд на активном агенте поддерживается начиная с Zabbix agent 7.0. Более старые активные агенты будут игнорировать любые удалённые команды, включённые в ответ сервера на активные проверки.
id	*number*	no	Идентификатор удалённой команды.
value	*string*	no	Результат выполнения удалённой команды, если выполнение прошло успешно.
error	*string*	no	Сообщение об ошибке выполнения удалённой команды, если выполнение завершилось неудачей.
session	*string*	yes	Уникальный идентификатор сессии, генерируемый при каждом запуске агента.
host	*string*	yes	Имя узла сети.
version	*string*	yes	Номер версии агента.
variant	*number*	yes	Вариант агента (*1* - Zabbix agent, *2* - Zabbix agent 2).

Каждому значению назначается виртуальный ID. ID значения — это простой возрастающий счётчик, уникальный в пределах одной сессии данных (идентифицируемой токеном сессии). Этот ID используется для отбрасывания дублирующихся значений, которые могут быть отправлены в условиях нестабильного соединения.

Ответ сервера

Ответ с данными агента отправляется сервером обратно агенту после обработки запроса данных агента.

```
{
  "response": "success",
  "info": "processed: 2; failed: 0; total: 2; seconds spent: 0.003534"
}
```

Field	Type	Mandatory	Value
response	*string*	yes	`success` | `failed`
info	*string*	yes	Результаты обработки элементов данных.

Attention:
Если отправка некоторых значений завершается ошибкой на стороне сервера (например, если узел сети или элемент данных были отключены или удалены), агент не будет повторно пытаться отправить эти значения.

Например:

1. Агент открывает TCP-соединение
2. Агент отправляет список значений
3. Сервер обрабатывает данные и отправляет обратно статус
4. TCP-соединение закрывается

Attention:
Сообщение об ошибке будет обрезано до 2048 символов на стороне сервера.

Сообщение heartbeat

Агент отправляет

Сообщение heartbeat отправляется активным агентом на сервер/прокси Zabbix каждые HeartbeatFrequency секунд (настраивается в файле конфигурации Zabbix agent/ agent 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	Обязательное	Значение
request	*string*	yes	`active check heartbeat`
host	*string*	yes	Имя узла сети.
heartbeat_freq	*number*	yes	Частота heartbeat агента (параметр конфигурации HeartbeatFrequency).
version	*string*	yes	Номер версии агента.
variant	*number*	yes	Вариант агента (*1* — Zabbix agent, *2* — Zabbix agent 2).

Ответ с перенаправлением

Когда узел сети был переназначен, сервер может указать агенту перенаправить его heartbeat (и последующие активные проверки) на другой экземпляр прокси или сервера.

```
  {
    "response": "failed",
    "redirect": {
      "revision": 2,
      "address": "192.0.2.0:10055"
    }
  }
```

Поле		Тип	Обязательное	Значение
response		*string*	yes	`success` | `failed`
redirect		*object*	yes	Инструкции по перенаправлению.
	revision	*number*	yes	Идентификатор редакции конфигурации.
	address	*string*	yes	Адрес целевого сервера/прокси.

Более старый XML протокол

Note:
Zabbix воспринимает до 16 MB XML данных, закодированных в Base64, но одиночное декодированное значение не должно быть длиннее 64 КБ, в противном случае значение будет обрезано до 64 КБ в процессе декодирования.

3 Минимальный уровень разрешений для элементов данных Windows агента

Обзор

При мониторинге систем с использованием агента хорошей практикой является получение метрик с того узла сети, на котором установлен агент. Чтобы использовать принцип наименьших привилегий, необходимо определить, какие метрики получаются от агента.

Таблица в этом документе позволяет выбрать минимальные права для гарантированно корректной работы Zabbix агент.

Если для работы агента выбран другой пользователь, а не 'LocalSystem', то для работы агента как службы Windows новый пользователь должен иметь право "Log on as a service" из "Local Policy→User Rights Assignment", а также право на создание, запись и удаление файла журнала Zabbix агент. Пользователь Active Directory должен быть добавлен в группу *Performance Monitor Users*.

Note:
При работе с правами агента на основе группы "минимально технически допустимых" требуется предварительное предоставление прав на объекты мониторинга.

Распространенные элементы данных агента, поддерживаемые в Windows

Ключ элемента данных	Группа пользователей	
	Рекомендуется	Минимально технически допустимо (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

Ключ элемента данных	Группа пользователей	
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 Moni-tor 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

Специфичные для Windows ключи элементов данных

Ключ элемента данных	Группа пользователей	
	Рекомендуется	Минимально технически допустимо (функциональность ограничена)
eventlog	Читатели журнала событий	Гости
net.if.list	Гости	Гости
perf_counter	Пользователи системного монитора	Пользователи системного монитора
proc_info	Администраторы	Гости
service.discovery	Гости	Гости
service.info	Гости	Гости
services	Гости	Гости
wmi.get	Администраторы	Гости
vm.vmemory.size	Гости	Гости

4 Кодировка возвращаемых значений

Сервер Zabbix ожидает, что каждое возвращаемое текстовое значение будет в кодировке UTF8. Это относится к любым типам проверок: агент Zabbix, SSH, Telnet и т. д.

Различные контролируемые системы/устройства и проверки могут возвращать в значении символы, отличные от ASCII. Для таких случаев почти все возможные ключи **zabbix** содержат дополнительный параметр ключа элемента данных — **<encoding>**. Этот параметр ключа является необязательным, но его следует указывать, если возвращаемое значение не в кодировке UTF8 и содержит символы, отличные от ASCII. В противном случае результат может быть неожиданным и непредсказуемым.

Ниже приводится описание поведения в таких случаях для различных серверных частей базы данных.

MySQL

Если значение содержит не-ASCII символ в кодировке, отличной от UTF8, этот символ и следующие за ним будут отброшены при сохранении этого значения в базе данных. Предупреждающие сообщения не будут записаны в *zabbix_server.log*.
Актуально как минимум для MySQL версии 5.1.61

PostgreSQL

Если значение содержит не-ASCII символ в кодировке, отличной от UTF8, это приведет к ошибке SQL-запроса (PGRES_FATAL_ERROR:ERROR 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

5 Поддержка больших файлов

Поддержка больших файлов, часто сокращаемая как LFS, — это термин, обозначающий возможность работать с файлами размером более 2 ГБ в 32-битных операционных системах. Поддержка больших файлов влияет как минимум на мониторинг файлов журналов и все элементы данных vfs.file.*. Поддержка больших файлов зависит от возможностей системы на момент компиляции Zabbix, но полностью отключена в 32-битной Solaris из-за несовместимости с procfs и swapctl.

6 Датчик

Обзор

Каждый чип датчика получает собственный каталог в дереве sysfs `/sys/devices`. Чтобы найти все чипы датчиков, проще перейти по символьным ссылкам устройств из `/sys/class/hwmon/hwmon*`, где `*` — это действительное число (0,1,2,...).

Показания датчиков находятся либо в каталоге `/sys/class/hwmon/hwmon*/` для виртуальных устройств, либо в каталоге `/sys/class/hwmon/hwmon*/device` для невиртуальных устройств. Файл с именем `name`, расположенный внутри каталогов `hwmon*` или `hwmon*/device`, содержит имя чипа, которое соответствует имени драйвера ядра, используемого этим чипом датчика.

В каждом файле содержится только одно значение показания датчика. Общая схема именования файлов, содержащих показания датчиков, внутри любого из упомянутых выше каталогов выглядит так: `<type><number>_<item>`, где

- **type** — для чипов датчиков это ”in” (напряжение), ”temp” (температура), ”fan” (вентилятор) и т. д.
- **item** — ”input” (измеренное значение), ”max” (верхний порог), ”min” (нижний порог) и т. д.
- **number** — всегда используется для элементов, которые могут присутствовать более одного раза (обычно начинается с 1, за исключением напряжений, которые начинаются с 0). Если файлы не относятся к конкретному элементу, они имеют простое имя без номера.

Информацию о датчиках, доступных на узле сети, можно получить с помощью инструментов **sensors-detect** и **sensors** (пакет lm-sensors). **Sensors-detect** помогает определить, какие модули необходимы для доступных датчиков. После загрузки модулей программу **sensors** можно использовать для отображения показаний всех чипов датчиков. Метки показаний датчиков, используемые этой программой, могут отличаться от общей схемы именования (`<type><number>_<item>`):

- если существует файл с именем `<type><number>_label`, то вместо имени `<type><number><item>` будет использоваться метка из этого файла;
- если файла `<type><number>_label` нет, программа ищет замену имени в `/etc/sensors.conf` (это также может быть `/etc/sensors3.conf` или другой файл).

Такая маркировка позволяет пользователю определить, какое именно оборудование используется. Если нет ни файла `<type><number>_label`, ни метки в файле конфигурации, тип оборудования можно определить по атрибуту name (`hwmon*/device/name`). Фактические имена датчиков, которые принимает `zabbix_agent`, можно получить, запустив программу **sensors** с параметром `-u` (`sensors -u`).

В программе **sensors** доступные датчики разделяются по типу шины (адаптер ISA, адаптер PCI, адаптер SPI, виртуальное устройство, интерфейс ACPI, адаптер HID).

Получение идентификаторов датчиков

Метки датчиков, выводимые командой `sensors`, не всегда можно использовать напрямую, поскольку обозначения различаются у разных производителей чипов. Например, вывод `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
```

Попытка использовать другие отображаемые метки (например, Vcore или +12V) не сработает:

```
zabbix_get -s 127.0.0.1 -k sensor[lm85-i2c-0-2e,Vcore]
ZBX_NOTSUPPORTED
```

Чтобы найти внутренний идентификатор датчика, который принимает Zabbix, выполните `sensors -u`. В 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 — in4. Согласно спецификации kernel hwmon sysfs, это напряжения на выводах чипа, и, вообще говоря, для них может потребоваться масштабирование.

```
zabbix_get -s 127.0.0.1 -k sensor[lm85-i2c-0-2e,in1]
1.301000
```

Zabbix может получать не только показания напряжения (in), но также тока (curr), температуры (temp) и скорости вращения вентилятора (fan).

Совместимость

Устаревшие дистрибутивы

Показания датчиков получаются из каталога /proc/sys/dev/sensors:

- **device** — имя устройства (если используется <mode>, это регулярное выражение)
- **sensor** — имя датчика (если используется <mode>, это регулярное выражение)
- **mode** — возможные значения: avg, max, min (если этот параметр опущен, device и sensor обрабатываются буквально)

Пример ключа: sensor[w83781d-i2c-0-2d,temp1].

Текущие дистрибутивы

Показания датчиков получаются из каталога /sys/class/hwmon:

- **device** — имя устройства (не регулярное выражение). Имя устройства может быть фактическим именем устройства (например, 0000:00:18.3) или именем, полученным с помощью программы sensors (например, k8temp-pci-00c3). Пользователь сам выбирает, какое имя использовать.
- **sensor** — имя датчика (не регулярное выражение).
- **mode** — возможные значения: avg, max, min (если этот параметр опущен, device и sensor 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].

7 Примечания о параметре 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

Поддерживаемые значения параметра 'memtype' в AIX приведены в таблице.

Поддерживаемое значение	Описание	Источник в структуре procentry64	Пытается быть совместимым 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	"ps gvw" столбец SIZE
dsize	Размер данных	pi_dsize	
tsize	Размер текста (кода)	pi_tsize	"ps gvw" столбец TSIZ
sdsize	Размер данных из разделяемой 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 процесса. В противном случае есть риск, что в результат proc.mem[] будут включены нежелательные процессы.

Пример:

```
$ zabbix_agentd -t proc.mem[,,,NonExistingProcess,rss]
proc.mem[,,,NonExistingProcess,rss]           [u|2879488]
```

Этот пример показывает, как указание только параметра командной строки (регулярного выражения для сопоставления) приводит к тому, что агент Zabbix учитывает сам себя — вероятно, это не то, что вам нужно.

2. Не используйте "ps -ef" для просмотра процессов — она показывает только некернельные процессы. Используйте "ps -Af", чтобы увидеть все процессы, которые будут видны агенту Zabbix.
3. Давайте на примере 'topasrec' рассмотрим, как proc.mem[] в агенте Zabbix выбирает процессы.

```
$ 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
```

У proc.mem[] есть аргументы:

```
proc.mem[<name>,<user>,<mode>,<cmdline>,<memtype>]
```

1-м критерием является имя процесса (аргумент <name>). В нашем примере агент Zabbix увидит его как 'topasrec'. Для совпадения необходимо либо указать 'topasrec', либо оставить его пустым. 2-м критерием является имя пользователя (аргумент <user>). Для совпадения необходимо либо указать 'root', либо оставить его пустым. 3-м критерием, используемым при выборе процесса, является 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 совпадения необходимо либо указать регулярное выражение, соответствующее этой строке, либо оставить его пустым.

Аргументы <mode> и <memtype> применяются после использования трех критериев, упомянутых выше.

FreeBSD

См. значения, поддерживаемые для параметра 'memtype' в FreeBSD, в таблице.

Поддерживаемое значение	Описание	Источник в структуре kinfo_proc	Пытается быть совместимым с
vsize	Размер виртуальной памяти	kp_eproc.e_vm.vm_map_size or ki_size	ps -o vsz
pmem	Процент реальной памяти	вычисляется на основе rss	ps -o pmem
rss	Размер резидентного набора	kp_eproc.e_vm.vm_rssize or ki_rssize	ps -o rss
size [1]	Размер процесса (код + данные + стек)	tsize + dsize + ssize	
tsize	Размер текста (кода)	kp_eproc.e_vm.vm_tsize or ki_tsize	ps -o tsiz
dsize	Размер данных	kp_eproc.e_vm.vm_dsize or ki_dsize	ps -o dsiz
ssize	Размер стека	kp_eproc.e_vm.vm_ssize or ki_ssize	ps -o 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 реальной 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 закрепленных страниц	VmPin
pte	Размер записей таблицы страниц	VmPTE
size	Размер кода процесса + сегментов данных + стека	VmExe + VmData + VmStk
stk	Размер сегмента стека	VmStk
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 самомониторинг процесса активной проверки агента Zabbix с помощью `proc.mem[...,...,...,...,data]` показывает значение, которое на 4 КБ больше, чем указано в строке `VmData` файла /proc/<pid>/status агента. В момент самоизмерения сегмент данных агента увеличивается на 4 КБ, а затем возвращается к предыдущему 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 — смотрите описание rss в ”man ps”.	pr_rssize	ps -o rss

Сноски

[1] Значение по умолчанию.

8 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 способ отображения текущей активности. Пользователь может увидеть эту активность, выполнив команды ps и top. Примеры таких программ: *PostgreSQL*, *Sendmail*, *Zabbix*.

Рассмотрим пример из Linux. Предположим, мы хотим отслеживать количество процессов агента Zabbix.

Команда ps показывает интересующие нас 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 дает нужный результат:

```
$ zabbix_get -s localhost -k 'proc.num[zabbix_agentd,zabbix]'
6
```

Теперь переименуем исполняемый файл zabbix_agentd в zabbix_agentd_30 и перезапустим его.

Теперь ps показывает:

```
$ 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 дает неверный результат:

```
$ zabbix_get -s localhost -k 'proc.num[zabbix_agentd_30,zabbix]'
1
```

Почему простое переименование исполняемого файла в более длинное имя привело к совершенно другому результату?

Агент Zabbix начинает с проверки имени процесса. Открывается файл /proc/<pid>/status и проверяется строка Name. В нашем случае строки Name выглядят так:

```
$ 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 усекается до 15 символов.

Похожий результат можно увидеть и с помощью 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, это не равно значению параметра name zabbix_agentd_30 в нашем proc.num[]. Не сумев сопоставить имя процесса из файла status, агент Zabbix переходит к файлу /proc/<pid>/cmdline.

То, как агент видит файл "cmdline", можно проиллюстрировать выполнением 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
```

Файлы /proc/<pid>/cmdline в нашем случае содержат невидимые, непечатаемые нулевые байты, используемые для завершения строк в языке *C*. В этом примере нулевые байты показаны как "<NUL>".

Агент Zabbix проверяет "cmdline" для основного процесса и получает zabbix_agentd_30, что соответствует значению нашего параметра name zabbix_agentd_30. Поэтому основной процесс учитывается элементом данных proc.num[zabbix_agentd

При проверке следующего процесса агент получает из файла cmdline zabbix_agentd_30: collector [idle 1 sec], и это не соответствует нашему параметру name zabbix_agentd_30. Поэтому учитывается только основной процесс, который не изменяет свою командную строку. Другие процессы агента изменяют свою командную строку и игнорируются.

Этот пример показывает, что параметр name нельзя использовать в proc.mem[] и proc.num[] для выбора процессов в данном случае.

Note:
Для элемента данных proc.get[], когда агент Zabbix проверяет "cmdline" для имени процесса, он использует только часть имени, начиная с последнего слеша и до первого пробела или двоеточия. Имя процесса, полученное из файла cmdline, будет использоваться только в том случае, если его начало полностью совпадает с сокращенным именем процесса в файле status. Алгоритм одинаков как для имени процесса в фильтре, так и для имени в выводе JSON.

Использование параметра cmdline с правильным регулярным выражением дает корректный результат:

```
$ zabbix_get -s localhost -k 'proc.num[,zabbix,,zabbix_agentd_30[ :]]'
6
```

Будьте внимательны при использовании элементов данных proc.get[], proc.mem[] и proc.num[] для мониторинга программ, которые изменяют свои командные строки.

Прежде чем использовать параметры name и cmdline в элементах данных proc.get[], proc.mem[] и proc.num[], может быть полезно проверить эти параметры с помощью элемента данных proc.num[] и команды ps.

Потоки ядра Linux

Потоки нельзя выбрать с помощью параметра cmdline в элементах данных proc.get[], proc.mem[] и proc.num[]

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 с помощью параметра процесса name:

```
$ zabbix_get -s localhost -k 'proc.num[kthreadd,root]'
1
```

Но выбор по параметру процесса cmdline не работает:

```
$ zabbix_get -s localhost -k 'proc.num[,root,,kthreadd]'
0
```

Причина в том, что агент Zabbix берет регулярное выражение, указанное в параметре cmdline, и применяет его к содержимому файла процесса /proc/<pid>/cmdline. Для потоков ядра файлы /proc/<pid>/cmdline пусты. Поэтому параметр cmdline никогда не совпадает.

Подсчет потоков в элементах данных proc.mem[] и proc.num[]

Потоки ядра Linux учитываются элементом данных proc.num[], но не сообщают об использовании памяти в элементе данных proc.mem[]. 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 произойдет, если существует пользовательский процесс с тем же именем, что и у потока ядра? Тогда это может выглядеть 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[] учитывает и поток ядра, и пользовательский процесс. proc.mem[] сообщает об использовании памяти только для пользовательского процесса и считает, что использование памяти потоком ядра равно 0. Это отличается от случая выше, когда сообщалось ZBX_NOTSUPPORTED.

Будьте внимательны при использовании элементов данных proc.mem[] и proc.num[], если имя программы случайно совпадает с именем одного из потоков.

Перед тем как указывать параметры в элементах данных proc.mem[] и proc.num[], может быть полезно проверить параметры с помощью элемента данных proc.num[] и команды ps.

9 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

Создает TCP-соединение и ожидает, что первые 4 символа ответа будут "220 ", затем отправляет "QUIT\r\n". Если порт не указан, по умолчанию используется порт 21.

http

Создает TCP-соединение, ничего не ожидая и ничего не отправляя. Если порт не указан, по умолчанию используется порт 80.

https

Использует libcurl (и работает только с ней), не проверяет подлинность сертификата, не проверяет имя узла сети в SSL-сертификате, получает только заголовок ответа (запрос HEAD). Если порт не указан, по умолчанию используется порт 443.

imap

Создает TCP-соединение и ожидает, что первые 4 символа ответа будут "* OK", затем отправляет "a1 LOGOUT\r\n". Если порт не указан, по умолчанию используется порт 143.

ldap

Открывает соединение с LDAP-сервером и выполняет операцию поиска LDAP с фильтром (objectClass=*). Ожидает успешного получения первого атрибута первой записи. Если порт не указан, по умолчанию используется порт 389.

nntp

Создает TCP-соединение и ожидает, что первые 3 символа ответа будут ”200” или ”201”, затем отправляет ”QUIT\r\n”. Если порт не указан, по умолчанию используется порт 119.

pop

Создает TCP-соединение и ожидает, что первые 3 символа ответа будут ”+OK”, затем отправляет ”QUIT\r\n”. Если порт не указан, по умолчанию используется порт 110.

smtp

Создает TCP-соединение и ожидает, что первые 3 символа ответа будут ”220”, за которыми следуют пробел, конец строки или дефис. Строки, содержащие дефис, относятся к многострочному ответу, и ответ будет считываться повторно до тех пор, пока не будет получена строка без дефиса. Затем отправляет ”QUIT\r\n”. Если порт не указан, по умолчанию используется порт 25.

ssh

Создает TCP-соединение. Если соединение установлено, обе стороны обмениваются строкой идентификации (SSH-major.minor-XXXX), где major и minor — версии протокола, а XXXX — строка. Zabbix проверяет, найдена ли строка, соответствующая спецификации, и затем отправляет в ответ строку ”SSH-major.minor-zabbix_agent\r\n” или ”0\n” в случае несоответствия. Если порт не указан, по умолчанию используется порт 22.

tcp

Создает TCP-соединение, ничего не ожидая и ничего не отправляя. В отличие от других проверок, требует обязательного указания параметра порта.

telnet

Создает TCP-соединение и ожидает приглашение для входа (’:’ в конце). Если порт не указан, по умолчанию используется порт 23.

Параметры элемента данных net.udp.service

ntp

Отправляет пакет SNTP по UDP и проверяет ответ в соответствии с RFC 4330, раздел 5. Если порт не указан, используется порт 123 по умолчанию.

10 Параметры proc.get

Обзор

Элемент данных **proc.get**[<name>,<user>,<cmdline>,<mode>] поддерживается в Linux, Windows, FreeBSD, OpenBSD и NetBSD.

Список параметров процесса, возвращаемых этим элементом данных, различается в зависимости от операционной системы и значения аргумента ’mode’.

Linux

Следующие параметры процесса возвращаются в Linux для каждого режима:

mode=process	mode=thread	mode=summary
pid: PID	pid: PID	name: имя процесса
ppid: PID родительского процесса	ppid: PID родительского процесса	processes: количество процессов
name: имя процесса	name: имя процесса	vsize: размер виртуальной памяти
cmdline: команда с аргументами	user: пользователь (реальный), от имени которого запущен процесс	pmem: процент использования физической памяти
user: пользователь (реальный), от имени которого запущен процесс	group: группа (реальная), от имени которой запущен процесс	rss: размер резидентного набора
group: группа (реальная), от имени которой запущен процесс	uid: ID пользователя	data: размер сегмента данных
uid: ID пользователя	gid: ID группы, от имени которой запущен процесс	exe: размер сегмента кода

mode=process	mode=thread	mode=summary
gid: ID группы, от имени которой запущен процесс	tid: ID потока	lib: размер разделяемых библиотек
vsize: размер виртуальной памяти	tname: имя потока	lck: размер заблокированной памяти
pmem: процент использования физической памяти	cputime_user: общее время CPU, затраченное в пользовательском режиме (значение выражено в тиках часов — разделите на `getconf CLK_TCK` или `sysconf(_SC_CLK_TCK)`, чтобы получить секунды)	pin: размер закреплённых страниц
rss: размер резидентного набора	cputime_system: общее время CPU, затраченное в системном режиме (значение выражено в тиках часов — разделите на `getconf CLK_TCK` или `sysconf(_SC_CLK_TCK)`, чтобы получить секунды)	pte: размер записей таблицы страниц
data: размер сегмента данных	state: состояние потока	size: размер сегментов кода процесса + данных + стека
exe: размер сегмента кода	ctx_switches: количество переключений контекста	stk: размер сегмента стека
hwm: пиковый размер резидентного набора	page_faults: количество страничных ошибок	swap: размер используемого пространства подкачки
lck: размер заблокированной памяти		cputime_user: общее время CPU, затраченное в пользовательском режиме (значение выражено в тиках часов — разделите на `getconf CLK_TCK` или `sysconf(_SC_CLK_TCK)`, чтобы получить секунды)
lib: размер разделяемых библиотек		cputime_system: общее время CPU, затраченное в системном режиме (значение выражено в тиках часов — разделите на `getconf CLK_TCK` или `sysconf(_SC_CLK_TCK)`, чтобы получить секунды)
peak: пиковый размер виртуальной памяти		ctx_switches: количество переключений контекста
pin: размер закреплённых страниц		threads: количество потоков
pte: размер записей таблицы страниц		page_faults: количество страничных ошибок
size: размер сегментов кода процесса + данных + стека		pss: память пропорционального размера набора
stk: размер сегмента стека		
swap: размер используемого пространства подкачки		
cputime_user: общее время CPU, затраченное в пользовательском режиме (значение выражено в тиках часов — разделите на `getconf CLK_TCK` или `sysconf(_SC_CLK_TCK)`, чтобы получить секунды)		
cputime_system: общее время CPU, затраченное в системном режиме (значение выражено в тиках часов — разделите на `getconf CLK_TCK` или `sysconf(_SC_CLK_TCK)`, чтобы получить секунды)		
state: состояние процесса (прозрачно получается из procfs, полная форма)		

mode=process	mode=thread	mode=summary
ctx_switches: количество переключений контекста threads: количество потоков page_faults: количество страничных ошибок pss: память пропорционального размера набора		

ОС на базе BSD

Следующие параметры процесса возвращаются в FreeBSD, OpenBSD и NetBSD для каждого режима:

mode=process	mode=thread	mode=summary
pid: PID	pid: PID	name: имя процесса
ppid: родительский PID	ppid: родительский PID	processes: количество процессов
jid: ID jail (только FreeBSD)	jid: ID jail (только FreeBSD)	vsize: размер виртуальной памяти
jname: имя jail (только FreeBSD)	jname: имя jail (только FreeBSD)	pmem: процент реальной памяти (только FreeBSD)
name: имя процесса	name: имя процесса	rss: размер резидентного набора
cmdline: команда с аргументами	user: пользователь (реальный), от имени которого выполняется процесс	size: размер процесса (код + данные + стек)
user: пользователь (реальный), от имени которого выполняется процесс	group: группа (реальная), от имени которой выполняется процесс	tsize: размер текста (кода)
group: группа (реальная), от имени которой выполняется процесс	uid: ID пользователя	dsize: размер данных
uid: ID пользователя	gid: ID группы, от имени которой выполняется процесс	ssize: размер стека
gid: ID группы, от имени которой выполняется процесс	tid: ID потока	cputime_user: общее время CPU, затраченное в пользовательском режиме (указывается в секундах)
vsize: размер виртуальной памяти	tname: имя потока	cputime_system: общее время CPU, затраченное в системном режиме (указывается в секундах)
pmem: процент реальной памяти (только FreeBSD)	cputime_user: общее время CPU, затраченное в пользовательском режиме (указывается в секундах)	ctx_switches: количество переключений контекста
rss: размер резидентного набора	cputime_system: общее время CPU, затраченное в системном режиме (указывается в секундах)	threads: количество потоков (не поддерживается в NetBSD)
size: размер процесса (код + данные + стек)	state: состояние потока	stk: размер сегмента стека
tsize: размер текста (кода)	ctx_switches: количество переключений контекста	page_faults: количество страничных ошибок
dsize: размер данных	io_read_op: количество случаев, когда системе требовалось выполнить ввод	fds: количество файловых дескрипторов (только OpenBSD)
ssize: размер стека	io_write_op: количество случаев, когда системе требовалось выполнить вывод	swap: размер используемого пространства подкачки
cputime_user: общее время CPU, затраченное в пользовательском режиме (указывается в секундах)		io_read_op: количество случаев, когда системе требовалось выполнить ввод
cputime_system: общее время CPU, затраченное в системном режиме (указывается в секундах)		io_write_op: количество случаев, когда системе требовалось выполнить вывод

mode=process	mode=thread	mode=summary
state: состояние процесса (ожидание диска/выполняется/спящий/остановлен трассировкой/зомби/другое) ctx_switches: количество переключений контекста threads: количество потоков (не поддерживается в NetBSD) page_faults: количество страничных ошибок fds: количество файловых дескрипторов (только OpenBSD) swap: размер используемого пространства подкачки io_read_op: количество случаев, когда системе требовалось выполнить ввод io_write_op: количество случаев, когда системе требовалось выполнить вывод		

Windows

Следующие параметры процесса возвращаются в Windows для каждого режима:

mode=process	mode=thread	mode=summary
pid: PID ppid: родительский PID name: имя процесса user: пользователь, от имени которого запущен процесс sid: SID пользователя vmsize: размер виртуальной памяти wkset: размер рабочего набора процесса cputime_user: общее время CPU, затраченное в пользовательском режиме (указывается в миллисекундах) cputime_system: общее время CPU, затраченное в системном режиме (указывается в миллисекундах) threads: количество потоков page_faults: количество страничных ошибок handles: количество дескрипторов io_read_b: количество прочитанных байт ввода-вывода io_write_b: количество записанных байт ввода-вывода	pid: PID ppid: родительский PID name: имя процесса user: пользователь, от имени которого запущен процесс sid: SID пользователя tid: ID потока	name: имя процесса processes: количество процессов vmsize: размер виртуальной памяти wkset: размер рабочего набора процесса cputime_user: общее время CPU, затраченное в пользовательском режиме (указывается в миллисекундах) cputime_system: общее время CPU, затраченное в системном режиме (указывается в миллисекундах) threads: количество потоков page_faults: количество страничных ошибок handles: количество дескрипторов io_read_b: количество прочитанных байт ввода-вывода io_write_b: количество записанных байт ввода-вывода io_read_op: операции чтения ввода-вывода io_write_op: операции записи ввода-вывода io_other_b: байты ввода-вывода, переданные не в операциях чтения и записи

mode=process	mode=thread	mode=summary
io_read_op: операции чтения ввода-вывода io_write_op: операции записи ввода-вывода io_other_b: байты ввода-вывода, переданные не в операциях чтения и записи io_other_op: операции ввода-вывода, отличные от операций чтения и записи		io_other_op: операции ввода-вывода, отличные от операций чтения и записи

11 Настройки недоступного/не отвечающего интерфейса узла сети

Обзор

Несколько параметров parameters определяют, как сервер Zabbix должен вести себя, 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 вносят вклад в общий статус 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 недоступными poller-процессами (или IPMI poller-процессами для IPMI-проверок).
По умолчанию интервал между последовательными проверками доступности составляет 15 секунд.

Attention:
Проверки, выполняемые асинхронными poller-процессами, не переводятся к недоступным poller-процессам.

В журнале сервера Zabbix недоступность обозначается сообщениями вида:

```
Zabbix "system.cpu.load[percpu,avg1]" "New host" : , 15
Zabbix "system.cpu.load[percpu,avg15]" "New host" : , 15
```

Сообщения журнала указывают точный элемент данных, для которого произошёл сбой, и его тип (агент Zabbix).

Note:
Параметр *Timeout* также влияет на то, насколько рано интерфейс будет перепроверен в период недоступности. Если Timeout установлен в 20 секунд, а UnreachableDelay — в 30 секунд, следующая проверка произойдёт через 50 секунд после первой попытки.

Параметр **UnreachablePeriod** определяет общую длительность периода недоступности. По умолчанию значение UnreachablePeriod составляет 45 секунд.
Это значение должно быть в несколько раз больше, чем UnreachableDelay, чтобы интерфейс был перепроверен несколько раз перед тем, как будет помечен как недоступный.

Внутренний элемент данных zabbix[host,active_agent,available] позволяет отслеживать доступность активных проверок в сценариях недоступности.

Переключение интерфейса обратно в состояние доступности

Когда период недоступности заканчивается, интерфейс снова опрашивается, при этом приоритет для элемента данных, который перевёл интерфейс в состояние недоступности, снижается. Если недоступный интерфейс снова появляется, мониторинг автоматически возвращается в нормальное состояние:

```
Zabbix "New host":
```

Note:
Как только интерфейс становится доступным, узел сети не опрашивает сразу все свои элементы данных по двум причинам:

- Это может перегрузить узел сети.
- Время восстановления интерфейса не всегда совпадает с запланированным временем опроса элементов данных.

Поэтому после того, как интерфейс становится доступным, элементы данных не опрашиваются немедленно, а переносятся на следующий цикл опроса.

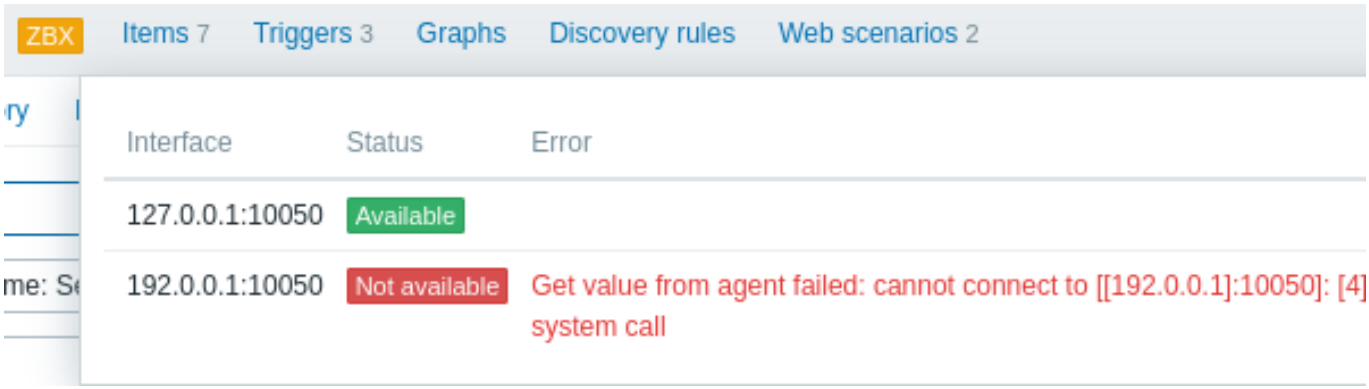
Недоступный интерфейс

После окончания периода UnreachablePeriod, если интерфейс не появился снова, он считается недоступным.

В журнале сервера это обозначается сообщениями следующего вида:

```
temporarily disabling Zabbix agent checks on host "New host": interface unavailable
```

а в веб-интерфейсе значок доступности узла сети меняется с зеленого/серого на желтый/красный (подробности о недоступном интерфейсе можно увидеть во всплывающей подсказке, которая отображается при наведении указателя мыши на значок доступности узла сети):



Параметр **UnavailableDelay** определяет, как часто интерфейс проверяется в период его недоступности.

По умолчанию это 60 секунд (то есть в данном случае «temporarily disabling» из сообщения журнала выше будет означать отключение проверок на одну минуту).

Когда соединение с интерфейсом восстанавливается, мониторинг также автоматически возвращается в нормальный режим:

```
enabling Zabbix agent checks on host "New host": interface became available
```

12 Удаленный мониторинг статистики Zabbix

Обзор

Некоторые внутренние метрики сервера и прокси Zabbix можно сделать доступными удаленно для другого экземпляра Zabbix или стороннего инструмента. Это может быть полезно, чтобы специалисты по поддержке/поставщики услуг могли удаленно отслеживать серверы/прокси Zabbix своих клиентов, либо в организациях, где Zabbix не является основным инструментом мониторинга, чтобы внутренние метрики Zabbix могли отслеживаться сторонней системой в рамках централизованного мониторинга.

Внутренняя статистика Zabbix предоставляется для настраиваемого набора адресов, перечисленных в новом параметре 'StatsAllowedIP' server/proxy. Запросы будут приниматься только с этих адресов.

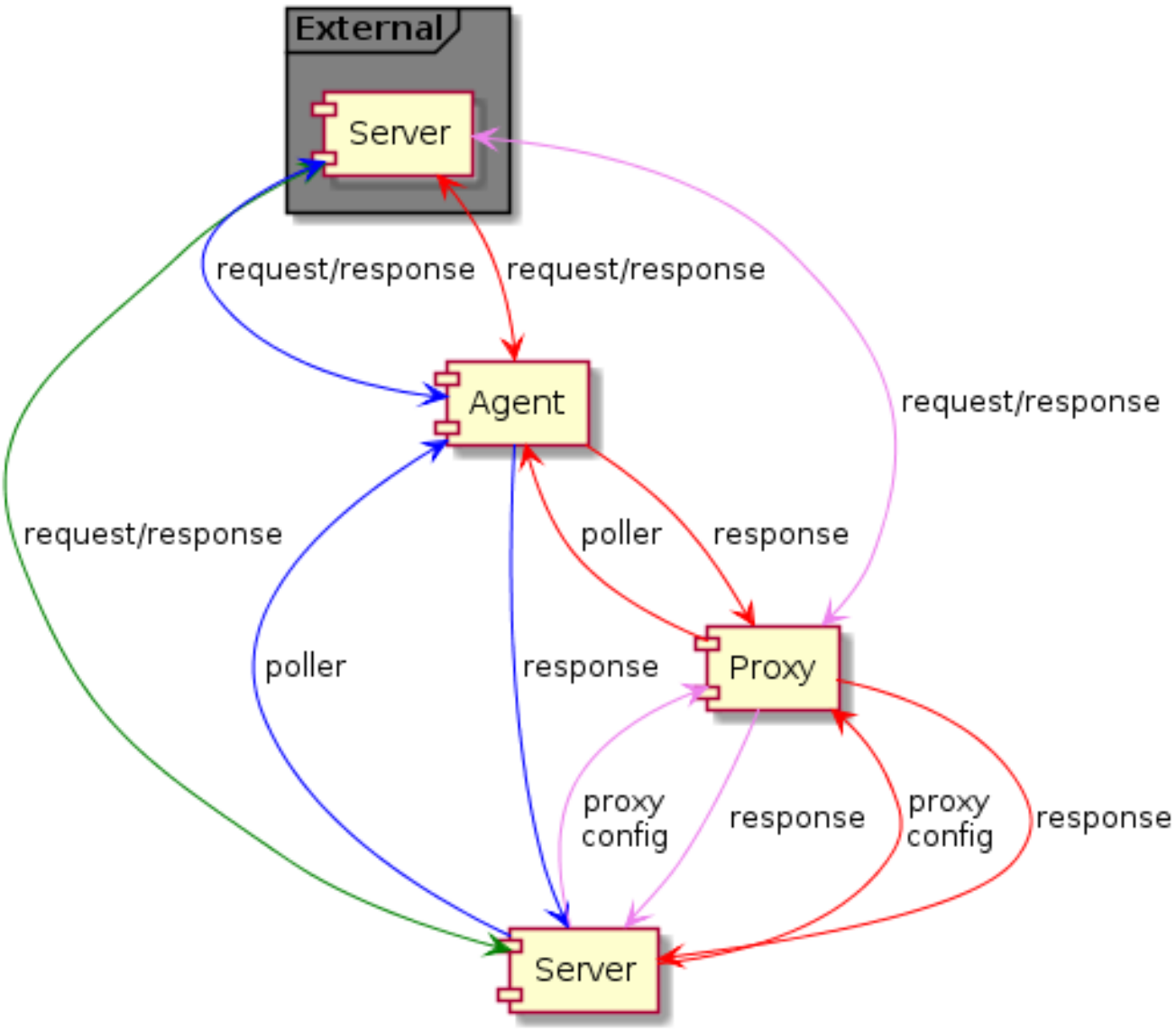
Элементы данных

Чтобы настроить запрос внутренней статистики на другом экземпляре Zabbix, вы можете использовать два элемента данных:

- внутренний элемент данных `zabbix[stats,<ip>,<port>]` — для прямых удалённых запросов к серверу/прокси Zabbix. <ip> и <port> используются для идентификации целевого экземпляра.
- элемент данных агента `zabbix.stats[<ip>,<port>]` — для удалённых запросов к серверу/прокси Zabbix через агент. <ip> и <port> используются для идентификации целевого экземпляра.

См. также: Внутренние элементы данных, Элементы данных агента Zabbix

Следующая диаграмма иллюстрирует использование того или иного элемента данных в зависимости от контекста.



-  - Сервер → внешний экземпляр Zabbix (`zabbix[stats,<ip>,<port>]`)
-  - Сервер → прокси → внешний экземпляр Zabbix (`zabbix[stats,<ip>,<port>]`)
-  - Сервер → агент → внешний экземпляр Zabbix (`zabbix.stats[<ip>,<port>]`)
-  - Сервер → прокси → агент → внешний экземпляр Zabbix (`zabbix.stats[<ip>,<port>]`)

Чтобы убедиться, что целевой экземпляр разрешает запросы к нему со стороны внешнего экземпляра, укажите адрес внешнего экземпляра в параметре 'StatsAllowedIP' на целевом экземпляре.

Доступные метрики

Элементы данных статистики собирают статистику пакетно и возвращают JSON, который служит основой для зависимых элементов данных, получающих из него свои данные. Следующие внутренние метрики возвращаются любым из двух элементов данных:

- `zabbix[boottime]`
- `zabbix[hosts]`
- `zabbix[items]`
- `zabbix[items_unsupported]`
- `zabbix[preprocessing]` (только сервер)
- `zabbix[preprocessing_queue]` (только сервер)
- `zabbix[process,<type>,<mode>,<state>]` (только статистика на основе типа процесса)
- `zabbix[rcache,<cache>,<mode>]`
- `zabbix[requiredperformance]`

- zabbix[triggers] (только сервер)
- zabbix[uptime]
- zabbix[vcache,buffer,<mode>] (только сервер)
- zabbix[vcache,cache,<parameter>]
- zabbix[version]
- zabbix[vmware,buffer,<mode>]
- zabbix[wcache,<cache>,<mode>] (тип кэша 'trends' — только сервер)

Шаблоны

Шаблоны доступны для удаленного мониторинга внутренних метрик сервера Zabbix или прокси из внешнего экземпляра:

- Состояние удаленного сервера Zabbix
- Состояние удаленного прокси Zabbix

Обратите внимание, что для использования шаблона для удаленного мониторинга нескольких внешних экземпляров требуется отдельный узел сети для мониторинга каждого внешнего экземпляра.

Процесс trapper

Получение запросов внутренних метрик от внешнего экземпляра Zabbix обрабатывается процессом trapper, который проверяет запрос, собирает метрики, создает буфер данных JSON и отправляет подготовленный JSON обратно, например, от 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 ещё два элемента данных, специально предназначенные для удалённого запроса статистики внутренней очереди на другом экземпляре Zabbix:

- внутренний элемент данных `zabbix[stats,<ip>,<port>,queue,<from>,<to>]` — для прямых запросов внутренней очереди к удалённому серверу/прокси Zabbix
- элемент данных агента `zabbix.stats[<ip>,<port>,queue,<from>,<to>]` — для запросов внутренней очереди к удалённому серверу/прокси Zabbix через агент

См. также: Внутренние элементы данных, Элементы данных агента Zabbix

13 Настройка Kerberos в Zabbix

Обзор

Аутентификация Kerberos может использоваться в веб-мониторинге и HTTP-элементах данных в Zabbix.

На этой странице описан пример настройки Kerberos для сервера Zabbix, чтобы выполнять веб-мониторинг `www.example.com` с использованием принципала Kerberos для процесса Zabbix в Debian/Ubuntu.

Настройка

1. Установите KDC и клиентские утилиты:

```
sudo apt update
sudo apt install krb5-kdc krb5-admin-server krb5-user
```

Во время установки 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 (необязательно, для локального тестирования).

Отредактируйте /etc/hosts и добавьте запись для вашего DC и веб-сервера, если у вас нет DNS:

```
sudo vi /etc/hosts
```

Пример строки, которую можно добавить:

```
192.168.1.100  dc01.example.com dc01
```

3. Настройте клиент Kerberos и realm 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 domain→realm, чтобы сопоставление hostname→realm работало корректно.
Несоответствия здесь вызывают ошибки `Server not found in Kerberos database`.

4. Инициализируйте базу данных Kerberos (однократно, на узле KDC).
При появлении запроса задайте надежный мастер-пароль:

```
sudo krb5_newrealm
```

5. Создайте principal `HTTP/host.fqdn@REALM`, используя точное имя узла, которое будут использовать клиенты; предпочтительно в нижнем регистре (например, `HTTP/web.example.com@EXAMPLE.COM`).
Несоответствие регистра/имени вызывает `Server not found in Kerberos database`.

```
sudo kadmin.local
```

Внутри `kadmin.local`:

```
addprinc kerb-admin@EXAMPLE.COM     # administrative principal
addprinc -randkey HTTP/dc01.example.com@EXAMPLE.COM
ktadd -k /etc/apache2/http.keytab HTTP/dc01.example.com@EXAMPLE.COM
quit
```

Переместите keytab на веб-узел сети (или оставьте локально, если это одна и та же машина) и задайте права, позволяющие Apache использовать его:

```
chown www-data:www-data /etc/apache2/http.keytab
chmod 600 /etc/apache2/http.keytab
#### verify
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 (скорректируйте `DocumentRoot` / путь к вашему пользовательскому интерфейсу Zabbix):

```
sudo vi /etc/apache2/sites-available/zabbix.conf
```

Внутри `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/запустите службы KDC и проверьте прослушиваемые порты (узел KDC):

```
sudo systemctl enable --now krb5-kdc krb5-admin-server
ss -tnlp | grep :80    # or: sudo netstat -tnlp | grep :80
```

9. Получите TGT для тестирования (запускайте от имени пользователя, который будет использовать билет).

Ожидается, что в списке билетов будет `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 интерактивных тестов SSO в браузере).
Билеты, выданные другому пользователю ОС, не будут видны, если не настроены `KRB5CCNAME` и соответствующие права доступа.

```
kinit kerb-admin@EXAMPLE.COM
klist
```

10. Проверьте обмен SPNEGO с помощью curl (с клиента с действительным TGT).
Ответ 200 `OK` (или перенаправление в приложение) означает, что SPNEGO выполнен успешно:

```
curl -v --negotiate -u : http://dc01.example.com/
```

11. При необходимости, если пользовательский интерфейс Zabbix должен принимать входы, аутентифицированные по HTTP, включите HTTP-аутентификацию во веб-интерфейсе Zabbix (ui/conf/zabbix.conf.php):

```
$ALLOW_HTTP_AUTH = true;
```

В веб-интерфейсе перейдите в *Users* > *Authentication* и откройте вкладку *HTTP settings*.
Установите флажок *Enable HTTP authentication* и нажмите *Ok* во всплывающем окне.
Выберите "HTTP login form" в выпадающем списке *Default login form*.
Определите, соответствует ли параметр *Case-sensitive login* политике вашего каталога.
Нажмите кнопку *Update* для завершения.

12. Настройка браузера (в качестве примера используется Firefox): установите `network.negotiate-auth.trusted-uris` в значение узла(ов), выполняющих Negotiate (`dc01.example.com`), чтобы браузер автоматически отправлял токены Kerberos.

Внутри `about:config`:

```
network.negotiate-auth.trusted-uris = dc01.example.com
```

Теперь при переходе на `http://dc01.example.com` вы должны сразу входить в Zabbix без формы входа.

13. Поддерживайте ключи/билеты в актуальном состоянии.
Стандартное время жизни билета Kerberos составляет примерно 10 часов.
Добавьте cron/systemd timer, чтобы избежать истечения срока действия:

```
####for the web service
kinit -kt /etc/apache2/http.keytab HTTP/dc01.example.com@EXAMPLE.COM
####for the monitoring user
kinit -kt /var/lib/zabbix/kerb.keytab kerb-admin@EXAMPLE.COM
```

14. Проверки обслуживания:

- `klist -k /etc/apache2/http.keytab` — убедитесь, что service principal присутствует 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/правами или отсутствие principal).
- Возврат 401/403 от `curl --negotiate` часто означает неверный principal, отсутствие билета, несоответствие заголовка host или проблему с правами доступа к файловой системе; проверьте журналы и сопоставления доменов в `/etc/krb5.conf`.

Примечания по безопасности и правам доступа к файлам

Файлы keytab должны быть доступны для чтения только той учетной записи, которой они необходимы. Примеры прав доступа: 0400 с владельцем `zabbix:zabbix` для keytab пользователя zabbix или 0440 и `root:www-data` для keytab Apache.

Избегайте хранения на узле сети долгоживущих паролей в открытом виде. По возможности используйте keytab или машинные субъекты безопасности, присоединенные к домену.

При запуске тестов или сценариев, которые задают `KRB5CCNAME` или копируют keytab, повторно проверяйте владельца и права доступа после выполнения операции — отказ веб-сервера принимать учетные данные часто связан именно с проблемой прав доступа к файлу.

14 Параметры modbus.get

Обзор

В таблице ниже представлены детальные описания параметров элемента данных.

Параметры

Параметр	Описание	По умолчанию	Пример
endpoint	Протокол и адрес конечной точки, задаваемые как `protocol://connection_string` Возможные значения протокола: *rtu*, *ascii* (только Agent 2), *tcp* Формат строки подключения: для *tcp* — `address:port` для последовательной линии: *rtu*, *ascii* — `port_name:speed:params` где 'speed' — 1200, 9600 и т. д. 'params' — биты данных (5,6,7 или 8), четность (n,e или o для none/even/odd), стоповые биты (1 или 2)	protocol: none *rtu/ascii* protocol: port_name: none speed: 115200 params: 8n1 *tcp* protocol: address: none port: 502	tcp://192.168.6.1:511 tcp://192.168.6.2 tcp://[::1]:511 tcp://::1 tcp://localhost:511 tcp://localhost rtu://COM1:9600:8n ascii://COM2:1200:7o2 rtu://ttyS0:9600 ascii://ttyS1
slave id	Адрес устройства Modbus, для которого предназначен запрос (от 1 до 247), см. MODBUS Messaging Implementation Guide (стр. 23) tcp-устройство (не GW) будет игнорировать это поле	serial: 1 tcp: 255 (0xFF)	2
function	Пустое значение или значение поддерживаемой функции: 1 — Read Coil, 2 — Read Discrete Input, 3 — Read Holding Registers, 4 — Read Input Registers	empty	3
address	Адрес первого регистра, coil или input. Если 'function' пуст, то 'address' должен находиться в диапазоне: Coil — 00001 - 09999 Discrete input — 10001 - 19999 Input register — 30001 - 39999 Holding register — 40001 - 49999 Если 'function' не пуст, поле 'address' будет иметь значение от 0 до 65535 и использоваться без изменений (PDU)	empty function: 00001 non-empty function: 0	9999
count	Количество последовательных значений 'type', которые будут считаны с устройства, где: для Coil или Discrete input 'type' = 1 бит для остальных случаев: (count*sizeof(type))/2 = фактическое количество регистров для чтения Если 'offset' не равен 0, его значение будет добавлено к 'фактическому количеству' Допустимый диапазон для 'фактического количества' — 1:65535	1	2

Параметр	Описание	По умолчанию	Пример
type	Тип данных: для Read Coil и Read Discrete Input — *bit* для Read Holding Registers и Read Input Registers: *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
endianness	Тип порядка байтов: *be* — Big Endian *le* — Little Endian *mbe* — Mid-Big Endian *mle* — Mid-Little Endian Ограничения: для 1 бита — be для 8 бит — be,le для 16 бит — be,le	be	le
offset	Количество регистров, начиная с 'address', результат чтения которых будет отброшен. Размер каждого регистра составляет 16 бит (необходимо для поддержки оборудования, которое не поддерживает произвольный доступ при чтении).	0	4

15 Создание пользовательских имён счётчиков производительности для VMware

Обзор

Путь к счетчику производительности VMware имеет формат `group/counter[rollup]`, где:

- `group` — группа счетчика производительности, например *cpu*
- `counter` — имя счетчика производительности, например *usagemhz*
- `rollup` — тип агрегации счетчика производительности, например *average*

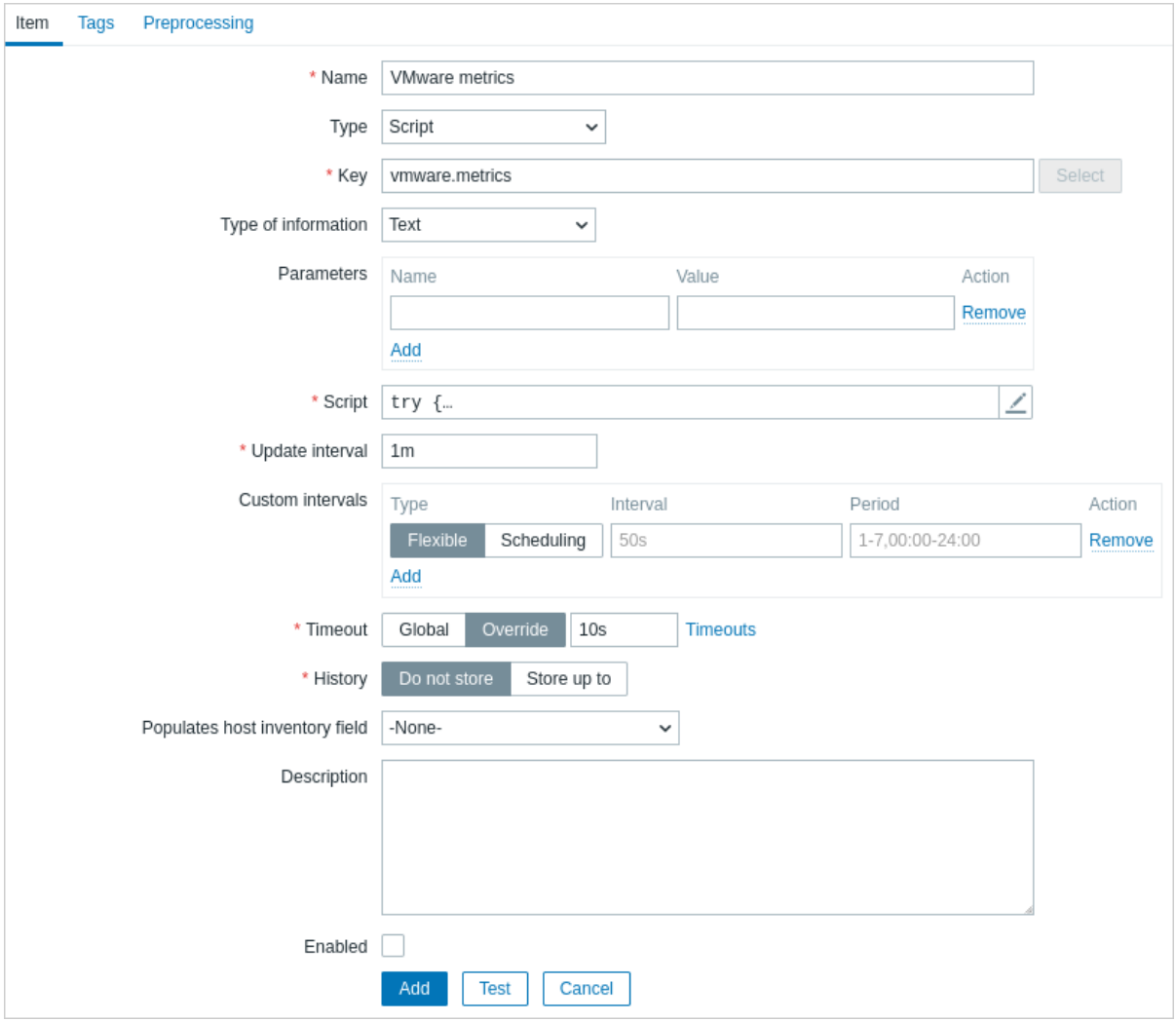
Таким образом, приведенный выше пример даст следующий путь к счетчику: `cpu/usagemhz[average]`

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-скрипт в Zabbix.

Настройка

1. Создайте отключенный элемент данных Script на основном VMware узле сети (где присутствует элемент данных **eventlog[]**) со следующими параметрами:



- *Имя*: VMware metrics
- *Тип*: Script
- *Ключ*: vmware.metrics
- *Тип информации*: Text
- *Скрипт*: скопируйте и вставьте скрипт, приведенный ниже
- *Таймаут*: 10
- *История*: Не хранить
- *Включен*: не отмечено

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 нажмите *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]`

16 Возвращаемые значения для system.sw.packages.get

Обзор

В этом разделе приводятся сведения о возвращаемом значении для элемента данных агента Zabbix `system.sw.packages.get`.

Подробности

Вывод этого элемента данных представляет собой массив объектов, каждый из которых содержит следующие ключи:

- **name** — имя пакета
- **manager** — менеджер пакетов, сообщивший эти данные (`rpm`, `dpkg`, `pacman`, `pkgtool` или `portage`)
- **version** — версия пакета
- **size** — размер пакета в байтах без сжатия (если недоступно, устанавливается в 0)
- **arch** — архитектура пакета
- **buildtime** — объект с 2 записями:
  - **timestamp** — метка времени UNIX, когда пакет был собран (если недоступно, устанавливается в 0)
  - **value** — дата и время сборки пакета в удобочитаемом виде (если недоступно, устанавливается в пустую строку)
- **installtime** — объект с 2 записями:
  - **timestamp** — метка времени UNIX, когда пакет был установлен (если недоступно, устанавливается в 0)
  - **value** — дата и время установки пакета в удобочитаемом виде (если недоступно, устанавливается 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": ""
        }
    }
]
```

17 Возвращаемые значения для net.dns.get

Обзор

В этом разделе приведены сведения о возвращаемом значении для элемента данных Zabbix агент 2 `net.dns.get`.

Подробности

Вывод этого элемента данных представляет собой объект, содержащий информацию о DNS-записи на основе параметров, указанных в ключе элемента данных.

Например, элемент данных `net.dns.get[,example.com]` может вернуть следующий JSON для отклонённого запроса:

```
{
    "flags": [
        "RA"
    ],
    "query_time": "0.019030",
    "question_section": [
        {
            "qclass": "IN",
            "qname": "example.com.",
            "qtype": "SOA"
        }
    ],
    "response_code": "REFUSED",
    "zbx_error_code": 0
}
```

При указании IP-адреса DNS-сервера элемент данных `net.dns.get[192.0.2.0,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 возникает проблема с подключением, элемент данных `net.dns.get[192.0.2.0,example.com]` может вернуть ошибку:


```
{
    "zbx_error_code": -1,
    "zbx_error_msg": "Communication error: read udp 192.0.2.0:12345->192.0.2.0:53: i/o timeout"
}
```


Возможны следующие типы кодов ошибок:

Сценарий	"zbx_error_code"	"zbx_error_msg"
Ошибок нет, и DNS-ответ был получен и разобран.	0	
DNS недоступен.	-1	"Communication error"
Ошибка возникает во время разбора JSON	-2	"Received unexpected response"

При использовании дополнительных параметров элемент данных `net.dns.get[192.0.2.0,example.com,ANY,5,5,tcp,"cdflag` 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": "192.0.2.0"
            },
            "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
}
```


См. также

Для получения дополнительной информации о DNS-записях см.:

- Доменные имена — реализация и спецификация
- Параметры системы доменных имён (DNS)

18 Примечания по элементам данных system.cpu.util в Windows

Элемент данных system.cpu.util предоставляет процент использования CPU.

Когда на Zabbix агенте для Windows запускается процесс сбора, 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 данных в буфере используется для `system.cpu.util[total]`.

Zabbix автоматически выбирает различные счетчики производительности для поддержки как NUMA-систем, так и более старых версий Windows без поддержки групп процессоров.

Если количество логических процессоров (потоков) меньше или равно 64, а количество групп процессоров равно 1, тогда:

```
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
```

19 Поддержка больших значений JSON

Чтобы обеспечить надежную обработку больших значений с JSON типом данных (1MiB и более), при необходимости проверьте и примените следующие изменения конфигурации:

1. Если вы используете MySQL или MariaDB, увеличьте максимальный допустимый размер пакета в `/etc/mysql/my.cnf` (в некоторых системах по умолчанию он может составлять 1MB):

```
[mysqld]
max_allowed_packet=128M

[mysqldump]
max_allowed_packet=1G
```

2. При необходимости скорректируйте дополнительные системные переменные сервера MySQL/MariaDB и системные переменные InnoDB. 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 `HistoryCacheSize` до 2G в файле конфигурации сервера Zabbix.

4. Если вы используете Zabbix прокси, увеличьте `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, тайм-ауты элемента данных или значение параметра `Timeout` в файлах конфигурации Zabbix сервера/прокси и агента). В противном случае обмен сообщениями между ними может 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 скрипта или тайм-аут тестирования элемента данных.

5 Поддерживаемые функции

См. Выражение триггера.

6 Макросы

Можно использовать готовые поддерживаемые макросы и пользовательские макросы, поддерживаемые в зависимости от расположения.

1 Макросы, поддерживаемые местоположением

Обзор

Таблица макросов, поддерживаемых по расположению содержит справочную информацию обо всех макросах (встроенных, выражений и определяемых пользователем), которые поддерживаются в конкретном расположении.

Если поддерживаются все макросы из связанной группы, то указывается только имя группы. Нажмите на имя каждого макроса или имя группы, чтобы увидеть полные сведения о макросах.

Список сведений о макросах содержит подробный перечень всех **встроенных** макросов, сгруппированных по области применения. Обратите внимание, что для настройки значений макросов (например, чтобы сократить их или извлечь определённые подстроки) 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 всё поле целиком; несколько макросов/смешивание 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 инвентарных данных 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 функции 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 инвентарных данных 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 {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	
Авторегистрация	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 инвентарных данных 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)	
Поля URL, Имя пользователя, Пароль, 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 всё поле целиком; несколько макросов/смешивание с текстом не поддерживаются.
Имя графика	Макрос выражения: {?EXPRESSION}	В этом расположении в данном макросе поддерживается только одна функция **avg**, **last**, **max**, **min** с секундами в качестве параметра (можно использовать суффиксы времени). Макрос {HOST.HOST<1-9>} может использоваться как узел сети внутри макроса (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 всё поле целиком; несколько макросов/смешивание с текстом не поддерживаются

Расположение	Поддерживаемые макросы	Комментарии
Порт интерфейса	Пользовательские макросы	Один макрос должен заполнять всё поле целиком; несколько макросов/смешивание 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 в интерфейс Zabbix агент, SNMP, JMX или IPMI узла сети в указанном порядке приоритета либо в 'UNKNOWN', если у узла сети нет ни одного интерфейса.
Поля Update interval, Custom intervals, Timeout (доступно для поддерживаемых типов элементов данных), Store up to (для history/trends)	Пользовательские макросы	Один макрос должен заполнять всё поле целиком; несколько макросов/смешивание с текстом не поддерживаются.
Поля Delete lost resources, Disable lost resources	Пользовательские макросы	Эти поля поддерживаются только для правил LLD. Один макрос должен заполнять всё поле целиком; несколько макросов/смешивание с текстом не поддерживаются.

Расположение	Поддерживаемые макросы	Комментарии
Параметры шага предобработки (включая пользовательские скрипты), *параметры пользовательской обработки ошибок* (*поля Set value to* и *Set error to*)	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.
Регулярные выражения фильтра переопределения	Пользовательские макросы	Эти поля поддерживаются только для правил LLD.
Операции переопределения: *Update interval*, *History storage period*, *Trend storage period* (для прототипов элементов данных)	Пользовательские макросы	Эти поля поддерживаются только для правил LLD. Один макрос должен заполнять всё поле целиком; несколько макросов/смешивание с текстом не поддерживаются.
Типы элементов данных		
Browser элемент данных *имена и значения параметров*	Макросы узла сети: {HOST.CONN}, {HOST.DNS}, {HOST.HOST}, {HOST.IP}, {HOST.NAME}, {HOST.PORT} Макросы инвентарных данных узла сети Макросы элемента данных: {ITEM.ID}, {ITEM.KEY}, {ITEM.KEY.ORIG} Пользовательские макросы	
Script элемент данных *имена и значения параметров*		
Calculated/aggregate формула элемента данных (*константы выражения*, *параметры функции*, *параметры ключа элемента данных*, (только для aggregate элемента данных) *условия фильтра* (имя группы узлов сети и имя тега))	Пользовательские макросы	
Database monitor элемент данных *поля Username, Password, SQL query*	Пользовательские макросы	

Расположение	Поддерживаемые макросы	Комментарии
HTTP агент элемент данных *поля URL, Query, Request body, Headers (имена и значения), 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-адреса, содержащие секретный макрос, не будут работать, так как макрос в них будет разрешён как "******"
HTTP агент элемент данных *поле HTTP Proxy*	Макросы элемента данных: {ITEM.ID}, {ITEM.KEY}, {ITEM.KEY.ORIG} Пользовательские макросы	
HTTP агент элемент данных *поля Required status codes, HTTP authentication username/password, SSL key password*	Пользовательские макросы	
HTTP агент элемент данных *поле Allowed hosts*	Макросы узла сети: {HOST.CONN}, {HOST.DNS}, {HOST.HOST}, {HOST.IP}, {HOST.NAME}, {HOST.PORT} Пользовательские макросы	
JMX агент элемент данных *поле Endpoint*	Макросы узла сети: {HOST.CONN}, {HOST.DNS}, {HOST.HOST}, {HOST.IP}, {HOST.PORT} Пользовательские макросы	
Trapper элемент данных *поле Allowed hosts*		
SNMP агент элемент данных *поле SNMP OID*	Пользовательские макросы	
SSH and Telnet элемент данных *скрипт*	Макросы узла сети: {HOST.CONN}, {HOST.DNS}, {HOST.HOST}, {HOST.IP}, {HOST.NAME}, {HOST.PORT} Пользовательские макросы	
SSH агент элемент данных *поля Username, Public key file, Private key file, Password*	Пользовательские макросы	
Telnet агент элемент данных *поля Username, Password*	Пользовательские макросы	
Карта		
URL, имя URL	Макросы узла сети: {HOST.CONN}, {HOST.DESCRIPTION}, {HOST.DNS}, {HOST.HOST}, {HOST.ID}, {HOST.IP}, {HOST.NAME}, {HOST.PORT} {HOSTGROUP.ID} Макросы инвентарных данных узла сети Макросы карты {TRIGGER.ID}	

Расположение	Поддерживаемые макросы	Комментарии
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 инвентарных данных 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 расположении в данном макросе поддерживается только одна функция **avg**, **last**, **max**, **min** с секундами в качестве параметра (можно использовать суффиксы времени). Макрос {HOST.HOST<1-9>} может использоваться как узел сети внутри макроса. 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 связи	Макрос выражения: {?EXPRESSION} [1]	
Метка фигуры	Макрос выражения: {?EXPRESSION} [1]	
Текстовое поле в фигурах карты	{MAP.NAME}	
Сетевое обнаружение		
Поле Update interval	Пользовательские макросы (только глобальные)	Один макрос должен заполнять всё поле целиком; несколько макросов/смешивание 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 всё поле целиком; несколько макросов/смешивание 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 всё поле целиком; несколько макросов/смешивание с текстом не поддерживаются.
Скрипты		
Команды, текст подтверждения (Тип: script, ручное **действие узла сети**)	Макросы узла сети: {HOST.CONN}, {HOST.DNS}, {HOST.HOST}, {HOST.ID} [1], {HOST.IP}, {HOST.NAME}, {HOST.PORT} Макросы инвентарных данных узла сети [2] {MANUALINPUT} Макросы имени пользователя Пользовательские макросы	[1] Только для текста подтверждения [2] Поддерживается для Zabbix server и Zabbix proxy
Команды (Тип: script, ручное **действие события**)	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 инвентарных данных 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, ручное **действие события**)	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 инвентарных данных узла сети {MANUALINPUT} Макросы имени пользователя Пользовательские макросы	
URL, текст подтверждения (Тип: URL, ручное **действие узла сети/события**)		

Расположение	Поддерживаемые макросы	Комментарии
Имя и значение тега	Макросы узла сети: {HOST.CONN}, {HOST.DNS}, {HOST.HOST}, {HOST.ID}, {HOST.IP}, {HOST.NAME}, {HOST.PORT} Макросы инвентарных данных узла сети Макросы элемента данных: {ITEM.LASTVALUE*}, {ITEM.LOG.*}, {ITEM.VALUE*} Макросы триггера: {TRIGGER.ID} (только для значения 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 функции 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}	
Операционные 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-адреса, содержащие секретный макрос, не будут работать, так как макрос в них будет разрешён 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 User-Parameter Веб-сценарий	Позиционные макросы/ссылки: $1...$9	

Расположение	Поддерживаемые макросы	Комментарии
Поля Name, Agent, HTTP proxy	Пользовательские макросы	
Поле Update interval	Пользовательские макросы	Один макрос должен заполнять всё поле целиком; несколько макросов/смешивание 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 всё поле целиком; несколько макросов/смешивание 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 всё поле целиком; несколько макросов/смешивание с текстом не поддерживаются.
Скрипт JavaScript, имя параметра *скрипта JavaScript*	Пользовательские макросы (только глобальные)	
Menu entry name, Menu entry URL	{EVENT.TAGS.<tag name>}	
Виджеты		
Gauge виджет *Поле Description*	Макросы узла сети: {HOST.CONN}, {HOST.DESCRIPTION}, {HOST.DNS}, {HOST.HOST}, {HOST.ID}, {HOST.IP}, {HOST.NAME}, {HOST.PORT} Макросы инвентарных данных узла сети Макросы элемента данных (кроме {ITEM.STATE.ERROR}) Пользовательские макросы (только глобальные)	
Honeycomb виджет *Поле Primary/secondary label*		
Item value виджет *Поле Description*		
Top hosts виджет *Столбец Text data*	Макросы узла сети: {HOST.CONN}, {HOST.DESCRIPTION}, {HOST.DNS}, {HOST.HOST}, {HOST.ID}, {HOST.IP}, {HOST.NAME}, {HOST.PORT} Макросы инвентарных данных узла сети Пользовательские макросы (только глобальные)	

Расположение	Поддерживаемые макросы	Комментарии
URL (динамический) виджет *Поле URL*	Макросы узла сети: {HOST.CONN}, {HOST.DNS}, {HOST.HOST}, {HOST.ID}, {HOST.IP}, {HOST.NAME}, {HOST.PORT} Пользовательские макросы	
Другие расположения		
Поле Working time	Пользовательские макросы (только глобальные)	Один макрос должен заполнять всё поле целиком; несколько макросов/смешивание 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)	Рекомендуется использовать секретные макросы.
Плановые отчёты *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 многобайтные UTF-символы), должно быть заключено в двойные кавычки. Кавычки и обратные косые черты внутри имени тега в кавычках должны быть экранированы обратной косой чертой.

{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 причинного события, на которое ссылаются по имени тега.
 Имя тега, содержащее неалфавитно-цифровые символы (включая неанглийские многобайтные UTF-символы), должно быть заключено в двойные кавычки. Кавычки и обратные слеши внутри имени тега в кавычках должны быть экранированы обратным слешем.
 Используется в контексте события-симптома.

{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, важность, возраст, теги сервиса и значения.
 Этот макрос используется в контексте события-причины и возвращает информацию о событиях-симптомах.

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}

Разделённый запятыми список тегов события восстановления. Разворачивается в пустую строку, если теги отсутствуют.

{EVENT.RECOVERY.TAGSJSON}

Массив JSON, содержащий объекты тегов события восстановления. Разворачивается в пустой массив, 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}

Понятное для человека название действия(й), выполненного(ых) во время обновления проблемы.
 Преобразуется в следующие значения: *подтверждено*, *не подтверждено*, *прокомментировано*, *изменена важность с (исходная важность) на (обновленная важность)* и *закрыто* (в зависимости от того, сколько действий выполняется в одном обновлении).

{EVENT.UPDATE.ACTIONJSON}

Массив JSON, содержащий сведения о действии(ях), выполненном(ых) при обновлении 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 важности `old` и `new`).

 Например: `{"acknowledge":true,"message":"Monthly maintenance.","severity":{"old":2,"new":1},"suppress_until`

{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 на основе элемента 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. индексированные макросы.

{FUNCTION.RECOVERY.VALUE}

Значение N-й функции на основе элемента данных в выражении восстановления на момент события.*
 Учитываются только функции, у которых первым параметром является / / .

 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. индексированные макросы.

Макросы узла сети

{HOST.CONN}

IP-адрес или DNS-имя узла сети, в зависимости от настроек узла сети.
 В глобальных скриптах, полях IP/DNS интерфейса и веб-сценариях макрос будет разрешён в основной интерфейс агента. Если интерфейс агента не определён, будет использован основной интерфейс SNMP. Если интерфейс SNMP также не определён, будет использован основной интерфейс JMX. Если интерфейс JMX тоже не определён, будет использован основной интерфейс IPMI. Если у узла сети нет ни одного интерфейса, макрос разрешается в 'UNKNOWN'.

 Этот макрос можно использовать с числовым индексом как {HOST.CONN<1-9>}, чтобы указать на первый, второй, третий и т. д. узел сети в выражении триггера. См. индексированные макросы.

{HOST.DESCRIPTION}

Описание узла сети.

 Этот макрос можно использовать с числовым индексом как {HOST.DESCRIPTION<1-9>}, чтобы указать на первый, второй, третий и т. д. узел сети в выражении триггера. См. индексированные макросы.

{HOST.DNS}

DNS-имя узла сети.
 В глобальных скриптах, полях IP/DNS интерфейса и веб-сценариях этот макрос будет разрешаться в основной интерфейс агента. Если интерфейс агента не определён, будет использован основной интерфейс SNMP. Если интерфейс SNMP также не определён, будет использован основной интерфейс JMX. Если интерфейс JMX тоже не определён, будет использован основной интерфейс IPMI. Если у узла сети нет ни одного интерфейса, макрос разрешается в 'UNKNOWN'.

 Этот макрос можно использовать с числовым индексом как {HOST.DNS<1-9>}, чтобы указать на первый, второй, третий и т. д. узел сети в выражении триггера. См. индексированные макросы.

{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 как {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 как {HOST.ID<1-9>}, чтобы указать на первый, второй, третий и т. д. узел сети в выражении триггера. См. индексированные макросы.

{HOST.IP}

IP-адрес узла сети.
 В глобальных скриптах, полях IP/DNS интерфейса и веб-сценариях этот макрос будет разрешаться в основной интерфейс агента. Если интерфейс агента не определён, будет использован основной интерфейс SNMP. Если интерфейс SNMP также не определён, будет использован основной интерфейс JMX. Если интерфейс JMX тоже не определён, будет использован основной интерфейс IPMI. Если у узла сети нет ни одного интерфейса, макрос разрешается в 'UNKNOWN'.

 Этот макрос можно использовать с числовым индексом как {HOST.IP<1-9>}, чтобы указать на первый, второй, третий и т. д. узел сети в выражении триггера. См. индексированные макросы.

{HOST.METADATA}

Метаданные узла сети.

 Используется только для активной авторегистрации агента.

{HOST.NAME}

Видимое имя узла сети.

 Этот макрос можно использовать с числовым индексом как {HOST.NAME<1-9>}, чтобы указать на первый, второй, третий и т. д. узел сети в выражении триггера. См. индексированные макросы.

{HOST.PORT}

Порт узла сети (агента).
 В глобальных скриптах, полях IP/DNS интерфейса и веб-сценариях этот макрос будет разрешён в основной интерфейс агента. Если интерфейс агента не определён, будет использован основной интерфейс SNMP. Если интерфейс SNMP также не определён, будет использован основной интерфейс JMX. Если интерфейс JMX тоже не определён, будет использован основной интерфейс IPMI. Если у узла сети нет ни одного интерфейса, макрос будет разрешён в 'UNKNOWN'.

 Этот макрос можно использовать с числовым индексом как {HOST.PORT<1-9>}, чтобы указать на первый, второй, третий и т. д. узел сети в выражении триггера. См. индексированные макросы.

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 *Псевдоним* в инвентарных данных 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. индексированные макросы.

{INVENTORY.ASSET.TAG}

Поле *Asset tag* в инвентарных данных 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. индексированные макросы.

{INVENTORY.CHASSIS}

Поле *Шасси* в инвентарных данных 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. индексированные макросы.

{INVENTORY.CONTACT}

Поле *Contact* в инвентарных данных 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. индексированные макросы.

{INVENTORY.CONTRACT.NUMBER}

Поле *Номер контракта* в инвентарных данных 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. индексированные макросы.

{INVENTORY.DEPLOYMENT.STATUS}

Поле *Статус развертывания* в инвентарных данных 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. индексированные макросы.

{INVENTORY.HARDWARE}

Поле *Оборудование* в инвентарных данных 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. индексированные макросы.

{INVENTORY.HARDWARE.FULL}

Поле *Оборудование (полные сведения)* в инвентарных данных 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. индексированные макросы.

{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. индексированные макросы.

{INVENTORY.HOST.NETWORKS}

Поле *Сети узла сети* в инвентарных данных 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. индексированные макросы.

{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. индексированные макросы.

{INVENTORY.HW.ARCH}

Поле *Архитектура оборудования* в инвентарных данных 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. индексированные макросы.

{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. индексированные макросы.

{INVENTORY.HW.DATE.EXPIRY}

Поле *Дата окончания обслуживания оборудования* 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. индексированные макросы.

{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. индексированные макросы.

{INVENTORY.HW.DATE.PURCHASE}

Поле *Дата приобретения 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. индексированные макросы.

{INVENTORY.INSTALLER.NAME}

Поле *Имя установщика* в инвентарных данных 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. индексированные макросы.

{INVENTORY.LOCATION}

Поле *Location* в инвентарных данных 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. индексированные макросы.

{INVENTORY.LOCATION.LAT}

Поле *Широта местоположения* в инвентарных данных 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. индексированные макросы.

{INVENTORY.LOCATION.LON}

Поле *Долгота местоположения* в инвентарных данных 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. индексированные макросы.

{INVENTORY.MACADDRESS.A}

Поле *MAC address A* в инвентарных данных 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. индексированные макросы.

{INVENTORY.MACADDRESS.B}

Поле *MAC address B* в инвентарных данных 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. индексированные макросы.

{INVENTORY.MODEL}

Поле *Model* в инвентарных данных 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. индексированные макросы.

{INVENTORY.NAME}

Поле *Name* в инвентарных данных 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. индексированные макросы.

{INVENTORY.NOTES}

Поле *Заметки* в инвентарных данных 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. индексированные макросы.

{INVENTORY.OOB.IP}

Поле *OOB IP address* в инвентарных данных 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. индексированные макросы.

{INVENTORY.OOB.NETMASK}

Поле *маски 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. индексированные макросы.

{INVENTORY.OOB.ROUTER}

Поле *OOB router* в инвентарных данных 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. индексированные макросы.

{INVENTORY.OS}

Поле *OS* в инвентарных данных узла сети.

 Этот макрос можно использовать с числовым индексом, например {INVENTORY.OS<1-9>}, чтобы указать на первый, второй, третий и т. д. узел сети в выражении триггера. См. индексированные макросы.

{INVENTORY.OS.FULL}

Поле *ОС (полные сведения)* в инвентарных данных 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. индексированные макросы.

{INVENTORY.OS.SHORT}

Поле *OS (Short)* в инвентарных данных 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. индексированные макросы.

{INVENTORY.POC.PRIMARY.CELL}

Поле *Основной мобильный телефон контактного лица* в инвентарных данных узла сети.

 Этот макрос можно использовать с числовым индексом, например {INVENTORY.POC.PRIMARY.CELL<1-9>}, чтобы указать на первый, второй, третий и т. д. узел сети в выражении триггера. См. индексированные макросы.

{INVENTORY.POC.PRIMARY.EMAIL}

Поле *Primary POC email* в инвентарных данных 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 *Имя основного контактного лица* в инвентарных данных 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. индексированные макросы.

{INVENTORY.POC.PRIMARY.NOTES}

Поле *Примечания основного контактного лица* в инвентарных данных 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. индексированные макросы.

{INVENTORY.POC.PRIMARY.PHONE.A}

Поле *Основной телефон А контактного лица* в инвентарных данных 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. индексированные макросы.

{INVENTORY.POC.PRIMARY.PHONE.B}

Поле *Primary POC phone B* в инвентарных данных 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. индексированные макросы.

{INVENTORY.POC.PRIMARY.SCREEN}

Поле *Имя экрана основного контактного лица (POC)* в инвентарных данных 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. индексированные макросы.

{INVENTORY.POC.SECONDARY.CELL}

Поле *Secondary POC cell* в инвентарных данных 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. индексированные макросы.

{INVENTORY.POC.SECONDARY.EMAIL}

Поле *Secondary POC email* в инвентарных данных 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. индексированные макросы.

{INVENTORY.POC.SECONDARY.NAME}

Поле *Имя вторичного контактного лица* в инвентарных данных 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. индексированные макросы.

{INVENTORY.POC.SECONDARY.NOTES}

Поле *Заметки о вторичном контактном лице* в инвентарных данных 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. индексированные макросы.

{INVENTORY.POC.SECONDARY.PHONE.A}

Поле *Secondary POC phone A* в инвентарных данных 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. индексированные макросы.

{INVENTORY.POC.SECONDARY.PHONE.B}

Поле *Secondary POC phone B* в инвентарных данных 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. индексированные макросы.

{INVENTORY.POC.SECONDARY.SCREEN}

Поле *Имя экрана дополнительного контактного лица* в инвентарных данных 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. индексированные макросы.

{INVENTORY.SERIALNO.A}

Поле *Серийный номер A* в инвентарных данных 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. индексированные макросы.

{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. индексированные макросы.

{INVENTORY.SITE.ADDRESS.A}

Поле *Адрес площадки A* в инвентарных данных 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. индексированные макросы.

{INVENTORY.SITE.ADDRESS.B}

Поле *Адрес площадки 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. индексированные макросы.

{INVENTORY.SITE.ADDRESS.C}

Поле *Адрес площадки C* в инвентарных данных 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. индексированные макросы.

{INVENTORY.SITE.CITY}

Поле *Город площадки* в инвентарных данных 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. индексированные макросы.

{INVENTORY.SITE.COUNTRY}

Поле *Страна местоположения* в инвентарных данных 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. индексированные макросы.

{INVENTORY.SITE.NOTES}

Поле *Примечания о местоположении* в инвентарных данных 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 *Расположение стойки на площадке* в инвентарных данных 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. индексированные макросы.

{INVENTORY.SITE.STATE}

Поле *Штат/область местоположения* в инвентарных данных 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. индексированные макросы.

{INVENTORY.SITE.ZIP}

Поле *Почтовый индекс местоположения* в инвентарных данных 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. индексированные макросы.

{INVENTORY.SOFTWARE}

Поле *Software* в инвентарных данных 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. индексированные макросы.

{INVENTORY.SOFTWARE.APP.A}

Поле *Программное приложение A* в инвентарных данных 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. индексированные макросы.

{INVENTORY.SOFTWARE.APP.B}

Поле *Программное приложение B* в инвентарных данных 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. индексированные макросы.

{INVENTORY.SOFTWARE.APP.C}

Поле *Программное приложение C* в инвентарных данных 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. индексированные макросы.

{INVENTORY.SOFTWARE.APP.D}

Поле *Программное приложение 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. индексированные макросы.

{INVENTORY.SOFTWARE.APP.E}

Поле *Программное приложение 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. индексированные макросы.

{INVENTORY.SOFTWARE.FULL}

Поле *Программное обеспечение (полные сведения)* в инвентарных данных 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. индексированные макросы.

{INVENTORY.TAG}

Поле *Tag* в инвентарных данных 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. индексированные макросы.

{INVENTORY.TYPE}

Поле *Тип* в инвентарных данных 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. индексированные макросы.

{INVENTORY.TYPE.FULL}

Поле *Тип (полные сведения)* в инвентарных данных 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 инвентарных данных 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. индексированные макросы.

{INVENTORY.URL.B}

Поле *URL B* в инвентарных данных 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. индексированные макросы.

{INVENTORY.URL.C}

Поле *URL C* в инвентарных данных 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. индексированные макросы.

{INVENTORY.VENDOR}

Поле *Vendor* в инвентарных данных 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. индексированные макросы.

Макросы элементов данных

{ITEM.DESCRIPTION}

Описание N-го элемента данных в выражении триггера, которое вызвало уведомление.

 Этот макрос можно использовать с числовым индексом как {ITEM.DESCRIPTION<1-9>}, чтобы указать на первый, второй, третий и т. д. узел сети в выражении триггера. См. индексированные макросы.

{ITEM.DESCRIPTION.ORIG}

Исходное описание (с неразрешёнными макросами) N-го элемента данных в выражении триггера, которое вызвало уведомление.

 Этот макрос можно использовать с числовым индексом как {ITEM.DESCRIPTION.ORIG<1-9>}, чтобы указать на первый, второй, третий и т. д. узел сети в выражении триггера. См. индексированные макросы.

{ITEM.ID}

Числовой ID N-го элемента данных в выражении триггера, который вызвал уведомление.

 Этот макрос можно использовать с числовым индексом как {ITEM.ID<1-9>}, чтобы указать на первый, второй, третий и т. д. узел сети в выражении триггера. См. индексированные макросы.

{ITEM.KEY}

Ключ N-го элемента данных в выражении триггера, который вызвал 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 можно использовать с числовым индексом как {ITEM.KEY<1-9>}, чтобы указать на первый, второй, третий и т. д. узел сети в выражении триггера. См. индексированные макросы.

{ITEM.KEY.ORIG}

Исходный ключ (с неразрешёнными макросами) N-го элемента данных в выражении триггера, который вызвал уведомление.

 Этот макрос можно использовать с числовым индексом как {ITEM.KEY.ORIG<1-9>}, чтобы указать на первый, второй, третий и т. д. узел сети в выражении триггера. См. индексированные макросы.

{ITEM.LASTVALUE}

Последнее значение N-го элемента данных в выражении триггера, которое вызвало уведомление.
 В веб-интерфейсе будет преобразовано в *UNKNOWN*, если последнее значение истории было собрано более *Max history display period* назад (задаётся в разделе меню Administration → General).
 При использовании в имени проблемы макрос не будет преобразован в последнее значение элемента данных при просмотре событий проблем; вместо этого будет сохранено значение элемента данных на момент возникновения проблемы.
При использовании в уведомлениях в некоторых случаях макрос может не преобразоваться в последнее значение элемента данных на момент срабатывания триггера. Например, если элемент данных быстро получает два значения, ”A” и ”B”, и триггер срабатывает на ”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 преобразуется в значение на момент срабатывания триггера, обеспечивая использование правильного значения в уведомлении.
 Является псевдонимом для `last(/{HOST.HOST}/{ITEM.KEY})`.
 Преобразованное значение для текстовых/log элементов данных усекается до 20 символов в веб-интерфейсе в следующих местах:
- Рабочие 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 усекает 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 получением последнего 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. индексированные макросы.

{ITEM.LASTVALUE.DATE}

Дата, когда было получено последнее значение элемента данных.
 Отображается в формате `YYYYMMDD`.
 Часовой пояс сервера используется в именах триггеров (в Monitoring -> Problems list), именах событий, а также именах и значениях тегов. Во всех остальных случаях используется часовой пояс пользователя.

 Этот макрос можно использовать с числовым индексом, например {ITEM.LASTVALUE.DATE<1-9>}, чтобы указать на первый, второй, третий и т. д. элемент данных в выражении триггера. См. индексированные макросы.

{ITEM.LASTVALUE.TIME}

Время, когда было получено последнее значение элемента данных.
 Отображается в формате `HHMMSS`.
 Часовой пояс сервера используется в именах триггеров (в Monitoring -> Problems list), именах событий, а также именах и значениях тегов. Во всех остальных случаях используется часовой пояс пользователя.

 Этот макрос можно использовать с числовым индексом, например {ITEM.LASTVALUE.TIME<1-9>}, чтобы указать на первый, второй, третий и т. д. элемент данных в выражении триггера. См. индексированные макросы.

{ITEM.LASTVALUE.TIMESTAMP}

UNIX-метка времени, когда было получено 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. индексированные макросы.

{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. индексированные макросы.

{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. индексированные макросы.

{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. индексированные макросы.

{ITEM.LOG.NSEVERITY}

Числовая важность 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 данных в выражении триггера. См. индексированные макросы.

{ITEM.LOG.SEVERITY}

Текстовое значение важности 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 данных в выражении триггера. См. индексированные макросы.

{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 данных в выражении триггера. См. индексированные макросы.

{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. индексированные макросы.

{ITEM.LOG.TIMESTAMP}

UNIX timestamp, 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. индексированные макросы.

{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. индексированные макросы.

{ITEM.NAME.ORIG}

Исходное имя (с неразрешёнными макросами) элемента данных.

 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. индексированные макросы.

{ITEM.STATE}

Последнее состояние N-го элемента данных в выражении триггера, которое вызвало уведомление.
 Возможные значения: *Не поддерживается*, *Нормальное*.

 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. индексированные макросы.

{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}

Преобразуется в одно из следующих значений:

- историческое значение (на момент события) N-го элемента данных в выражении триггера, 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 вне 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 преобразовано в *UNKNOWN*, если значение из истории уже удалено или никогда не сохранялось.
Во втором случае, и только в веб-интерфейсе, будет преобразовано в *UNKNOWN*, если последнее значение истории было получено более *Max history display period* назад (задаётся в разделе меню Administration→General).
Преобр значение для текстовых/лог элементов данных усекается 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, вы можете использовать функции макросов, так как сервер не усекает значения. Например: `{{ITEM.VALUE}.regsub("(.*)", \1)}`.

Этот макрос можно использовать с числовым индексом, например {ITEM.VALUE<1-9>}, чтобы указать на первый, второй, третий и т. д. элемент данных в выражении триггера. См. индексированные макросы.

{ITEM.VALUE.AGE}

Время, прошедшее между получением 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 данных в выражении триггера. См. индексированные макросы.

{ITEM.VALUE.DATE}

Дата, когда было собрано значение элемента данных.
 Отображается в формате `YYYYMMDD`.
 Часовой пояс сервера используется в именах триггеров (в Monitoring -> Problems list), именах событий, а также именах и значениях тегов. Во всех остальных случаях используется часовой пояс пользователя.

Эта макрос может использоваться с числовым индексом, например {ITEM.VALUE.DATE<1-9>}, чтобы указать на первый, второй, третий и т. д. элемент данных в выражении триггера. См. индексированные макросы.

{ITEM.VALUE.TIME}

Время, когда было собрано значение элемента данных.
 Отображается в формате `HHMMSS`.
 В именах триггеров (в списке Monitoring -> Problems), именах событий, а также именах и значениях тегов используется часовой пояс сервера. Во всех остальных случаях используется часовой пояс пользователя.

 Этот макрос можно использовать с числовым индексом, например {ITEM.VALUE.TIME<1-9>}, чтобы указать на первый, второй, третий и т. д. элемент данных в выражении триггера. См. индексированные макросы.

{ITEM.VALUE.TIMESTAMP}

UNIX-метка времени, 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. индексированные макросы.

{ITEM.VALUETYPE}

Тип значения N-го элемента данных в выражении триггера, который вызвал уведомление.
 Возможные значения: *0* — Числовой с плавающей точкой, *1* — Символьный, *2* — Журнал, *3* — Числовой без знака, *4* — Текст.

 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. индексированные макросы.

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: *Не поддерживается*, *Нормальное*.

 Этот макрос можно использовать с числовым индексом, например {LLDRULE.STATE<1-9>}, чтобы указать на первый, второй, третий и т. д. элемент данных в выражении триггера. См. индексированные макросы.

{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.
 Преобразуется в одно из следующих значений:

- Прокси N-го элемента данных в выражении триггера (в уведомлениях на основе триггеров). Здесь можно использовать индексированные макросы;
- Прокси, который выполнил обнаружение (в уведомлениях об обнаружении). Используйте здесь {PROXY.DESCRIPTION} без индексации;
- Прокси, к которому зарегистрировался активный агент (в уведомлениях об авторегистрации). Используйте здесь {PROXY.DESCRIPTION} без индексации.

Этот макрос можно использовать с числовым индексом, например {PROXY.DESCRIPTION<1-9>}, чтобы указать на первый, второй, третий и т. д. элемент данных в выражении триггера. См. индексированные макросы.

{PROXY.NAME}

Имя прокси.
 Преобразуется в одно из следующих значений:

- Прокси N-го элемента данных в выражении триггера (в уведомлениях на основе триггеров). Здесь можно использовать индексированные макросы;
- Прокси, который выполнил обнаружение (в уведомлениях об обнаружении). Здесь используйте {PROXY.NAME} без индексации;
- Прокси, к которому зарегистрировался активный агент (в уведомлениях об авторегистрации). 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 данных 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}

Разделенный запятыми список тегов событий сервиса.
 Теги событий сервиса можно задать в разделе тегов конфигурации сервиса. Если теги отсутствуют, разворачивается в пустую строку.

{SERVICE.TAGSJSON}

Массив JSON, содержащий объекты тегов событий услуги.
 Теги событий услуги можно определить в разделе тегов конфигурации услуги. Разворачивается в пустой массив, если теги отсутствуют.

{SERVICE.TAGS.<tag name>}

Значение тега события сервиса, на которое ссылается имя тега.
 Теги событий сервиса можно определить в разделе тегов конфигурации сервиса.
 Имя тега, содержащее неалфавитно-цифровые символы (включая неанглийские многобайтные UTF-символы), должно быть заключено в двойные кавычки. Кавычки и обратные слеши внутри имени тега в кавычках должны быть экранированы обратным слешем.

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 на основе элементов данных вычисляются и заменяются результатами на момент генерации 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 в том виде, в котором они записаны в выражении. Может использоваться 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 индексированных макросов **{MACRO<1-9>}** может ссылаться только на N-й элемент данных, функцию или узел сети в поле *Expression* триггера:

- **{HOST.IP1}, {HOST.IP2}, {HOST.IP3}** преобразуются в IP-адреса первого, второго и третьего узлов сети в выражении триггера (если присутствуют).
- **{ITEM.VALUE1}, {ITEM.VALUE2}, {ITEM.VALUE3}** преобразуются в значения первого, второго и третьего элементов данных в выражении триггера на момент события (если присутствуют).
- **{FUNCTION.VALUE1}, {FUNCTION.VALUE2}, {FUNCTION.VALUE3}** преобразуются в значения первой, второй и третьей функций на основе элементов данных на момент события (если присутствуют).

В контексте триггеров индексированные макросы всегда относятся к полю *Expression* в конфигурации триггера, а не к *Recovery expression*. Например, в событии восстановления **{ITEM.VALUE2}** будет преобразован в 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 преобразован в узел сети второго элемента данных на графике.

Warning:
Индексированные макросы не будут преобразовываться ни в каком другом контексте, кроме упомянутых здесь случаев. В остальных контекстах используйте макросы без индекса (**{HOST.HOST}, {HOST.IP}** и т. д.).

2 Пользовательские макросы, поддерживаемые в зависимости от расположения

Обзор

В этом разделе содержится список мест, где поддерживаются пользовательские макросы.

Note:
Для *действий*, *обнаружения сети*, *прокси* и всех мест, перечисленных в разделе *Другие места* на этой странице, поддерживаются только пользовательские макросы глобального уровня. В указанных местах макросы уровня узла сети и уровня шаблона не будут разрешены.

Note:
Чтобы настраивать значения макросов (например, сокращать их или извлекать определенные подстроки), вы можете использовать функции макросов.

Действия

В действиях пользовательские макросы могут использоваться в следующих полях:

Расположение		Несколько макросов/смешивание с
Уведомления и команды на основе триггеров		да
Внутренние уведомления на основе триггеров		да
Уведомления об обновлении проблем		да
Уведомления и команды на основе сервисов		да
Уведомления об обновлении сервисов		да
Условие периода времени		нет
Операции		
	Длительность шага операции по умолчанию	нет
	Длительность шага	нет

Узлы сети/прототипы узлов сети

В конфигурации узла сети и прототипа узла сети пользовательские макросы можно использовать в следующих полях:

Расположение		Несколько макросов/смешивание с текстом[1]
IP/DNS интерфейса		только DNS
Порт интерфейса		нет
SNMP v1, v2		
	SNMP community	да
SNMP v3		
	Имя контекста	да
	Имя безопасности	да
	Парольная фраза аутентификации	да
	Парольная фраза приватности	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 могут использоваться в следующих полях:

Расположение	Несколько макросов/смешивание с текстом[1]
Имя элемента данных	да
Параметры ключа элемента данных	да

Расположение	Несколько макросов/смешивание с текстом[1]
Интервал обновления	нет
Пользовательские интервалы	нет
Таймаут (доступно 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 SSL-сертификата	да
Файл SSL-ключа	да
Пароль SSL-ключа	да
Разрешенные узлы сети	да
JMX агент	
Конечная точка JMX	да
Элемент данных-скрипт	
Имена и значения параметров	да
Элемент данных браузера	
Имена и значения параметров	да
SNMP агент	
SNMP OID	да
SSH агент	

Расположение	Несколько макросов/смешивание с текстом[1]
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 предобработки	
Параметры (включая пользовательские скрипты)	да
Параметры пользовательской обработки ошибок (поля *Установить значение в* и *Установить ошибку в*)	да

Обнаружение низкого уровня

В правиле обнаружения низкого уровня пользовательские макросы могут использоваться в следующих полях:

Расположение
Параметры ключа
Интервал обновления
Пользовательский интервал
Таймаут (доступно для поддерживаемых типов элементов данных)
Удаление потерянных ресурсов
Отключение потерянных ресурсов
Описание
SNMP агент
SNMP OID
SSH агент
Имя пользователя
Файл открытого ключа
Файл закрытого ключа
Пароль
Скрипт
TELNET агент
Имя пользователя
Пароль
Скрипт
Zabbix trapper
Разрешённые узлы сети
Монитор базы данных
Имя пользователя
Пароль
SQL-запрос
JMX агент
Конечная точка JMX
HTTP агент
URL[3]
Поля запроса
Тело запроса

Расположение		
		Заголовки (имена и значения) Требуемые коды состояния Имя пользователя для HTTP-аутентификации Пароль для HTTP-аутентификации
Фильтры		
		Регулярное выражение
Переопределения		
		Фильтры: регулярное выражение Операции: интервал обновления (для прототипов элементов Операции: период хранения истории (для прототипов элемен Операции: период хранения трендов (для прототипов элемен

Обнаружение сети

В правиле обнаружения сети пользовательские макросы можно использовать в следующих полях:

Расположение		Несколько макросов/смешивание с текстом[1]
Интервал обновления		нет
SNMP v1, v2		
	SNMP community	да
	SNMP OID	да
SNMP v3		
	Context name	да
	Security name	да
	Authentication passphrase	да
	Privacy passphrase	да
	SNMP OID	да

Прокси

В конфигурации прокси пользовательские макросы можно использовать в следующих полях:

Расположение	Несколько макросов/смешивание с текстом[1]
Адрес для активных агентов > *Порт* (если прокси принадлежит группе)	нет
Адрес и порт интерфейса (для пассивного прокси)	нет
Тайм-ауты для типов элементов данных	нет

Группы прокси

В конфигурации группы прокси пользовательские макросы можно использовать в следующих полях:

Расположение	Несколько макросов/смешивание с текстом[1]
Период переключения при отказе	нет
Минимальное количество прокси	нет

Шаблоны

В конфигурации шаблона пользовательские макросы можно использовать в следующих полях:

Расположение		Несколько макросов/смешивание с текстом[1]
Метки[2]		
	Имена меток	yes
	Значения меток	yes

Триггеры

В настройке триггера пользовательские макросы могут использоваться в следующих полях:

Расположение	Несколько макросов/смешивание с текстом[1]
Имя	да
Рабочие данные	да
Выражение (только в константах и параметрах функций; секретные макросы не поддерживаются)	да
Тег для сопоставления	да
Имя пункта меню	да
URL пункта меню[3]	да
Описание	да
Теги[2]	
Имена тегов	да
Значения тегов	да

Веб-сценарий

В конфигурации веб-сценария пользовательские макросы можно использовать в следующих полях:

Расположение		Несколько макросов/смешивание 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
	Таймаут	нет
	Требуемая строка	да
	Требуемые коды состояния	нет
Аутентификация		
	Пользователь	да
	Пароль	да
	SSL-сертификат	да
	Файл SSL-ключа	да
	Пароль SSL-ключа	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	да
Вебхуки	
JavaScript script	нет
Имя параметра JavaScript script	нет
Значение параметра JavaScript script	да
Панели	
Столбец типа данных *Text* в виджете панели *Top hosts*	да
Параметр *Description* в виджетах панели *Item value* и *Gauge*	да
Параметр *Text* основной/вторичной метки в виджете панели *Honeycomb*	да
Параметр *URL*[3] в виджете панели *URL*	да
Пользователи → *Пользователи* → *Способы оповещения*	
Когда активно	да
Администрирование → *Общие* → *GUI*	
Рабочее время	нет
Администрирование → *Общие* → *Тайм-ауты*	
Тайм-ауты для типов элементов данных	нет
Администрирование → *Общие* → *Коннекторы*	
URL	да
Имя пользователя	да
Пароль	да
Bearer token	да
Тайм-аут	нет
HTTP proxy	да
Файл SSL-сертификата	да
Файл SSL-ключа	да
Пароль SSL-ключа	да

Расположение	Несколько макросов/смешивание с текстом[1]
Оповещения → *Способы оповещения* → *Шаблоны сообщений*	
Тема	да
Сообщение	да
Оповещения → *Способы оповещения* → *Скрипт*	
Параметры скрипта	да
Оповещения → *Способы оповещения* → *Способ оповещения*	
Поля *Username* и *Password* для способа оповещения *Email* (когда *Authentication* установлено в "Username and password"; рекомендуется использовать секретные макросы)	да

Полный список всех макросов, поддерживаемых в Zabbix, см. в разделе поддерживаемые макросы.

Сноски

[1] Если несколько макросов в поле или макросы, смешанные с текстом, не поддерживаются для данного расположения, одно значение макроса должно заполнять всё поле целиком.

[2] Макросы, используемые в именах и значениях тегов, разрешаются только в процессе генерации события.

[3] URL-адреса, содержащие секретный макрос, не будут работать, так как макрос в них будет разрешён как "******".

8 Символы единиц измерения

Обзор

Работа с большими значениями, такими как 86400, 104857600 или 1000000, может быть затруднительной и приводить к ошибкам. Чтобы упростить настройку и повысить читаемость, Zabbix поддерживает символы единиц измерения (суффиксы), которые работают как множители значений.

Выражения триггеров без суффиксов:

```
last(/host/system.uptime)<86400
avg(/host/system.cpu.load,600s)<10
last(/host/vm.memory.size[available])<20971520
```

Выражения триггеров 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 сущностей, таких как элементы данных, виджеты и т. д., а также помочь отображать значения элементов данных в удобочитаемом формате.

Note:
Чтобы узнать, поддерживает ли поле настройки сущности суффиксы, всегда обращайтесь к соответствующей странице для настраиваемой сущности.

Суффиксы времени

Веб-интерфейс Zabbix поддерживает следующие суффиксы времени в настройках сущностей:

- **s** — секунды *(при использовании работает так же, как и необработанное значение)*
- **m** — минуты
- **h** — часы
- **d** — дни
- **w** — недели
- **M** — месяцы *(функции трендов only)*
- **y** — годы *(функции трендов only)*

Суффиксы времени поддерживаются только с целыми числами. Например, 1h поддерживается, а 1,5h или 1.5h не поддерживается; используйте вместо этого 90m.

Суффиксы размера памяти

Zabbix поддерживает следующие суффиксы размера памяти:

- **K** - килобайты
- **M** - мегабайты
- **G** - гигабайты
- **T** - терабайты

Суффиксы значений элементов данных

Суффиксы также можно использовать для отображения числовых значений элементов данных в удобочитаемом формате.

Чтобы включить это, используйте один из следующих суффиксов в поле *Units* при настройке элемента данных:

- **B** — байты
- **Bps** — байты в секунду
- **s** — секунды, отображаемые с использованием до трёх самых крупных ненулевых единиц времени
- **uptime** — прошедшее время в формате hh:mm:ss или N days, hh:mm:ss
- **unixtime** — временная метка Unix, отформатированная как yyyy.mm.dd hh:mm:ss

К тому, как эти суффиксы интерпретируются и отображаются, применяются следующие дополнительные правила:

- Для B и Bps Zabbix использует преобразование по основанию 2 (1K = 1024B) в соответствии со стандартом JEDEC.
- Для других единиц измерения (таких как Hz, W и т. д.) Zabbix использует преобразование по основанию 10 (1K = 1000).
- Для s (секунд):
  - Формат включает yyy mmm ddd hhh mmm sss ms; отображаются только до трёх самых крупных ненулевых единиц времени (например, 1M 10d 4h).
  - Если единица равна нулю и находится между двумя ненулевыми единицами, она опускается (например, 10d 56m вместо 10d 0h 56m).

При использовании *Units* к значениям элементов данных автоматически применяются следующие суффиксы множителей:

- **K**, **M**, **G**, **T** — кило, мега, гига, тера
- **P**, **E**, **Z**, **Y** — пета, экса, зетта, йотта *(применяются только в веб-интерфейсе)*

Чтобы предотвратить преобразование единиц, используйте префикс ! (например, !B или !s).

Следующие примеры показывают, как полученные значения элементов данных преобразуются в зависимости от указанных единиц:

```
1 B → 1 B
1024 B → 1 KB
1536 B → 1.5 KB
881764 B → 881.76 KB
881764 !B → 881764 B

0.0000155 s → 0.016ms
3470400 s → 1M 10d 4h
2606400 s → 1M 4h
2592000 s → 1M
```

```
2592001 s → 1M
2592001 !s → 2592001 s

17764 uptime → 04:56:04
86400 uptime → 1 day, 00:00:00
881764 uptime → 10 days, 04:56:04
32417764 uptime → 375 days, 04:56:04
32417764 !uptime → 32417764 uptime

881764 unixtime → 1970-01-11 04:56:04 AM

17764 Hz → 17.76 KHz
86400 Hz → 86.4 KHz
881764 Hz → 881.76 KHz
32417764 Hz → 32.42 MHz

0 ! → 0
0 !! → 0 !
```

Note:
До Zabbix 4.0 существовал жёстко заданный список исключений единиц измерения, состоящий из `ms`, `rpm`, `RPM`, `%`. Этот список исключений устарел, поэтому правильный способ предотвратить преобразование таких единиц — `!ms`, `!rpm`, `!RPM`, `!%`.

9 Выполнение команд

Zabbix использует общую функциональность для внешних проверок, пользовательских параметров, элементов данных system.run, пользовательских скриптов оповещений, удалённых команд и глобальных скриптов.

Шаги выполнения

Note:
По умолчанию все скрипты в Zabbix выполняются с использованием оболочки *sh*, и изменить оболочку по умолчанию невозможно. Чтобы использовать другую оболочку, можно применить обходной путь: создать файл скрипта и вызывать этот скрипт при выполнении команды.

Команда/скрипт выполняется одинаковым образом как на платформах Unix, так и Windows:

1. Zabbix (родительский процесс) создает канал для обмена данными
2. Zabbix назначает этот канал в качестве вывода для создаваемого дочернего процесса
3. Zabbix создает дочерний процесс (запускает команду/скрипт)
4. Для дочернего процесса создается новая группа процессов (в Unix) или задание (в Windows)
5. Zabbix читает из канала до тех пор, пока не произойдет тайм-аут или пока никто не будет записывать в другой конец канала (не будут закрыты ВСЕ дескрипторы handles/file descriptors). Обратите внимание, что дочерний процесс может создать дополнительные процессы и завершиться до того, как они завершатся или закроют handle/file descriptor.
6. Если тайм-аут не был достигнут, Zabbix ожидает, пока исходный дочерний процесс не завершится или пока не произойдет тайм-аут
7. Если исходный дочерний процесс завершился и тайм-аут не был достигнут, Zabbix проверяет код завершения исходного дочернего процесса и сравнивает его с 0 (ненулевое значение считается ошибкой выполнения, только для пользовательских скриптов оповещений, удаленных команд и пользовательских скриптов, выполняемых на сервере Zabbix и прокси Zabbix)
8. На этом этапе предполагается, что все завершено, и все дерево процессов (то есть группа процессов или задание) завершается

Attention:
Zabbix считает, что команда/скрипт завершила обработку, когда исходный дочерний процесс завершился И ни один другой процесс больше не удерживает открытым дескриптор вывода handle/file descriptor. Когда обработка завершена, завершаются ВСЕ созданные процессы.

Все двойные кавычки и обратные слеши в команде экранируются обратными слешами, а сама команда заключается в двойные кавычки.

Проверка кода завершения

Коды завершения проверяются при следующих условиях:

- Только для пользовательских скриптов оповещений, удаленных команд и пользовательских скриптов, выполняемых на сервере Zabbix и прокси Zabbix.
- Любой код завершения, отличный от 0, считается ошибкой выполнения.
- Содержимое стандартного потока ошибок и стандартного потока вывода для неудачных выполнений собирается и доступно в веб-интерфейсе (где отображается результат выполнения).
- Дополнительная запись в журнале может быть создана для удаленных команд, выполняемых на Zabbix агенте/прокси, если включить параметр LogRemoteCommands в agent/proxy.

Возможные сообщения в веб-интерфейсе и записи в журнале для команд/скриптов, завершившихся с ошибкой:

- Содержимое стандартного потока ошибок и стандартного потока вывода для неудачных выполнений (если есть).
- "Процесс завершился с кодом: N." (для пустого вывода и кода завершения, не равного 0).
- "Процесс был завершен сигналом: N." (для процесса, завершенного сигналом, только в Linux).
- "Процесс завершился неожиданно." (для процесса, завершенного по неизвестным причинам).

См. также

- Внешние проверки
- Пользовательские параметры
- Элементы данных system.run
- Пользовательские скрипты оповещений
- Удалённые команды
- Глобальные скрипты

9 Настройка периодов времени

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 времени 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`

10 Совместимость версий

Поддерживаемые агенты

Для совместимости с Zabbix 8.0 Zabbix агент должен быть не старее версии 1.4 и не новее 8.0.

Вам может потребоваться проверить конфигурацию старых агентов, так как некоторые параметры изменились, например параметры, связанные с журналированием для версий до 3.0.

Чтобы в полной мере воспользоваться новейшей функциональностью, метриками, повышенной производительностью и сниженным использованием памяти, используйте последнюю поддерживаемую версию агента.

Примечания для Windows XP

- В 32-разрядной Windows XP не используйте Zabbix агенты новее 6.0.x;
- В Windows XP/Server 2003 не используйте шаблоны агента новее Zabbix 4.0.x. Более новые шаблоны используют английские счётчики производительности, которые поддерживаются только начиная с Windows Vista/Server 2008.

Поддерживаемые агенты 2

Более ранние версии Zabbix агент 2, начиная с версии 4.4, совместимы с Zabbix 8.0; версия Zabbix агент 2 не должна быть новее 8.0.

Обратите внимание, что при использовании Zabbix агент 2 версий 4.4 и 5.0 по умолчанию используется интервал 10 минут для обновления неподдерживаемых элементов данных.

Чтобы в полной мере воспользоваться новейшей функциональностью, метриками, повышенной производительностью и сниженным использованием памяти, используйте последнюю поддерживаемую версию агент 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, например новые элементы данных.

По отношению к серверу Zabbix, прокси могут быть:

- *Текущими* (прокси и сервер имеют одинаковую основную версию);
- *Устаревшими* (версия прокси старше версии сервера, но частично поддерживается);
- *Неподдерживаемыми* (версия прокси старше версии предыдущего LTS-релиза сервера *или* версия прокси новее основной версии сервера).

Примеры:

Версия сервера	Версия *текущего* прокси	Версия *устаревшего* прокси	Версия *неподдерживаемого* прокси
6.4	6.4	6.0, 6.2	Старше 6.0; новее 6.4
7.0	7.0	6.0, 6.2, 6.4	Старше 6.0; новее 7.0
7.2	7.2	7.0	Старше 7.0; новее 7.2
7.4	7.4	7.0	Старше 7.0; новее 7.4
8.0	8.0	7.0, 7.2, 7.4	Старше 7.0; новее 8.0

Функциональность, поддерживаемая прокси:

Версия прокси	Обновление данных	Обновление конфигурации	Задачи
Текущая	Да	Да	Да
Устаревшая	Да	Нет	Удаленные команды (например, shell-скрипты); Немедленные проверки значений элементов данных (то есть *Выполнить сейчас*); Примечание: Тесты предобработки с реальным значением не поддерживаются.
Неподдерживаемая	Нет	Нет	Нет

Оповещения об использовании несовместимых версий демона Zabbix будут журналироваться.

Поддерживаемые XML-файлы

XML-файлы версии не старше 1.8 поддерживаются для импорта в Zabbix 8.0.

Attention:
В формате экспорта XML зависимости триггеров сохраняются только по имени. Если существует несколько триггеров с одинаковым именем (например, с разными уровнями важности и выражениями), между которыми определена зависимость, импортировать их невозможно. Такие зависимости необходимо вручную удалить из XML-файла и повторно добавить после импорта.

12 Динамически подключаемая библиотека Zabbix sender для Windows

Обзор

В среде Windows приложения могут отправлять данные на Zabbix сервер/прокси, используя динамически подключаемую библиотеку Zabbix sender (zabbix_sender.dll), вместо запуска внешнего процесса (zabbix_sender.exe).

Для компиляции пользовательских приложений с zabbix_sender.dll требуются файлы zabbix_sender.h и zabbix_sender.lib.

Получение

Существует 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 агент (не Zabbix агент 2).

Файлы zabbix_sender.h, zabbix_sender.lib и zabbix_sender.dll будут находиться внутри загруженного ZIP-архива в каталоге `bin\dev`. Распакуйте файлы в нужное место.

2. Соберите zabbix_sender.dll из исходного кода (см. инструкции).

Динамически подключаемая библиотека вместе с файлами для разработки будет находиться в каталоге `bin\winXX\dev`. Для ее использования подключите заголовочный файл zabbix_sender.h и выполните компоновку с библиотекой zabbix_sender.lib.

См. также

- пример простой утилиты Zabbix sender, реализованной с использованием динамически подключаемой библиотеки Zabbix sender, иллюстрирующий использование библиотеки;
- файл zabbix_sender.h с интерфейсными функциями динамически подключаемой библиотеки Zabbix sender. Этот файл содержит документацию, объясняющую назначение каждой интерфейсной функции, ее аргументы и возвращаемое значение.

13 Обновление мониторинга сервисов

Обзор В Zabbix 6.0 функциональность мониторинга сервисов была значительно переработана (см. Что нового в Zabbix 6.0.0 со списком изменений).

На этой странице описано, как сервисы и SLA, определённые в более ранних версиях Zabbix, изменяются при обновлении до Zabbix 6.0 или новее.

Сервисы В более старых версиях Zabbix у сервисов было два типа зависимостей: мягкие и жёсткие. После обновления все зависимости станут равнозначными.

Если сервис «Дочерний сервис» ранее был связан с «Родительским сервисом 1» через жёсткую зависимость, а дополнительно с «Родительским сервисом 2» через мягкую зависимость, то после обновления у «Дочернего сервиса» будет два родительских сервиса: «Родительский сервис 1» и «Родительский сервис 2».

Сопоставление проблем и сервисов на основе триггеров было заменено сопоставлением на основе тегов. В Zabbix 6.0 и новее форма настройки сервиса содержит новый параметр *Теги проблем*, который позволяет указать одну или несколько пар имени и значения тега для сопоставления проблем. Триггеры, которые были связаны с сервисом, получат новый тег

ServiceLink : <trigger ID>:<trigger name> (значение тега будет обрезано до 32 символов). Связанные сервисы получат тег проблемы ServiceLink 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 сервис имеет проблему → Наиболее критичный из дочерних сервисов
- Проблема, если все дочерние сервисы имеют проблемы → Наиболее критичный, если все дочерние сервисы имеют проблемы

SLA Ранее целевые показатели SLA нужно было определять отдельно для каждого сервиса. Начиная с Zabbix 6.0, SLA стал отдельной сущностью, которая содержит информацию о расписании сервиса, ожидаемом целевом уровне сервиса (SLO) и периодах простоя, которые следует исключить из расчета. После настройки SLA можно назначить нескольким сервисам с помощью тегов 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 этот же тег будет указан в параметре *Теги сервиса* соответствующего SLA.
- Время создания сервиса, новая метрика в отчетах SLA, будет установлено на 01/01/2000 00:00 для существующих сервисов.

14 Другие проблемы

Вход в систему и systemd

Мы рекомендуем создать пользователя *zabbix* как 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, где работает Zabbix. Это может привести к сбою демона Zabbix при выходе из системы. В этом случае в журнале сервера Zabbix вы увидите 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. Когда вы выходите из системы, семафоры, ранее созданные Zabbix, удаляются, что и вызывает сбой.

Цитата из документации systemd:

```
RemoveIPC=

       ,            IPC System V   POSIX,                ,
                      .                   .             ,
                          IPC                            .
               System V,                          ,       POSIX
                      .           ,         IPC         root
                                              .             — "yes".
```

Существует 2 решения этой проблемы:

1. (рекомендуется) Не использовать учетную запись *zabbix* ни для чего, кроме процессов Zabbix; создать отдельную учетную запись для других задач.
2. (не рекомендуется) Установить RemoveIPC=no в /etc/systemd/logind.conf и перезагрузить систему. Обратите внимание, что RemoveIPC — это общесистемный параметр, и его изменение повлияет на всю систему.

Использование веб-интерфейса Zabbix за прокси

Если веб-интерфейс Zabbix работает за прокси-сервером, путь cookie в файле конфигурации прокси необходимо переписать так, чтобы он соответствовал пути, используемому при обратном проксировании. См. примеры ниже. Если

путь cookie не будет переписан, пользователи могут столкнуться с проблемами авторизации при попытке входа 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
# ..
```

16 Примеры экранирования

Обзор

На этой странице приведены примеры правильного экранирования при использовании регулярных выражений в различных контекстах.

> **Note:**
> При использовании конструктора выражений триггеров 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]+"}`

Обратите внимание:

- обратные слеши не экранируются;
- кавычки экранируются.

Функция макроса внутри параметра ключа элемента данных

Регулярное выражение: `.+:(\d+)$<br>` Ключ элемента данных: `net.tcp.service[tcp,,"{{$ENDPOINT}.regsub(\".+:(\d+)$"`

Обратите внимание:

- регулярное выражение внутри функции макроса `regsub` заключено в двойные кавычки (поскольку содержит закрывающую круглую скобку);
- кавычки вокруг регулярного выражения экранируются (поскольку весь третий параметр элемента данных заключен в двойные кавычки);
- третий параметр ключа элемента данных заключен в двойные кавычки, поскольку содержит запятую.

Функция LLD-макроса

Регулярное выражение: `\.+\"([a-z]+)<br>` LLD-макрос: `{{#MACRO}.iregsub("\.+\\"([a-z]+)", \1)}`

Обратите внимание:

- обратные слеши не экранируются;
- кавычки экранируются.

Функция LLD-макроса внутри контекста пользовательского макроса

Регулярное выражение: `\.+\"([a-z]+)<br>` LLD-макрос: `{{#MACRO}.iregsub("\.+\\"([a-z]+)", \1)}<br>` Пользовательский макрос с контекстом: `{$MACRO:"{{#MACRO}.iregsub(\"\.+\\\"[a-z]+\", \1)}"}`

Обратите внимание:

- экранирование обратных слешей для LLD не меняется;
- при вставке LLD-макроса в контекст пользовательского макроса его необходимо поместить в строку:

1. вокруг выражения макроса добавляются кавычки;
2. кавычки экранируются; в сумме добавляются 3 новых обратных слеша.

Строковый параметр функции (любой)

В качестве примера используется `concat`.

Содержимое строки: `\.+\"[a-z]+<br>` Выражение: `concat("abc", "\\.\\\"[a-z]+")`

Обратите внимание:

- строковые параметры требуют экранирования как обратных слешей, так и кавычек.

Функция LLD-макроса внутри строкового параметра функции

Регулярное выражение: `\.+\"([a-z]+)<br>` LLD-макрос: `{{#MACRO}.iregsub("\.+\\"([a-z]+)", \1)}<br>` Выражение: `concat("abc, "{{#MACRO}.iregsub(\"\\.+\\\\\"([a-z]+)\", \\1)}")`

Обратите внимание:

- строковые параметры требуют экранирования как обратных слешей, так и кавычек;
- добавляется еще один уровень экранирования, поскольку макрос будет разрешен только после снятия кавычек со строки;

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]+\"}")`

Обратите внимание:

- как и в предыдущем примере, требуется дополнительный уровень экранирования;
- обратные слеши и кавычки экранируются только для верхнего уровня экранирования (поскольку это строковый параметр).

Функция LLD-макроса внутри контекста пользовательского макроса внутри функции

Регулярное выражение: `\.+\"([a-z])+<br>` LLD-макрос: `{{#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)}\"}")`

Обратите внимание на три уровня экранирования:

1. Для функции LLD-макроса, без экранирования обратных слешей;
2. Для пользовательского макроса с контекстом, без экранирования обратных слешей;
3. Для строкового параметра функции, с экранированием обратных слешей.

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]+\"}"`

Обратите внимание:

- строки также требуют экранирования обратных слешей;
- строки также требуют экранирования кавычек;
- снова случай с 2 уровнями экранирования:

1. Экранирование для контекста пользовательского макроса без экранирования обратных слешей;
2. Экранирование из-за того, что это строка, с экранированием обратных слешей.

21 Краткие справочные руководства

Обзор

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

Он разработан с учётом потребностей новых пользователей Zabbix и может использоваться как навигатор по другим разделам документации, содержащим информацию, необходимую для решения задачи.

Доступны следующие краткие справочные руководства:

- Мониторинг Linux с помощью агента Zabbix
- Мониторинг Windows с помощью агента Zabbix
- Мониторинг Apache по HTTP
- Мониторинг MySQL с помощью Zabbix agent 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 браузера
- Мониторинг сертификатов веб-сайта с помощью Zabbix agent 2 (пассивный)
- Мониторинг сетевого коммутатора или маршрутизатора с помощью Zabbix
- Мониторинг журнала событий Windows с использованием активных проверок

1 Мониторинг Linux с помощью Zabbix агента

Введение На этой странице вы узнаете о шагах, необходимых для запуска базового мониторинга машин Linux с помощью Zabbix. Шаги, описанные в этом руководстве, можно применить к любой операционной системе на базе Linux.

Для кого это руководство

Это руководство предназначено для новых пользователей Zabbix и содержит минимальный набор шагов, необходимых для включения базового мониторинга вашей машины Linux. Если вы ищете возможности глубокой настройки или вам требуется более продвинутая конфигурация, см. раздел Конфигурация руководства Zabbix.

Предварительные требования

Прежде чем продолжить работу с этим руководством, 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 отслеживать. В зависимости от ваших потребностей мониторинга это может быть та же машина, на которой установлен сервер Zabbix, или совершенно другая машина.

Выберите наиболее подходящий способ установки:

- Запуск в виде Docker-контейнера — см. список доступных образов в репозитории Zabbix Docker.
- Установка из пакетов Zabbix (доступно для Alma Linux, CentOS, Debian, Oracle Linux, Raspberry Pi OS, RHEL, Rocky Linux, SUSE Linux Enterprise Server, Ubuntu).
- Компиляция из исходных кодов.

Настройка Zabbix для мониторинга Zabbix агент может собирать метрики в активном или пассивном режиме (одновременно).

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

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

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

Пассивные 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 при необходимости порт) вашего сервера Zabbix в параметр server. Например:

```
Server=192.0.2.0:10051
```

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

3. Перезапустите агент Zabbix.

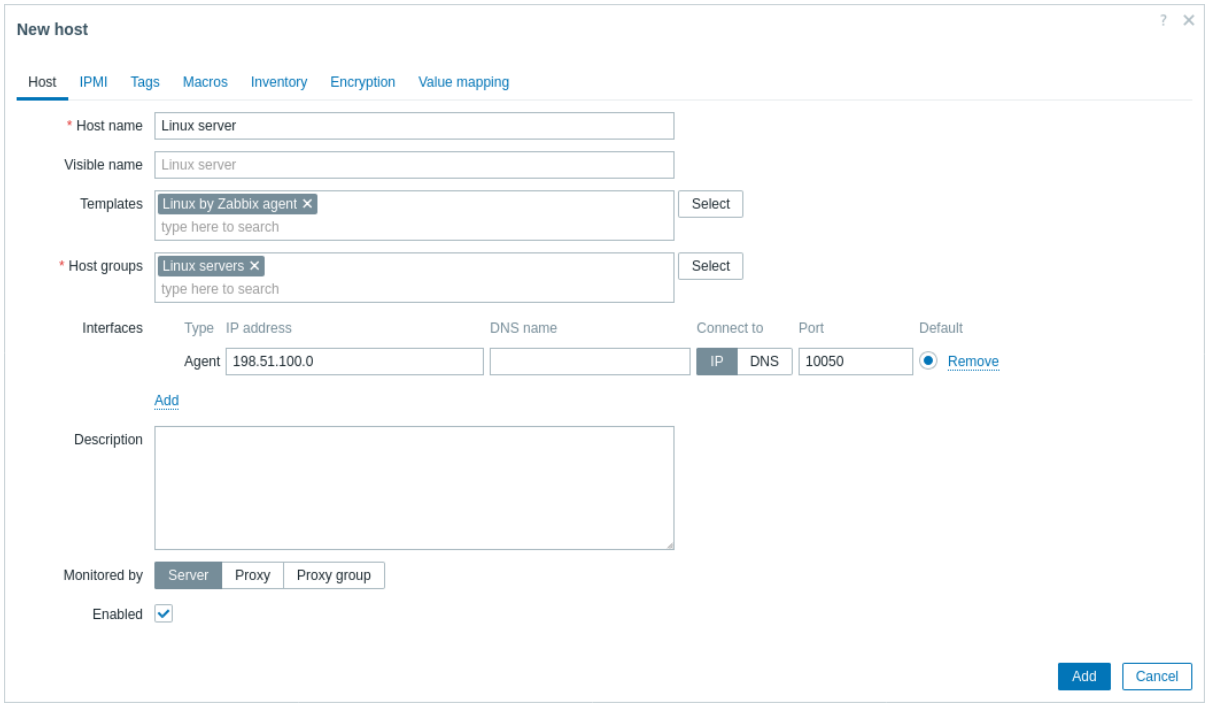
```
systemctl restart zabbix-agent
```

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 привязан к узлу 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 Linux-машины, на которой установлен агент.



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

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=192.0.2.0:10051
```

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

3. Укажите 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, потому что для активных проверок агент Zabbix использует его для получения правильной конфигурации узла сети с сервера. В частности, агент инициирует соединение с сервером 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.

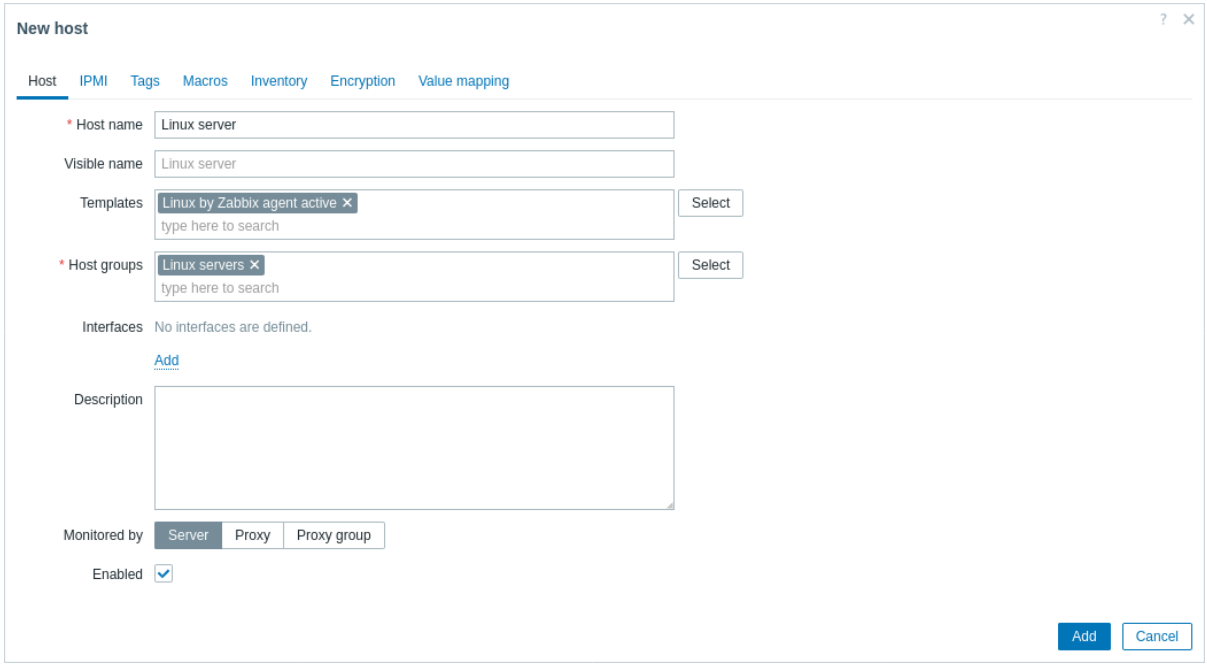
```
systemctl restart zabbix-agent
```

Веб-интерфейс Zabbix

1. Войдите в веб-интерфейс Zabbix.

2. Создайте узел сети в веб-интерфейсе Zabbix.

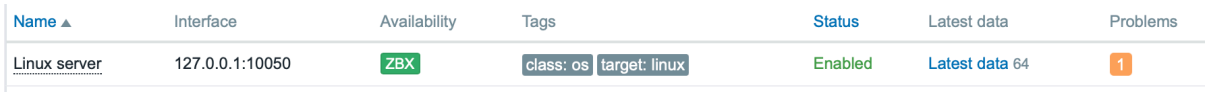
- В поле *Host name* введите имя узла сети (например, ”Linux server”), которое соответствует значению параметра *Hostname*, ранее заданному в файле конфигурации агента.
- В поле *Templates* введите или выберите шаблон ”Linux by Zabbix agent active”, который будет привязан к узлу сети.
- В поле *Host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



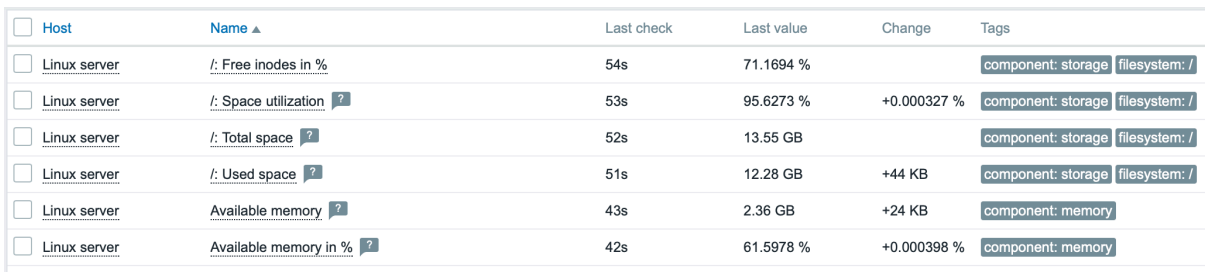
3. Нажмите *Add*, чтобы добавить узел сети. Этот узел сети будет представлять контролируемую Linux-машину.

Просмотр собранных метрик Поздравляем! На этом этапе Zabbix уже отслеживает вашу Linux-машину.

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

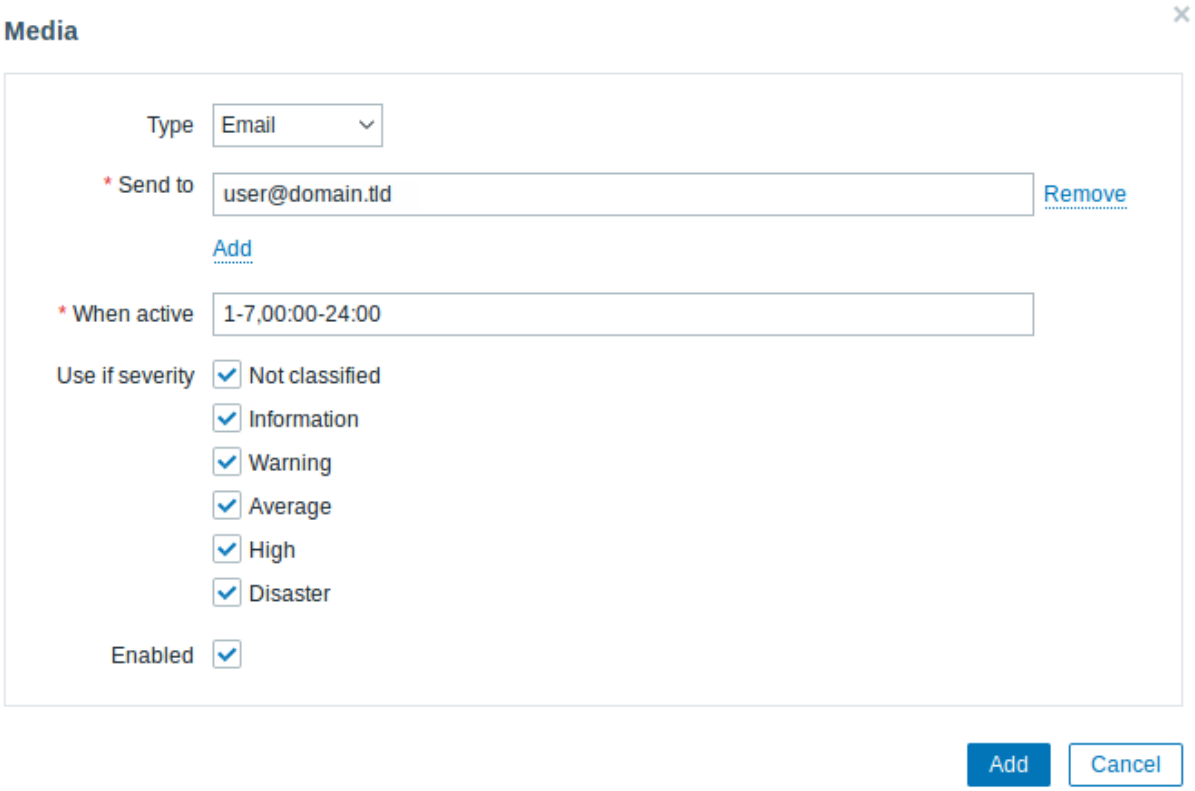


В результате откроется список всех последних метрик, собранных с узла сети Linux-сервера.



Настройка оповещений о проблемах Zabbix может уведомлять вас о проблеме в вашей инфраструктуре различными способами. В этом руководстве приведены шаги по настройке отправки оповещений по электронной почте.

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



2. Следуйте руководству Получение уведомлений о проблемах.

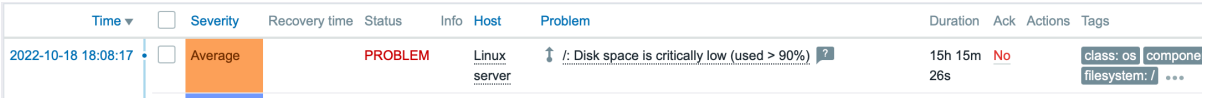
В следующий раз, когда Zabbix обнаружит проблему, вы должны получить оповещение по электронной почте.

Проверьте вашу конфигурацию В Linux вы можете сымитировать высокую нагрузку на CPU и в результате получить оповещение о проблеме, выполнив:

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

Вам может потребоваться запустить несколько процессов md5sum, чтобы нагрузка на CPU превысила пороговое значение.

Когда Zabbix обнаружит проблему, она появится в разделе Мониторинг->Проблемы.



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 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 будет устранена в течение 45 минут, отправить сообщение менеджеру центра обработки данных).
- Установка из пакетов - как установить компоненты Zabbix с использованием официальных пакетов RPM и DEB для различных дистрибутивов Linux, обеспечивая доступ к новейшим возможностям и исправлениям ошибок.

2 Monitor Windows with Zabbix agent

Введение На этой странице описаны шаги, необходимые для начала базового мониторинга машин Windows с помощью Zabbix.

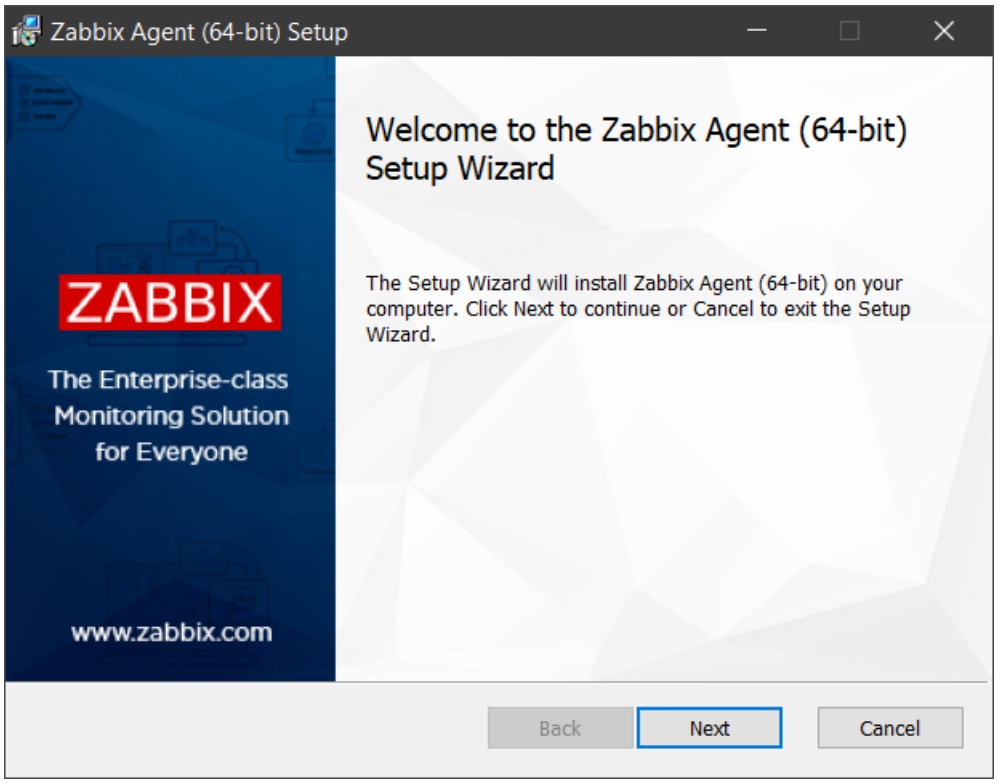
Для кого предназначено это руководство

Это руководство предназначено для новых пользователей Zabbix и содержит минимальный набор шагов, необходимых для включения базового мониторинга вашей машины Windows.
Если вам нужны широкие возможности настройки или требуется более сложная конфигурация, см. раздел Configuration руководства Zabbix.

Предварительные требования

Прежде чем продолжить работу с этим руководством по установке, 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. Его необходимо установить на компьютер под управлением Windows, который вы хотите контролировать. Следуйте инструкциям по установке агента Zabbix для Windows.



Настройка Zabbix для мониторинга Zabbix агент может собирать метрики в активном или пассивном режиме (одновременно).

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

Шаблоны мониторинга, предоставляемые Zabbix, обычно предлагают два варианта — шаблон для Zabbix агента и шаблон для Zabbix агента (active). В первом случае агент будет собирать метрики в пассивном режиме. Такие шаблоны обеспечивают идентичные результаты мониторинга, но с использованием разных протоколов связи.

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

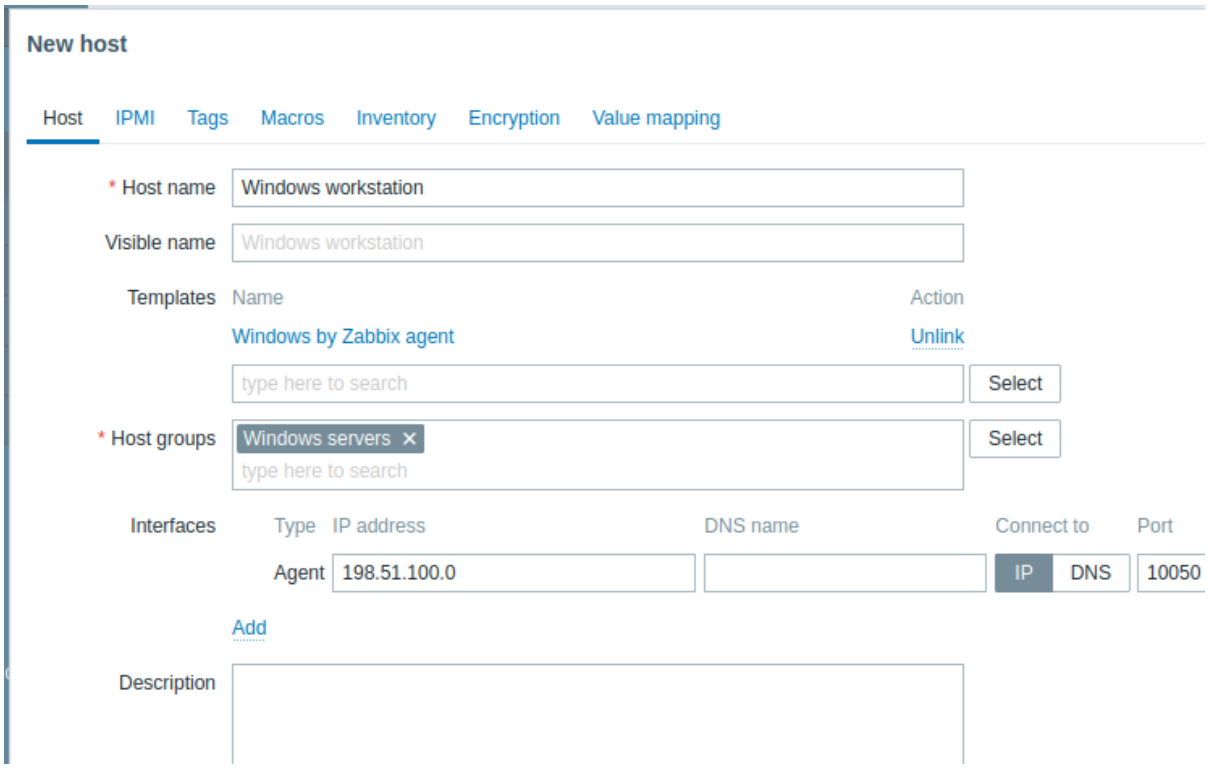
Пассивные проверки Zabbix frontend

1. Log into Zabbix frontend.
2. Create a host in Zabbix web interface.

This host will represent your Windows machine.

3. In the *Interfaces* parameter, add *Agent* interface and specify the IP address or DNS name of the Windows machine where the agent is installed.

4. In the *Templates* parameter, type or select *Windows by Zabbix agent*.



Zabbix agent

For passive checks Zabbix agent needs to know the IP address or DNS name of Zabbix server. If you have provided correct information during the agent installation, the configuration file is already updated. Otherwise, you need to manually specify it. Go to the `C:\Program files\Zabbix Agent` folder, open the file *zabbix_agentd.conf* and add the IP/DNS of your Zabbix server to the *Server* parameter.

Example:

```
Server=192.0.2.22
```

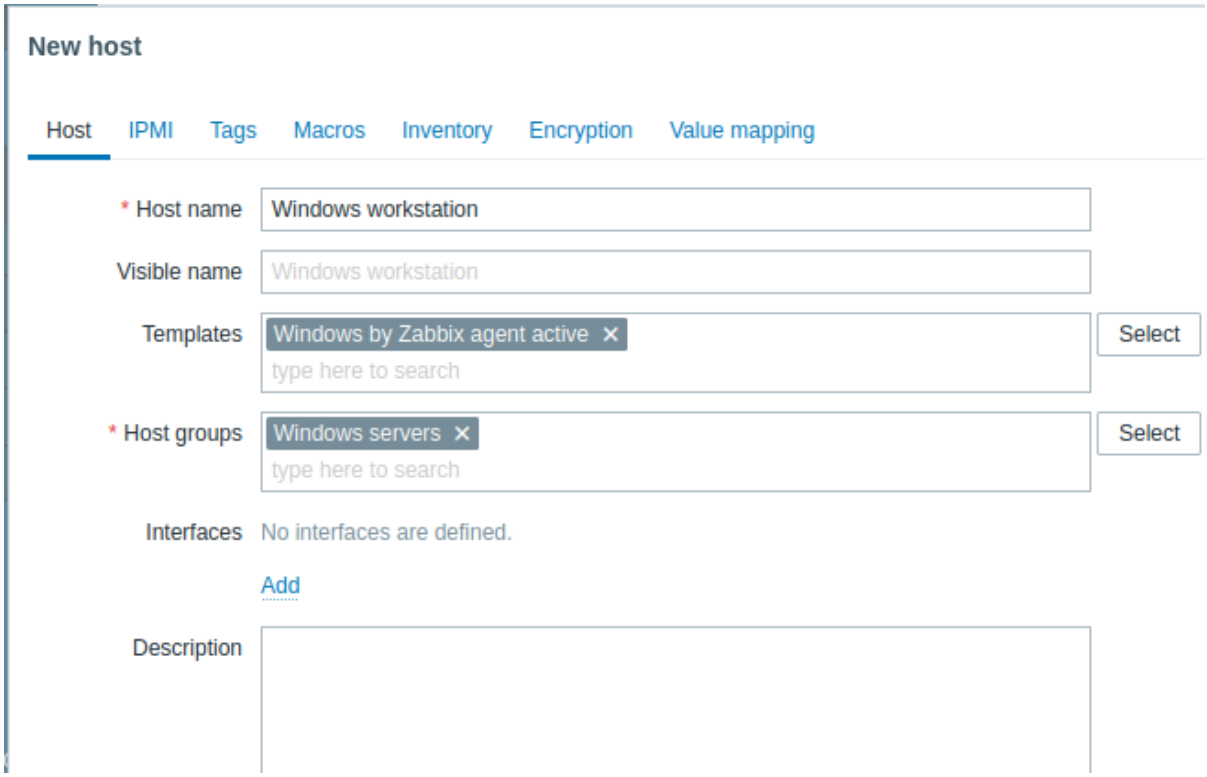
Активные проверки Веб-интерфейс Zabbix

1. Войдите в веб-интерфейс Zabbix.

2. Создайте узел сети в веб-интерфейсе Zabbix.

Этот узел сети будет представлять вашу машину Windows.

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



Zabbix agent

In the `C:\Program files\Zabbix Agent` folder open the file *zabbix_agentd.conf* and add:

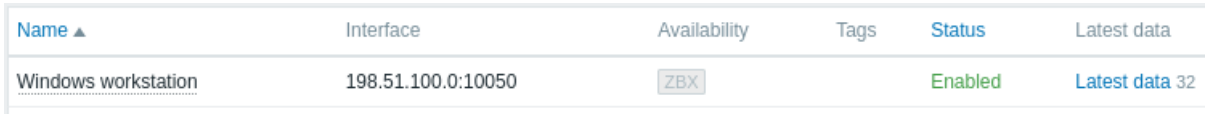
- The name of the host you created in Zabbix web interface to the *Hostname* parameter.
- The IP address or DNS name of your Zabbix server machine to the *ServerActive* parameter (might be prefilled if you have provided it during Zabbix agent setup).

Example:

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

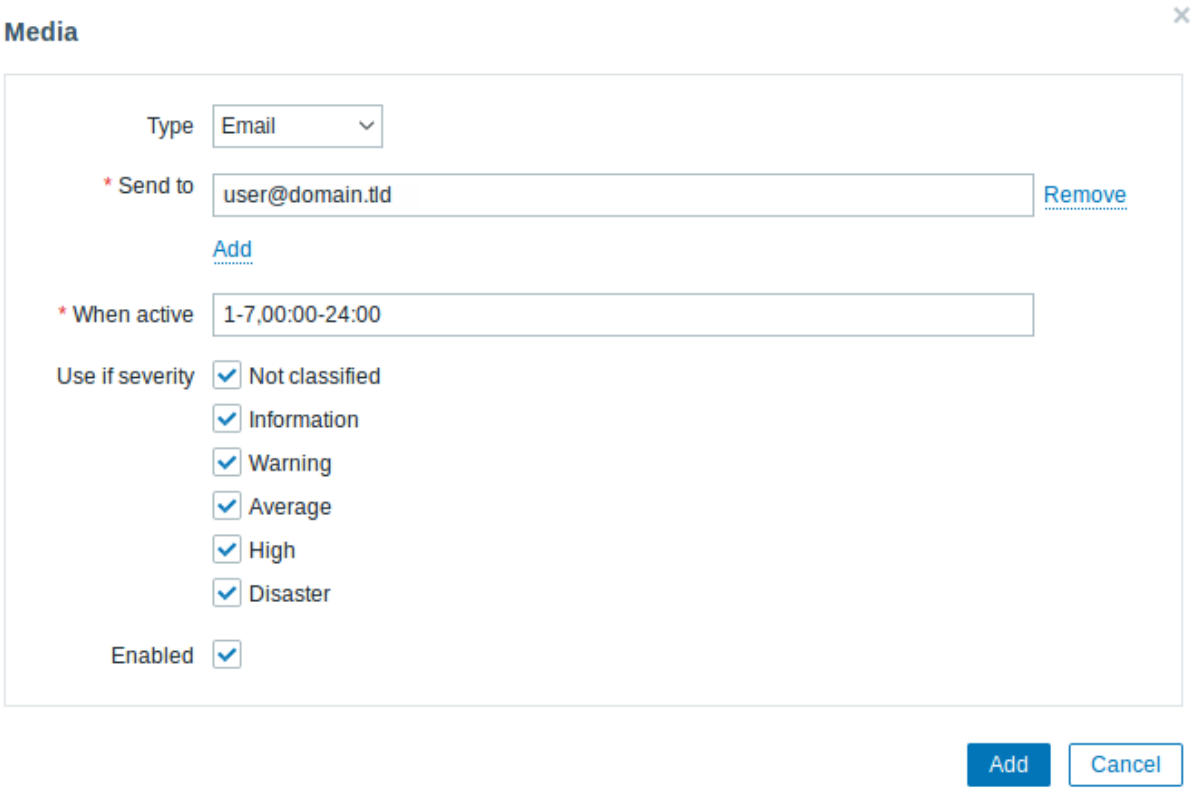
Просмотр собранных метрик Поздравляем! На этом этапе Zabbix уже выполняет мониторинг вашей машины Windows.

Чтобы просмотреть собранные метрики, откройте раздел меню *Мониторинг->Узлы сети* и нажмите *Последние данные* рядом с узлом сети.



Настройка оповещений о проблемах Zabbix может уведомлять вас о проблеме в вашей инфраструктуре различными способами. В этом руководстве приведены 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 Получение уведомления о проблеме.

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

Note:
В Windows вы можете использовать утилиту CpuStres для имитации высокой загрузки CPU и, как результат, 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 будет устранена в течение 45 минут, отправить сообщение менеджеру центра обработки данных).

3 Мониторинг Apache по 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

Прежде чем продолжить работу с этим руководством 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 вы не видите status_module в списке, 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
```

В ответе должна содержаться статистика веб-сервера Apache.

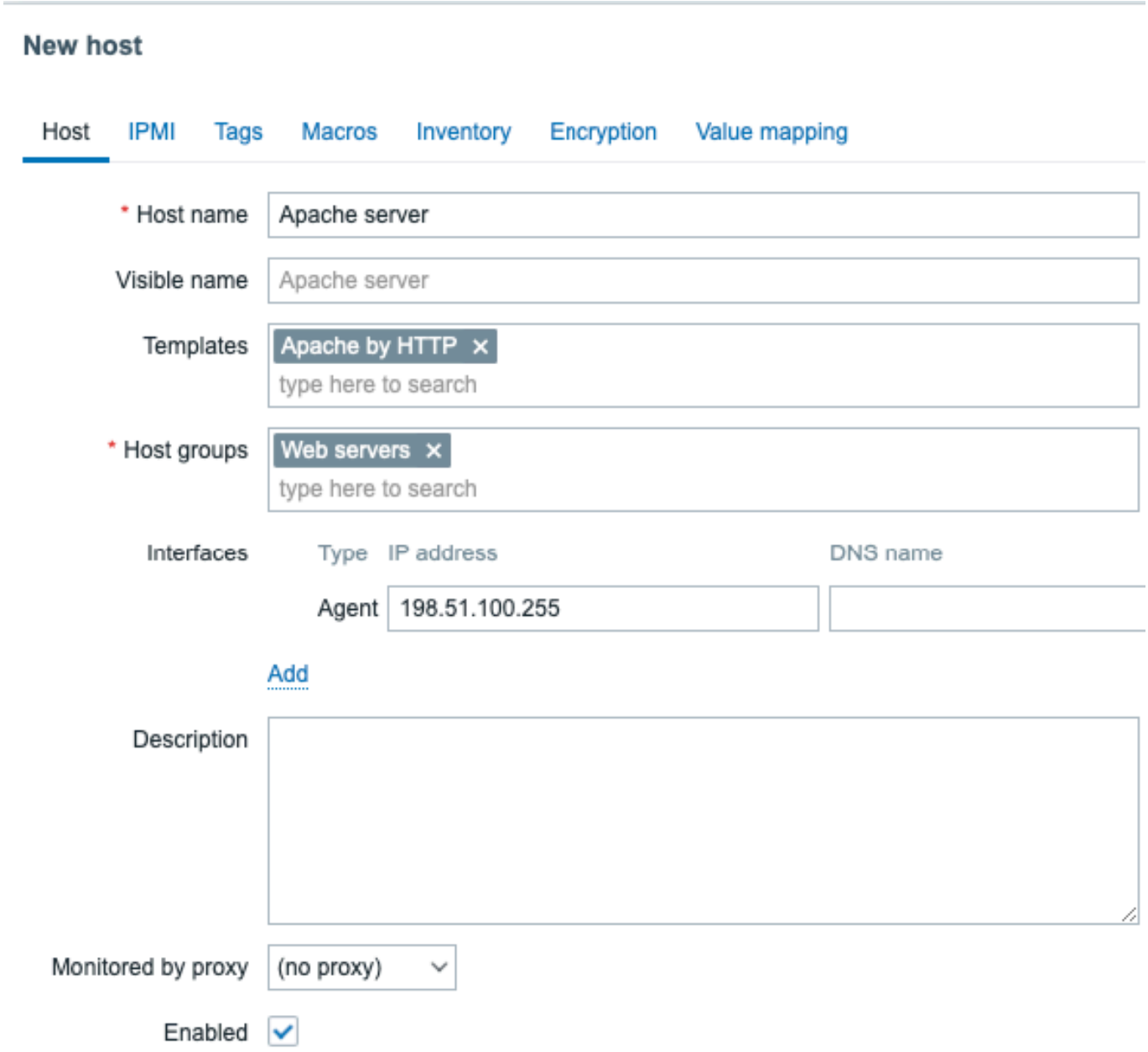
Настройка Zabbix для мониторинга 1. Войдите в веб-интерфейс Zabbix.

2. Создайте узел сети в веб-интерфейсе Zabbix.

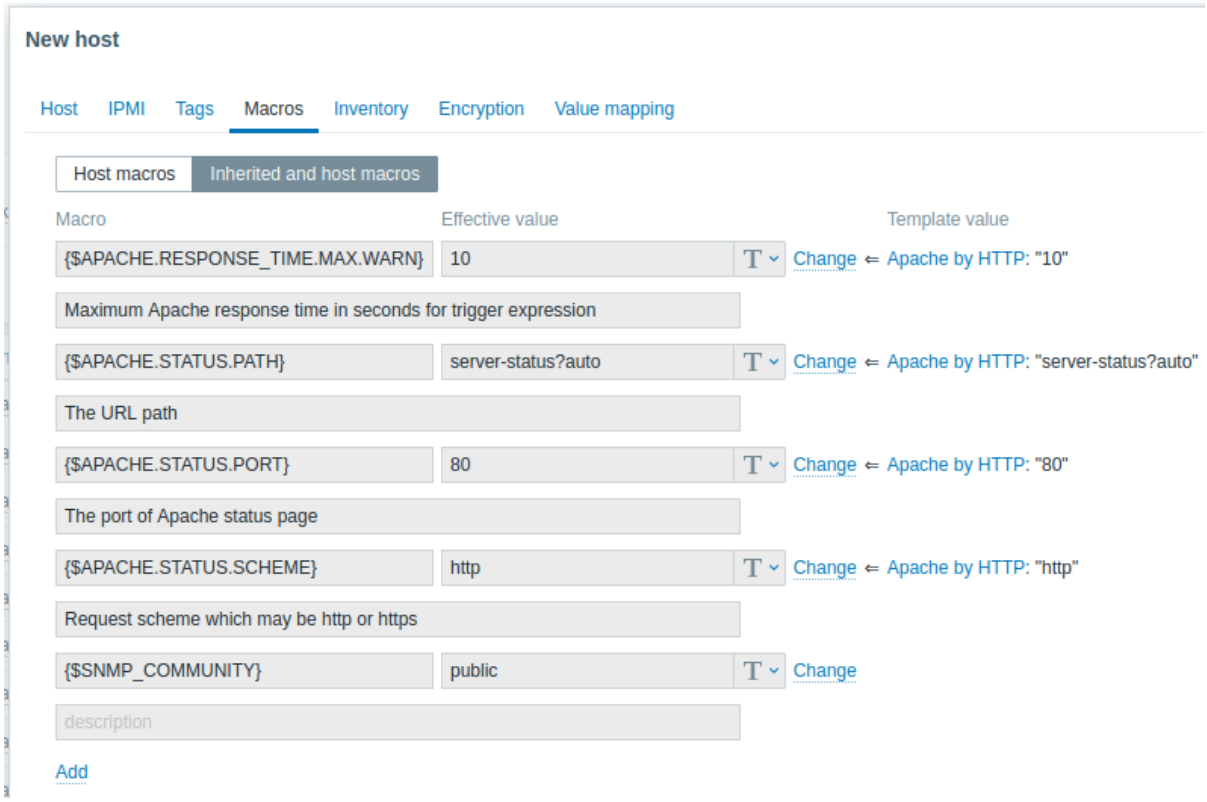
Этот узел сети будет представлять ваш сервер Apache.

3. В параметре *Interfaces* добавьте интерфейс *Агент* и укажите IP-адрес вашего экземпляра Apache. **Вам не нужно устанавливать агент Zabbix на эту машину**, 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*. Убедитесь, что значения макросов {$APACHE.STATUS.PORT} и {$APACHE.STATUS.SCHEME} соответствуют настройкам вашей установки. По умолчанию используется порт 80 и схема http. Измените значения макросов, если вы используете другой порт и/или схему.

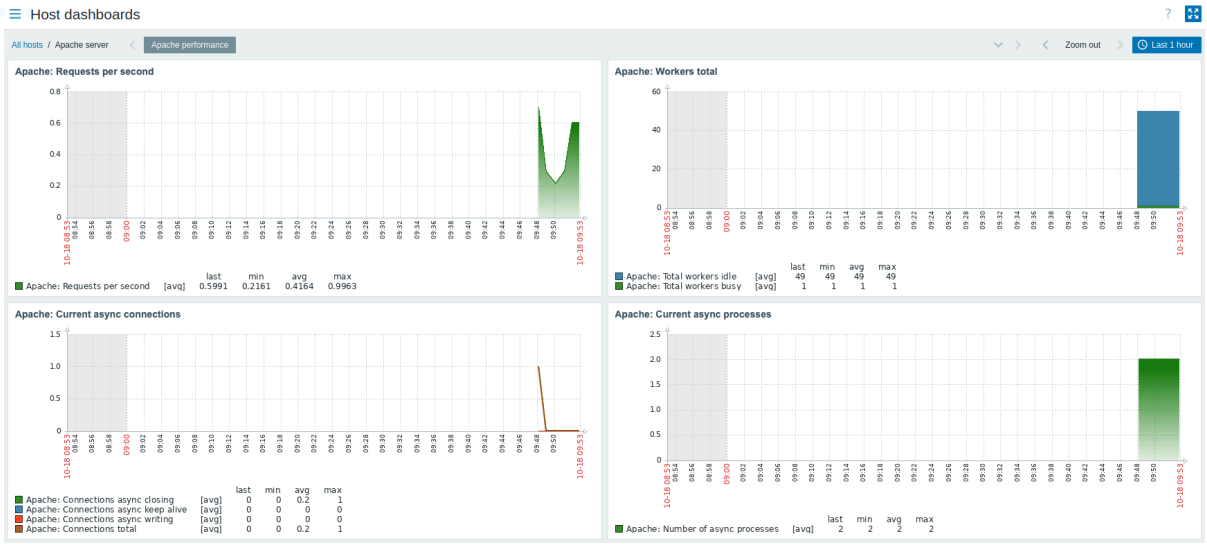


Просмотр собранных метрик Поздравляем! На этом этапе Zabbix уже выполняет мониторинг вашего веб-сервера Apache.

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



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



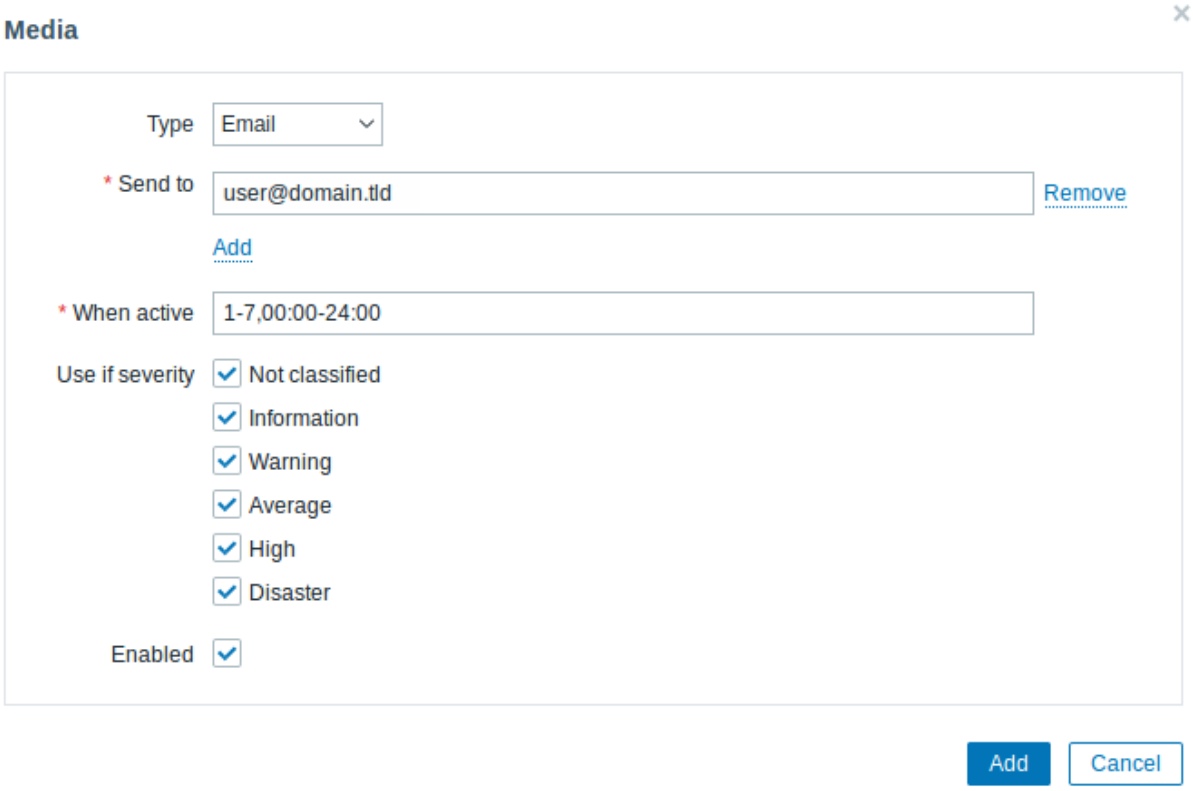
Кроме того, в разделе *Monitoring->Hosts* вы можете нажать *Latest data*, чтобы просмотреть все последние собранные метрики в виде списка.



Host	Name	Last check	Last value	Change	Tags	Info
Apache server	Apache: Bytes per request	32s	5.93 KB	+921.92 B	component: connection	Graph
Apache server	Apache: Bytes per second	32s	2.56 KBps	+1.57 KBps	component: network	Graph
Apache server	Apache: Connections async closing	32s	0	-1	component: connection	Graph
Apache server	Apache: Connections async keep alive	32s	0		component: connection	Graph
Apache server	Apache: Connections async writing	32s	0		component: connection	Graph
Apache server	Apache: Connections total	32s	0	-1	component: connection	Graph
Apache server	Apache: Get status	32s	{"Date":"Tue, 18 Oct 2022 ...		component: raw	History
Apache server	Apache: Number of async processes	32s	2		component: system	Graph
Apache server	Apache: Requests per second	32s	0.283	-0.7133	component: network	Graph

Настройка оповещений о проблемах Zabbix может уведомлять вас о проблемах в вашей инфраструктуре различными способами. В этом руководстве приведены 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 Получение 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:

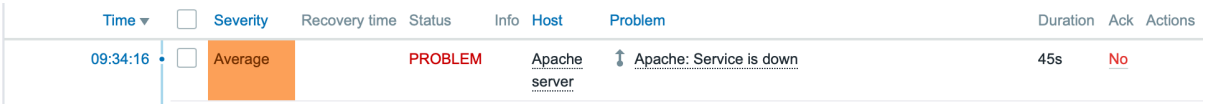
1. Откройте конфигурацию узла сети *Apache server* в Zabbix.

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

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

4. Нажмите Update, чтобы сохранить конфигурацию узла сети.

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



Time	Severity	Recovery time	Status	Info	Host	Problem	Duration	Ack	Actions
09:34:16	Average		PROBLEM		Apache server	Apache: Service is down	45s	No	

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.

- HTTP-элементы данных — как отслеживать пользовательские метрики с помощью HTTP-агента.
- Эскалации проблем — как создавать многошаговые сценарии оповещений
  (например, сначала отправить сообщение системному администратору, затем, если проблема не будет решена в течение 45 минут, отправить сообщение
  менеджеру центра обработки данных).

4 Мониторинг MySQL с помощью Zabbix агент 2

Введение

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

Для мониторинга сервера MySQL существует несколько подходов: Zabbix агент, Zabbix агент 2 или стандарт Open Database Connectivity (ODBC). Основное внимание в этом руководстве уделено мониторингу сервера MySQL с помощью Zabbix агента 2, который является **рекомендуемым** подходом благодаря простой настройке в различных средах. 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 требуется более расширенная конфигурация, смотрите раздел Configuration руководства Zabbix.

Предварительные требования

Прежде чем продолжить работу с этим руководством, 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 немного отличаться. Это руководство основано на среде с 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”, созданный при установке Zabbix), однако у этого пользователя больше привилегий, чем требуется для простого мониторинга (привилегии на DROP баз данных, DELETE записей из таблиц и т. д.). Поэтому необходимо создать пользователя MySQL, предназначенного *только* для мониторинга сервера MySQL.

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

```
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;
```

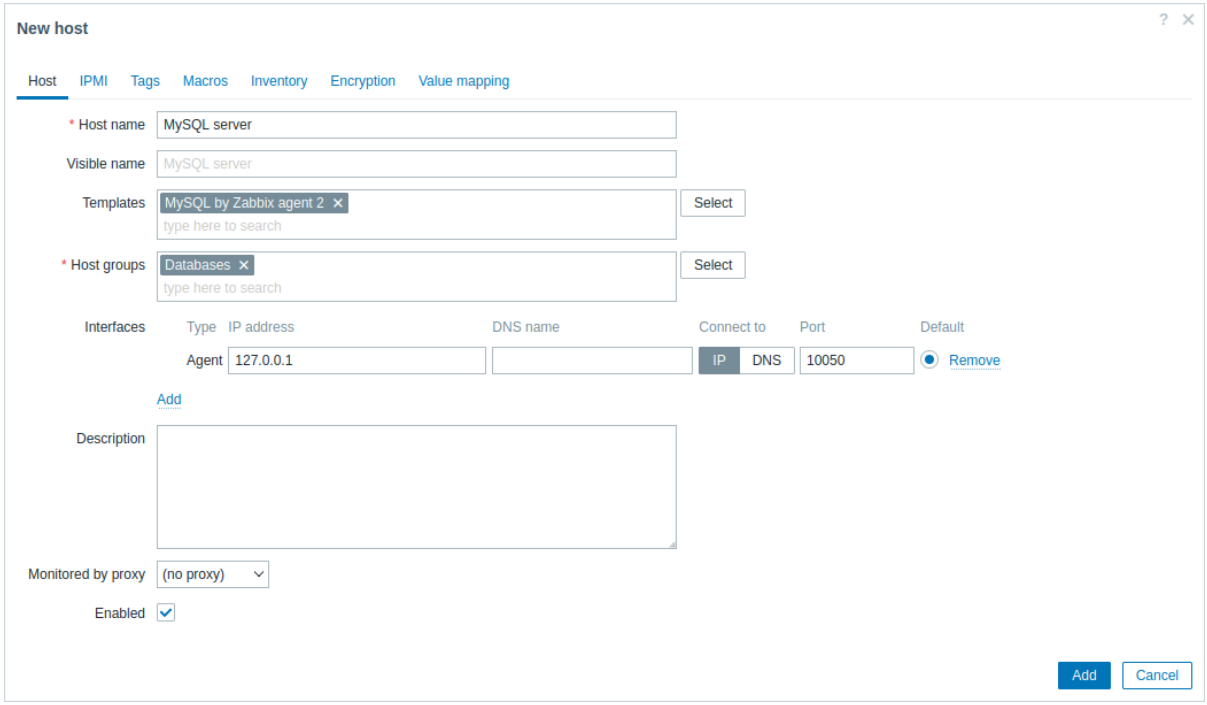
После создания пользователя можно переходить к следующему шагу.

Настройка веб-интерфейса Zabbix

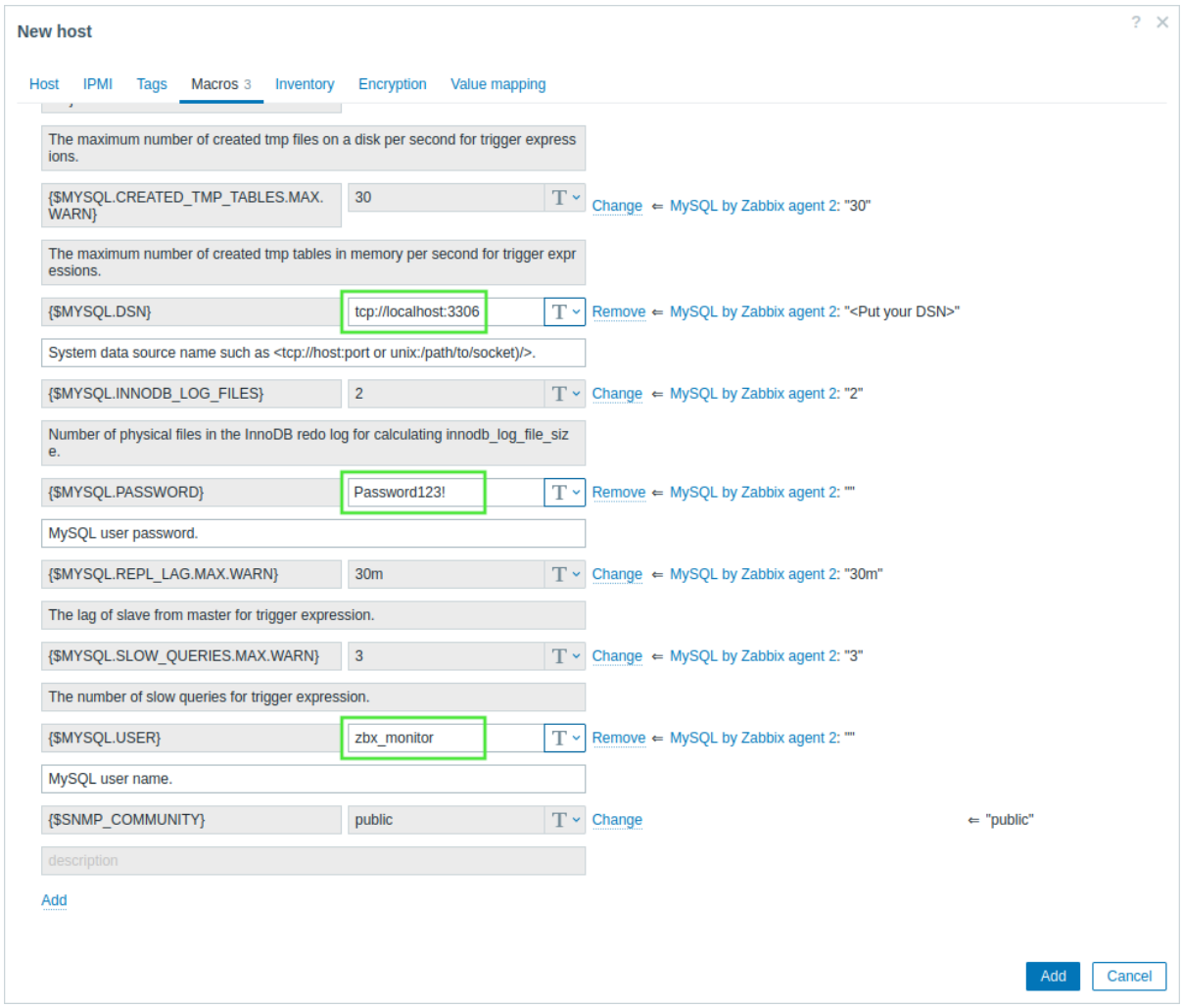
1. Войдите в веб-интерфейс Zabbix.

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 привязан к узлу 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 сервера MySQL. В этом руководстве используется ”127.0.0.1” (localhost) для мониторинга сервера MySQL, установленного на той же машине, что и сервер Zabbix и Zabbix агент 2.



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

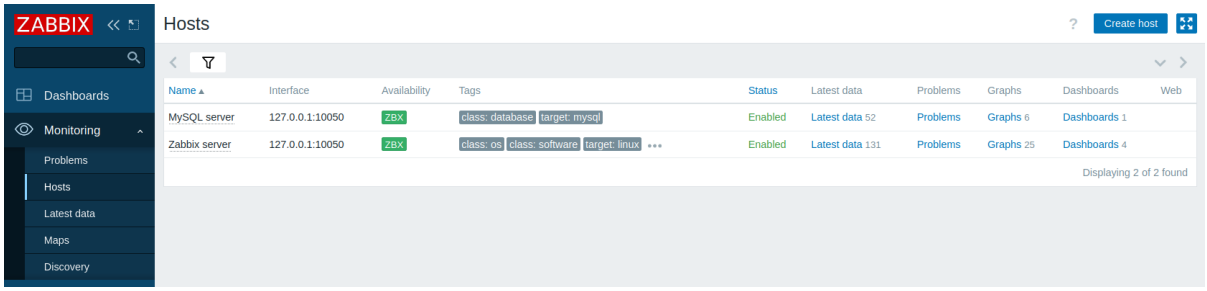


3. Нажмите *Add*, чтобы добавить узел сети. Этот узел сети будет представлять ваш сервер MySQL.

Просмотр собранных метрик

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

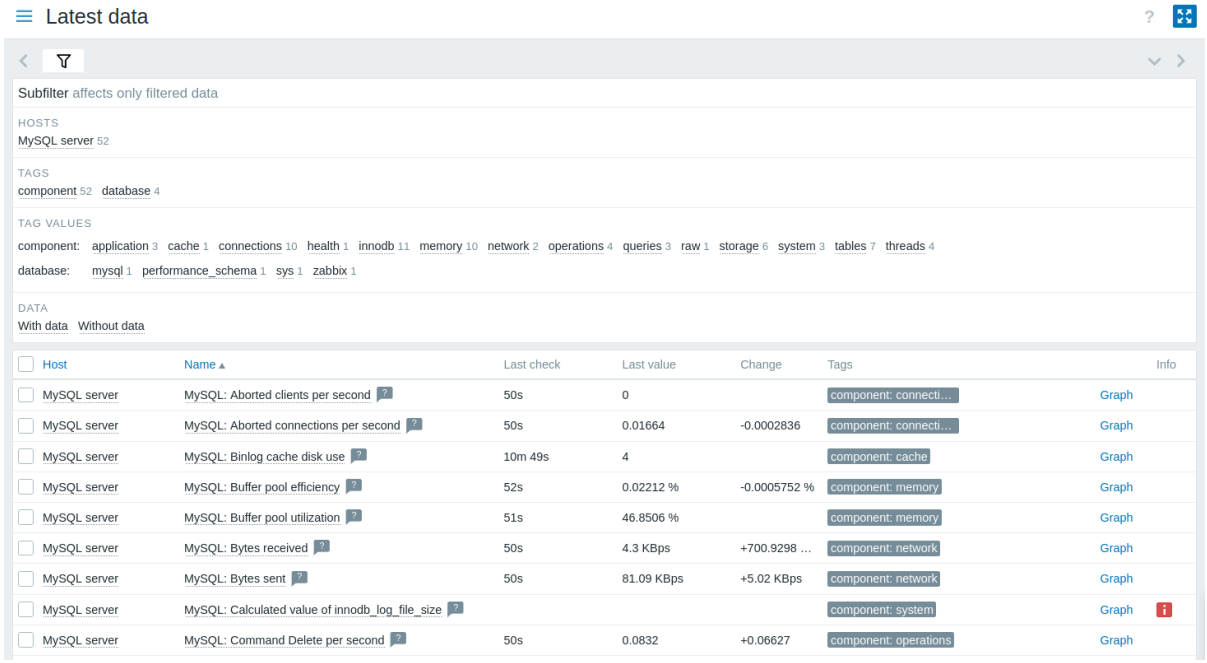
Чтобы просмотреть собранные метрики, перейдите в раздел меню *Мониторинг → Узлы сети* и нажмите *Панели* рядом с узлом сети.



Это действие перенаправит вас на панель узла сети (настроенную на уровне шаблона) с наиболее важными метриками, собранными с сервера MySQL.



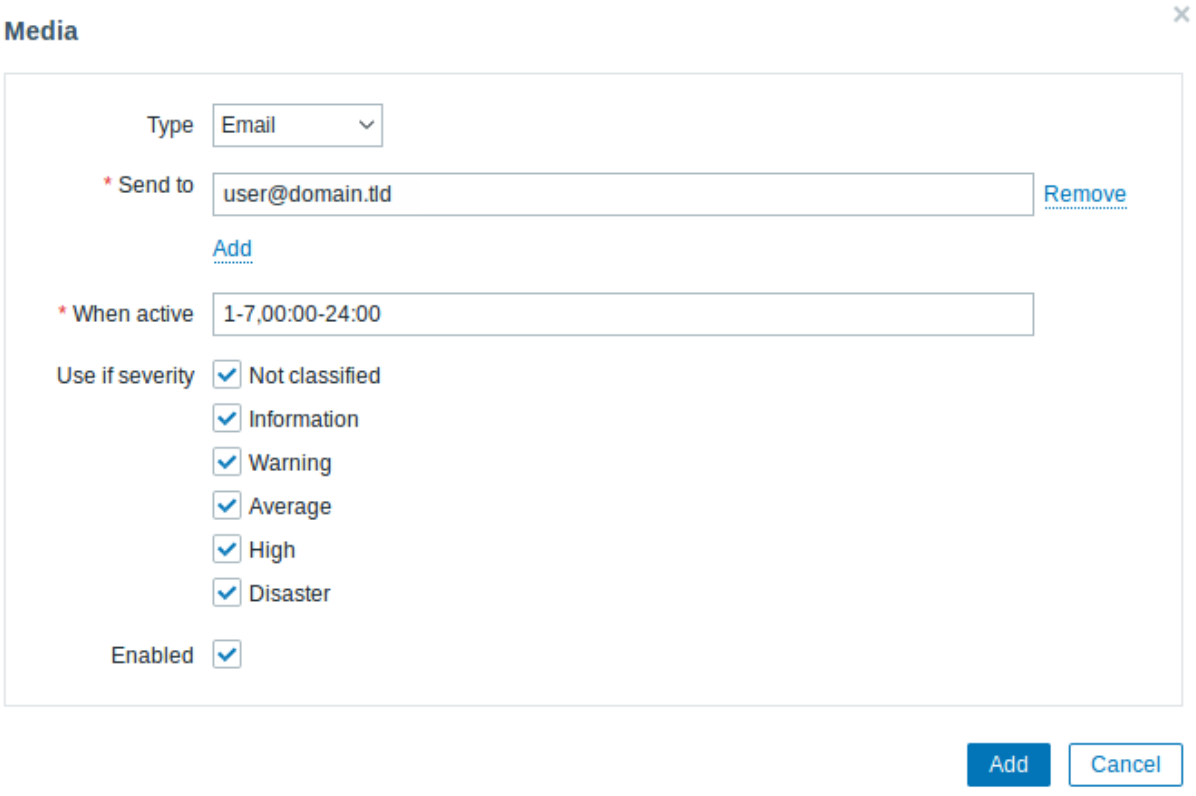
Кроме того, в разделе меню *Мониторинг → Узлы сети* вы можете нажать *Последние данные*, чтобы просмотреть все последние собранные метрики в виде списка. Обратите внимание, что для элемента данных *MySQL: Calculated value of innodb_log_file_size* отсутствие данных является ожидаемым, так как это значение будет вычислено на основе данных за последний час.



Настройка оповещений о проблемах

Zabbix может уведомлять вас о проблемах в вашей инфраструктуре различными способами. В этом руководстве приведены основные шаги по настройке отправки оповещений по электронной почте.

1. Перейдите в *Настройки пользователя → Профиль*, откройте вкладку *Способы оповещения* и добавьте свой адрес электронной почты.



2. Следуйте руководству Получение 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 сымитировать реальную проблему, обновив конфигурацию узла сети в веб-интерфейсе Zabbix.

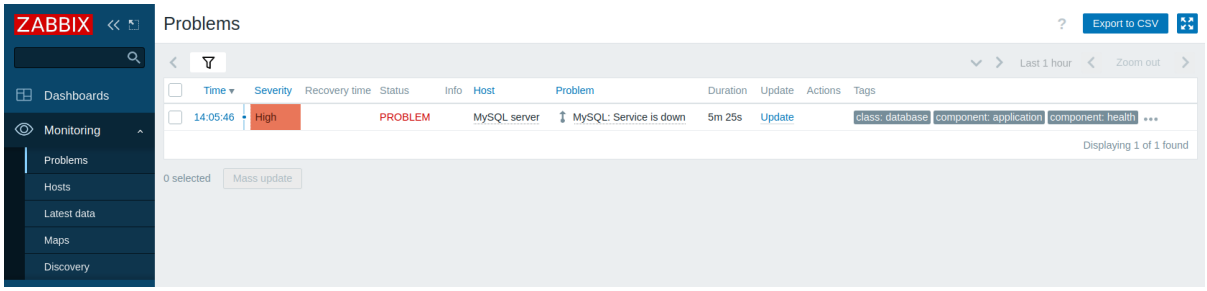
1. Откройте конфигурацию вашего узла сети MySQL server в Zabbix.

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

3. Нажмите *Change* рядом, например, с ранее настроенным значением макроса {$MYSQL.USER} и укажите другое имя пользователя MySQL.

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

5. Через некоторое время Zabbix обнаружит проблему «MySQL: Service is down», поскольку не сможет подключиться к серверу MySQL. Проблема появится в разделе *Monitoring → Problems*.



Если оповещения настроены, вы также получите уведомление о проблеме.

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

Другие подходы к мониторингу MySQL

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

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

Хотя многие шаги настройки аналогичны мониторингу сервера MySQL с помощью Zabbix агент 2, есть и существенные различия — вам необходимо настроить Zabbix агент или ODBC, чтобы иметь возможность мониторить сервер MySQL. Следующие инструкции помогут вам разобраться в этих **различиях**.

Мониторинг MySQL с помощью агента Zabbix

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

После успешной установки необходимых компонентов Zabbix необходимо 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 /usr/local/etc/zabbix/zabbix_agentd.d
```

> **Attention:**
> Каталог дополнительных конфигураций агента Zabbix должен находиться в том же каталоге, что и файл конфигурации агента Zabbix (*zabbix_agentd.conf*). В зависимости от вашей ОС и установки Zabbix этот каталог может находиться в другом месте, отличном от указанного в этом руководстве. Расположение по умолчанию смотрите в параметре `Include` в файле конфигурации агента 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 вами 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 получить доступ к серверу MySQL.

5. Перейдите в домашний каталог агента Zabbix (если он отсутствует в вашей системе, его необходимо 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 следующее 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

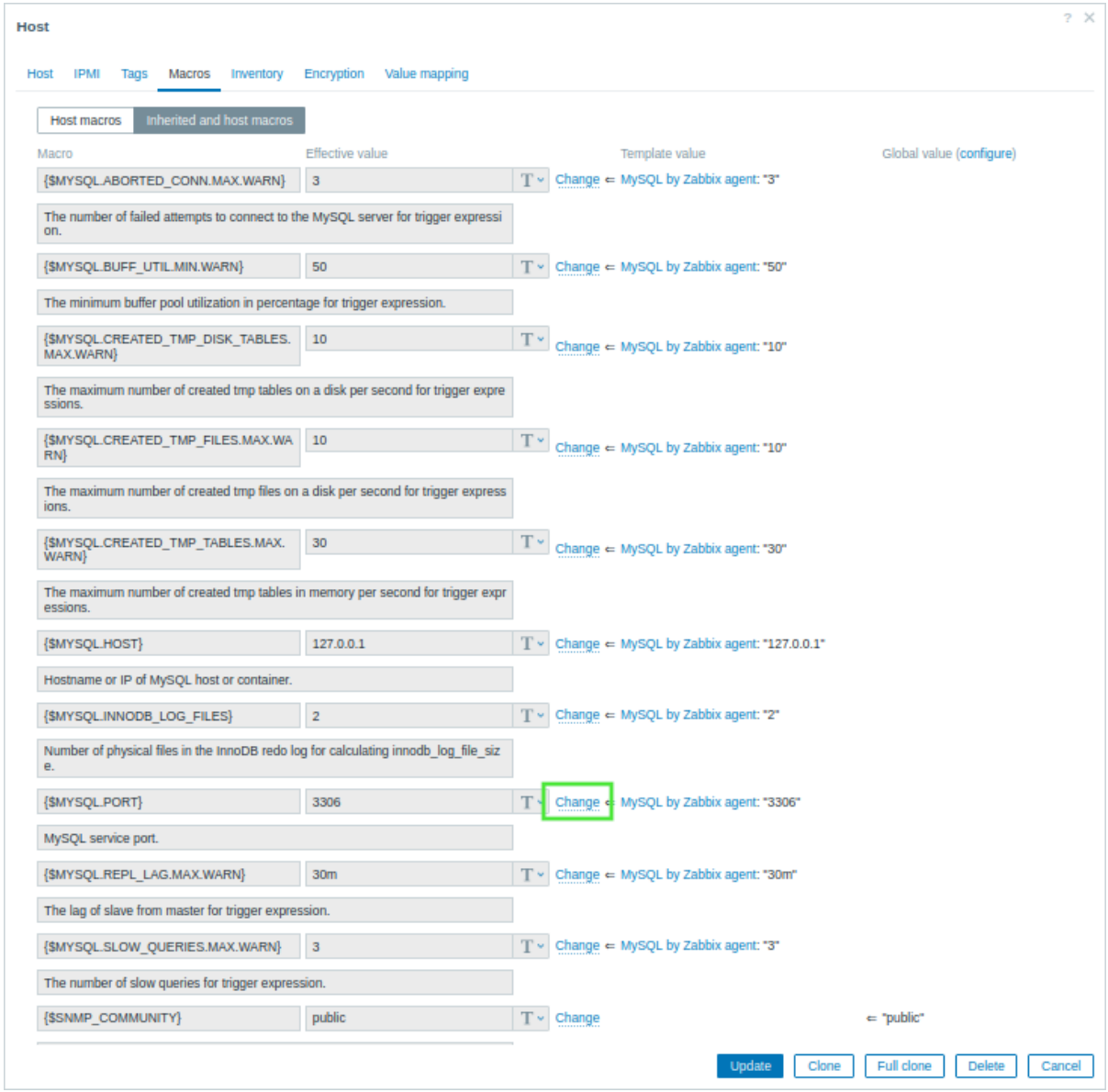
Чтобы настроить веб-интерфейс Zabbix, следуйте инструкциям в разделе *Настройка веб-интерфейса Zabbix* со следующими изменениями:

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

После настройки веб-интерфейса Zabbix вы можете просматривать собранные метрики и настроить оповещения о проблемах.

Чтобы проверить вашу конфигурацию, следуйте инструкциям в разделе *Проверка конфигурации* со следующими изменениями:

- В разделе *Inherited and host macros* конфигурации узла сети сервера MySQL нажмите *Change* рядом со значением макроса {$MYSQL.PORT} и укажите другой порт (например, ”6033”).



Мониторинг MySQL с помощью ODBC

Чтобы выполнять мониторинг сервера MySQL с помощью ODBC, необходимо загрузить и установить сервер Zabbix и веб-интерфейс Zabbix.

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

После создания пользователя MySQL необходимо настроить ODBC. Это включает установку одной из наиболее распространённых реализаций ODBC API 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. Несмотря на то, что у вас база данных MySQL, драйвер MariaDB unixODBC используется из соображений совместимости.

```
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 потребуется имя драйвера, которое указано 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 используется "127.0.0.1" (localhost) в качестве адреса сервера MySQL 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
```

Настройка веб-интерфейса Zabbix и проверка конфигурации

Чтобы настроить веб-интерфейс Zabbix, следуйте инструкциям из раздела *Настройка веб-интерфейса Zabbix* со следующими изменениями:

- В поле *Templates* введите или выберите шаблон "MySQL by ODBC", который будет привязан к узлу сети.
- Настройка *Interfaces* не требуется.
- Значение макроса {$MYSQL.DSN} в разделе *Inherited and host macros* конфигурации узла сети сервера MySQL должно быть установлено в имя DSN из файла *odbc.ini*.

После настройки веб-интерфейса Zabbix вы сможете просматривать собранные метрики, настроить оповещения о проблемах и проверить конфигурацию.

См. также

- Создание элемента данных — как начать мониторинг дополнительных метрик.
- Эскалации проблем — как создавать многошаговые сценарии оповещений (например, сначала отправить сообщение системному администратору, 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 базой 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*.

5 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 Настройка руководства Zabbix.

Предварительные требования

Прежде чем 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 немного отличаться. Это руководство основано на среде с Ubuntu.

Предполагается, что VMware уже настроен. В этом руководстве не рассматривается настройка VMware.

Настройка сервера Zabbix

Чтобы выполнять мониторинг VMware, необходимо включить процессы Zabbix *vmware collector*. Дополнительную информацию о том, как выполняется мониторинг VMware, смотрите в разделе Мониторинг виртуальных машин.

1. Откройте файл конфигурации сервера Zabbix.

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

2. Найдите и задайте параметр `StartVMwareCollectors` в файле конфигурации сервера Zabbix 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
```

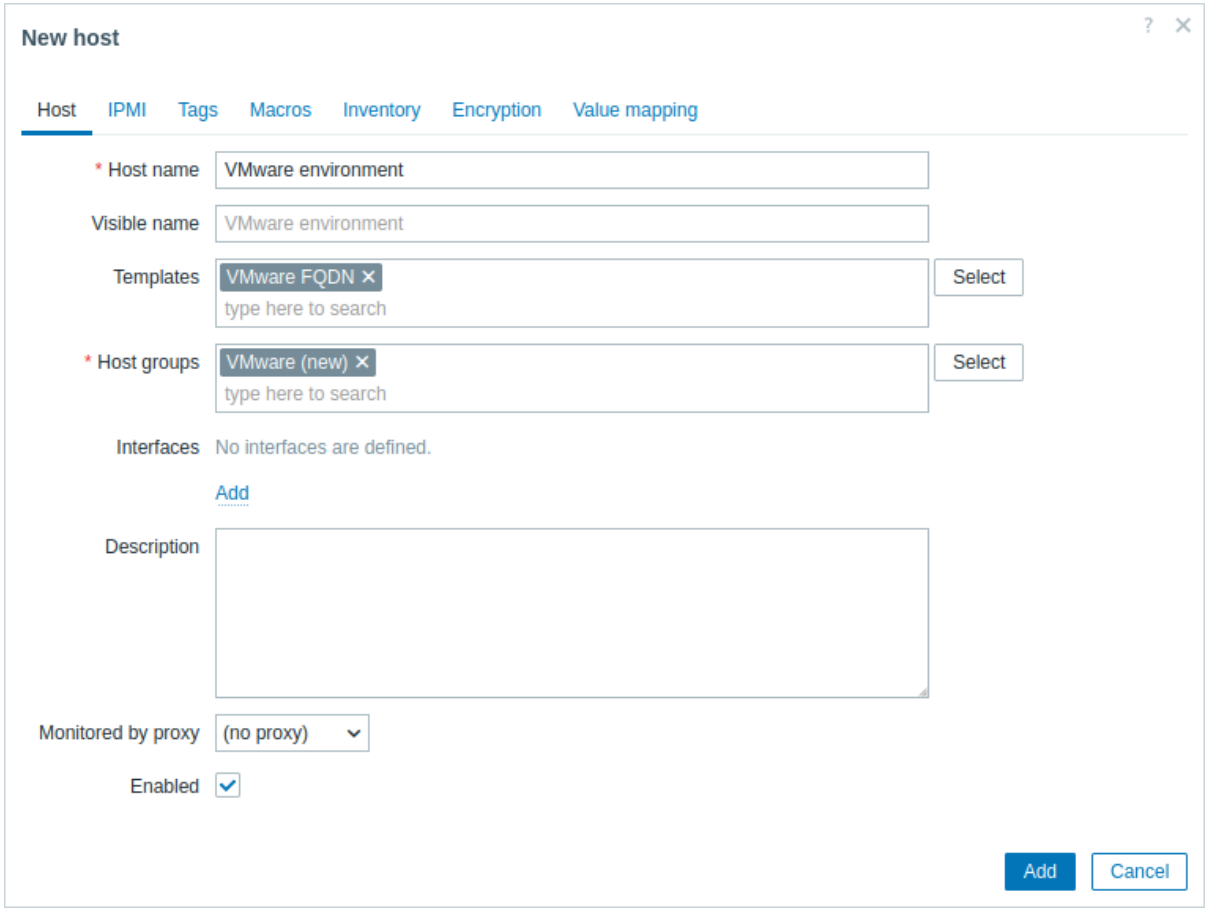
После запуска процессов *vmware collector* переходите к следующему шагу.

Настройка веб-интерфейса Zabbix

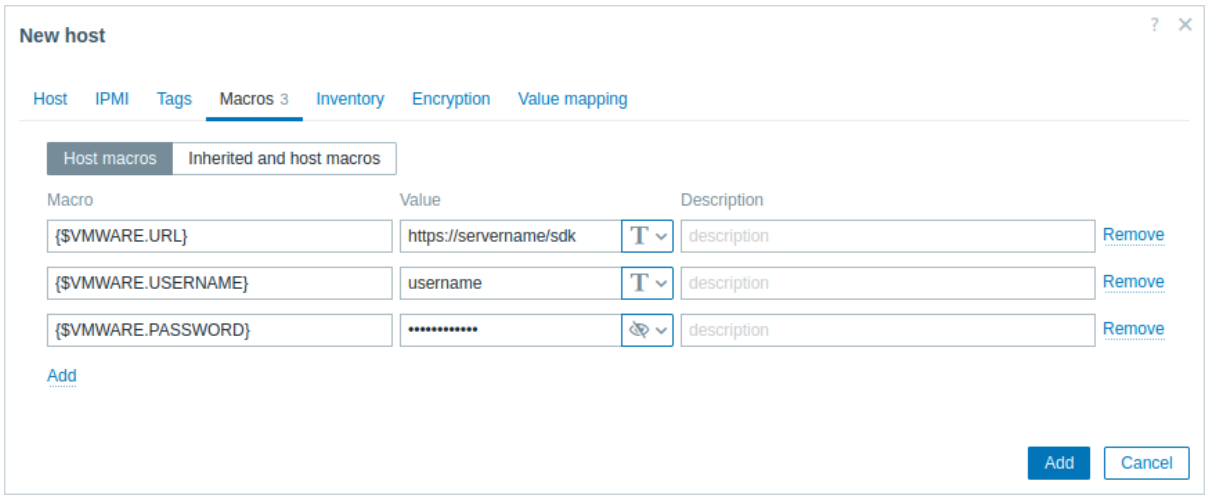
1. Войдите в веб-интерфейс Zabbix.

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”). Дополнительную информацию 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:
  - {$VMWARE.URL} - URL SDK службы VMware (vCenter или гипервизора ESXi) (https://servername/sdk)
  - {$VMWARE.USERNAME} - имя пользователя службы VMware
  - {$VMWARE.PASSWORD} - пароль пользователя {$VMWARE.USERNAME} службы VMware



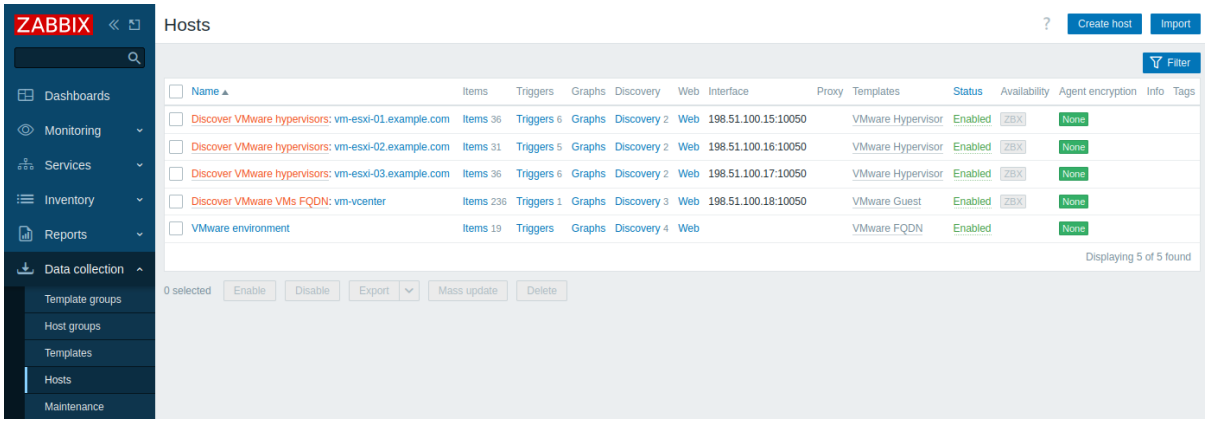
3. Нажмите кнопку *Add*, чтобы создать узел сети. Этот узел сети будет представлять вашу среду VMware.

Просмотр собранных метрик

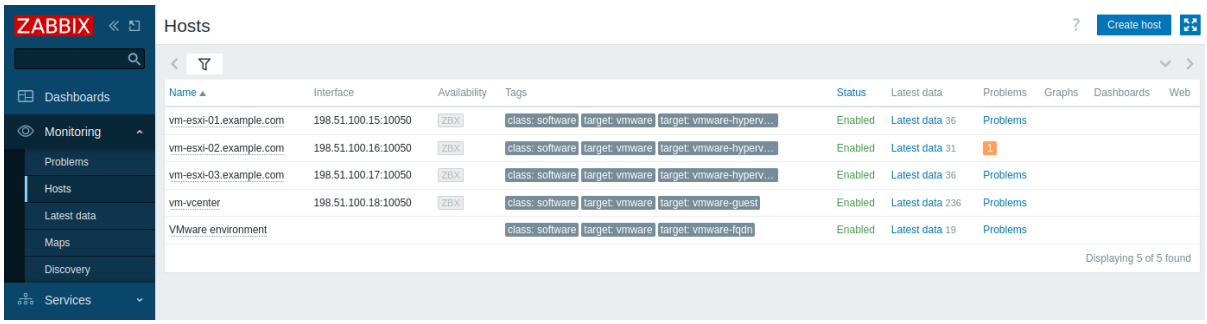
Поздравляем! На этом этапе Zabbix уже выполняет мониторинг вашей среды VMware.

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

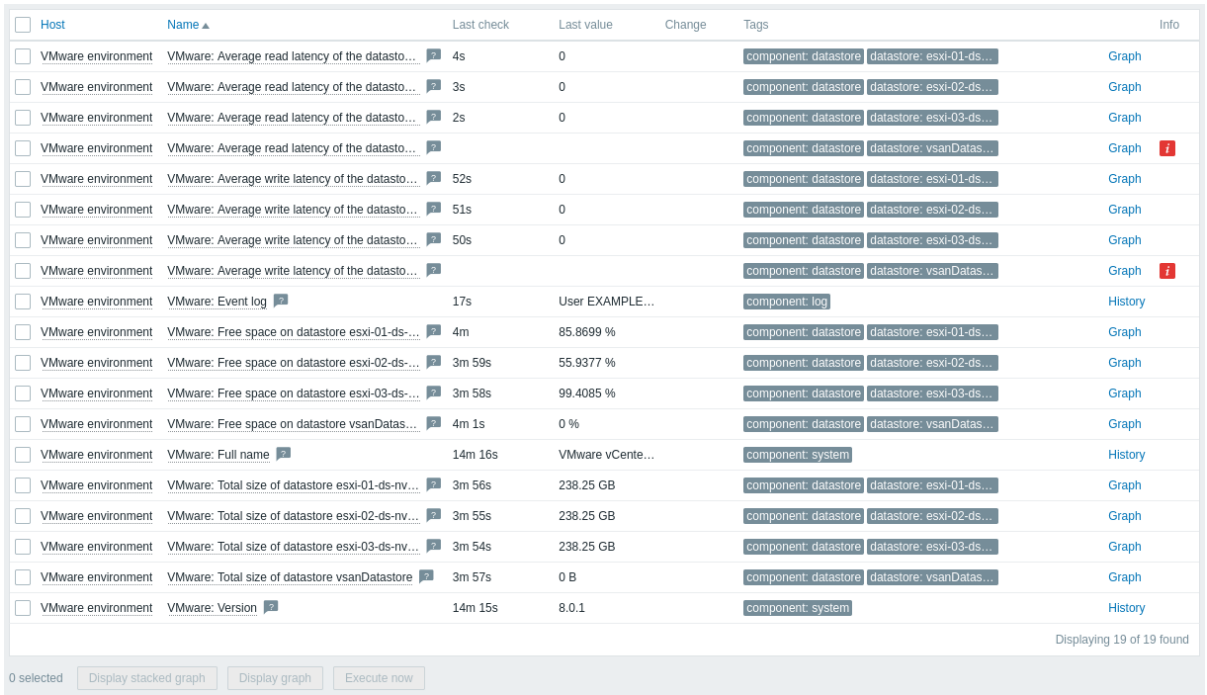
Чтобы просмотреть созданные узлы сети, перейдите в раздел меню *Сбор данных → Узлы сети*.



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.

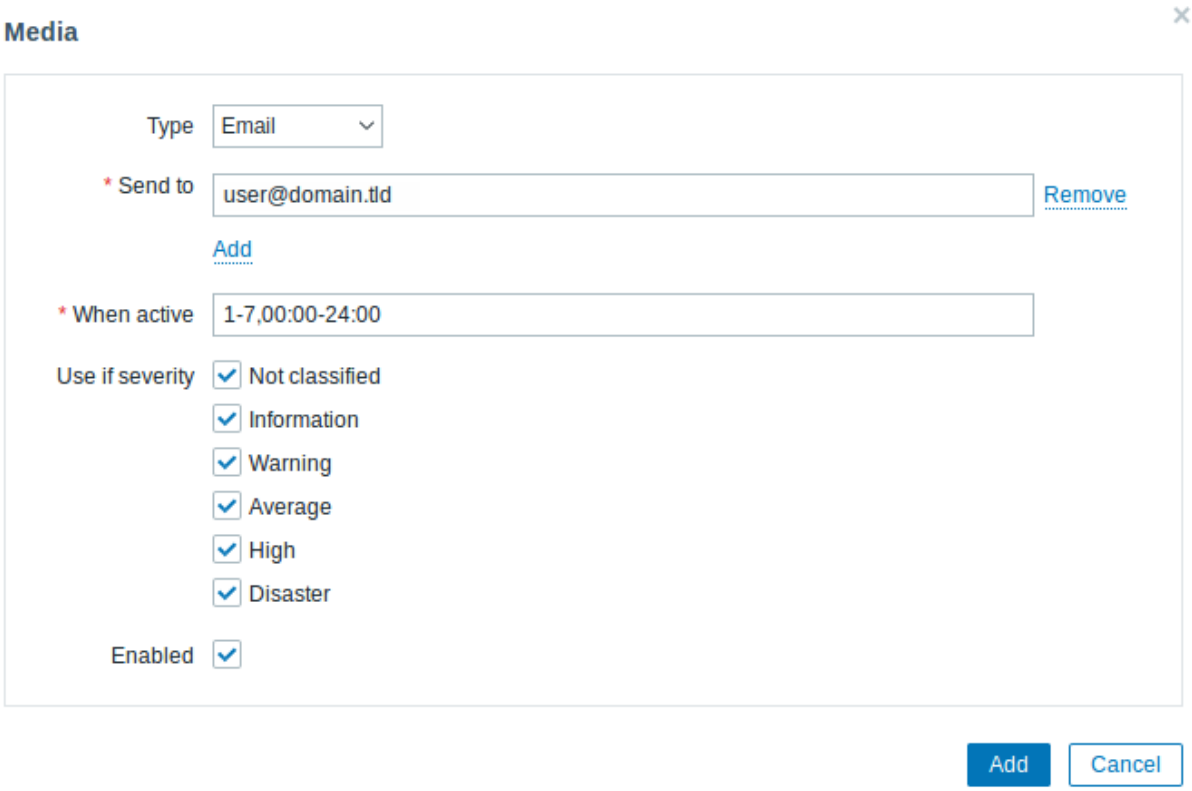


Обратите внимание, что некоторые элементы данных не содержат данных и имеют состояние *Не поддерживается*. Это связано с тем, что Zabbix не может найти допустимые счётчики производительности в конкретном хранилище данных, поскольку они не включены в отслеживаемой среде VMware.

Настройка оповещений о проблемах

Zabbix может уведомлять вас о проблеме в вашей инфраструктуре различными способами. В этом руководстве приведены основные шаги настройки для отправки оповещений по электронной почте.

1. Перейдите в *Настройки пользователя → Профиль*, откройте вкладку *Способы оповещений* и добавьте свой адрес электронной почты.



2. Следуйте руководству Получение 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, затем, если проблема не будет решена в течение 45 минут, отправить сообщение менеджеру центра обработки данных).
- Мониторинг виртуальных машин — дополнительная информация о мониторинге VMware (процесс сбора данных, параметры конфигурации сервера, рекомендации по устранению неполадок и т. д.).
- Ключи элементов данных 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*.

6 Мониторинг сетевого трафика с помощью Zabbix

Введение На этой странице описаны 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 требуется более расширенная конфигурация, см. раздел Configuration руководства Zabbix.

Предварительные требования

Прежде чем продолжить работу с этим руководством, необходимо загрузить и установить Zabbix сервер, веб-интерфейс Zabbix и агент Zabbix в соответствии с инструкциями для вашей ОС. Обратите внимание, что агент Zabbix следует установить на машину, 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 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 смотрите 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 (по умолчанию путь: */usr/local/etc/zabbix_agentd.conf*):

```
sudo vi /usr/local/etc/zabbix_agentd.conf
```

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

```
Server=192.0.2.22
```

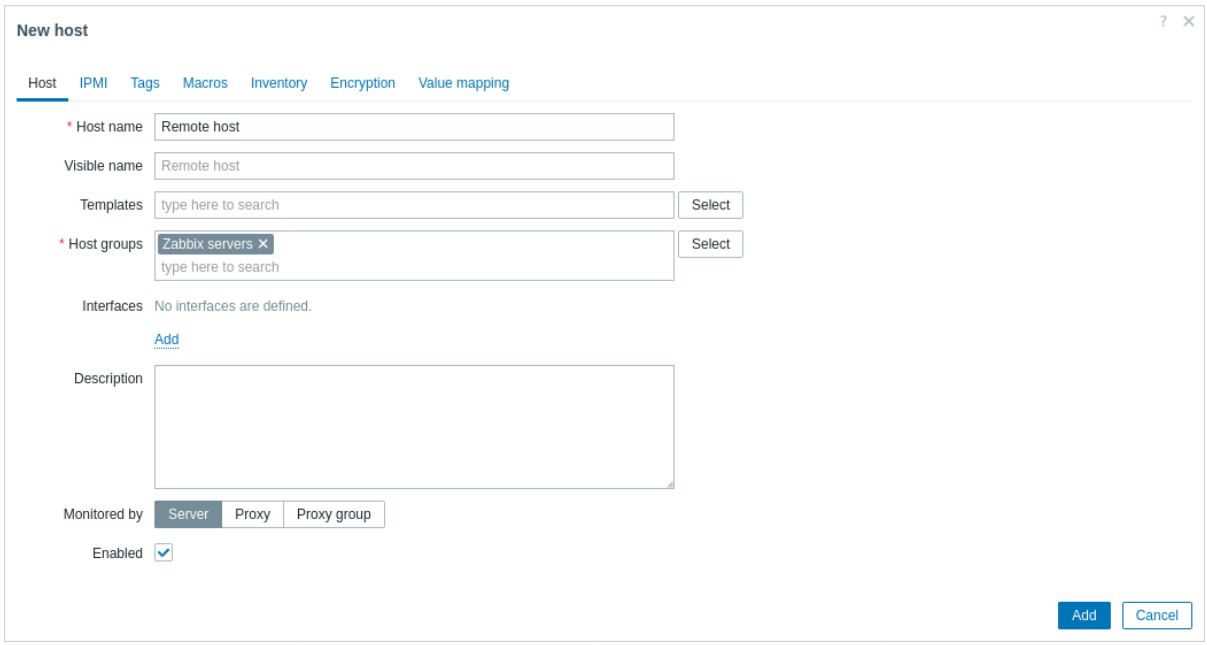
3. Перезапустите агент Zabbix:

```
systemctl restart zabbix-agent
```

Веб-интерфейс Zabbix

1. Войдите в веб-интерфейс Zabbix.

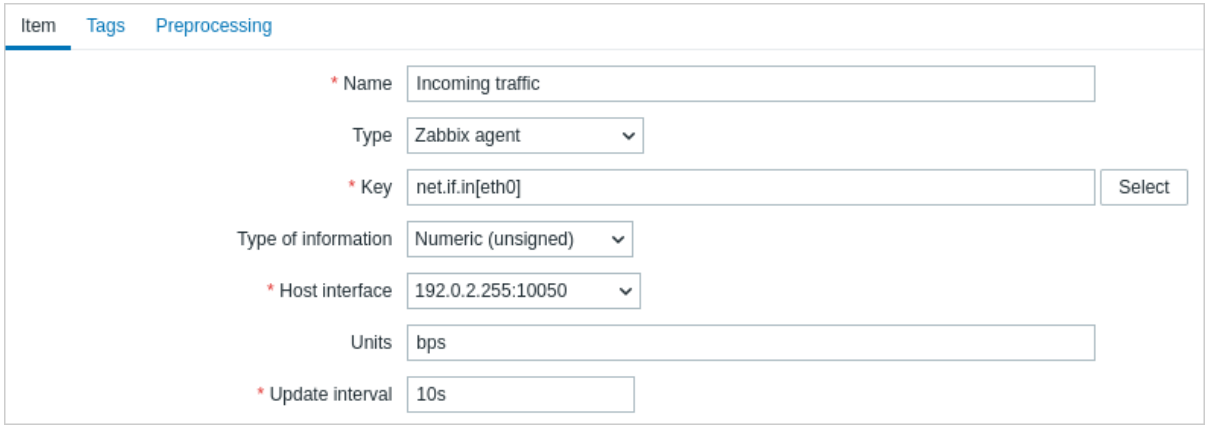
2. Создайте узел сети в веб-интерфейсе Zabbix, указав IP-адрес или DNS-имя машины, на которой установлен агент.



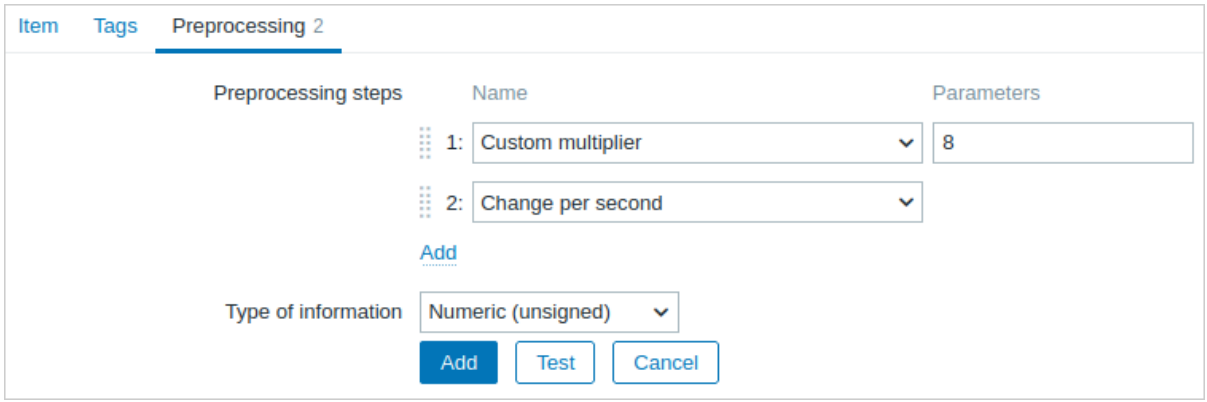
Создание элементов данных Следуйте инструкциям по созданию элемента данных, чтобы добавить элементы данных для мониторинга трафика, а именно:

- Входящий трафик
- Исходящий трафик
- Общий трафик

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

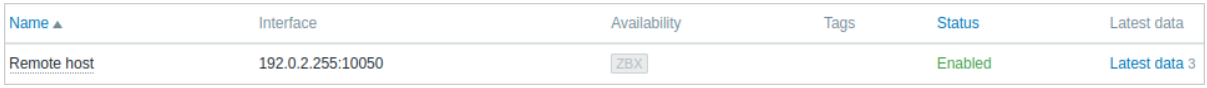


Чтобы сделать собранные данные пригодными для практического использования, при создании элементов данных можно задать несколько шагов предобработки. В данном случае это может быть умножение на 8 (для преобразования байтов в биты) и представление в виде изменения в секунду.

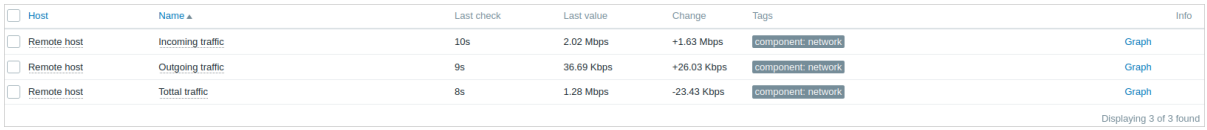


Просмотр собранных данных Поздравляем! На этом этапе Zabbix уже отслеживает трафик вашей сети.

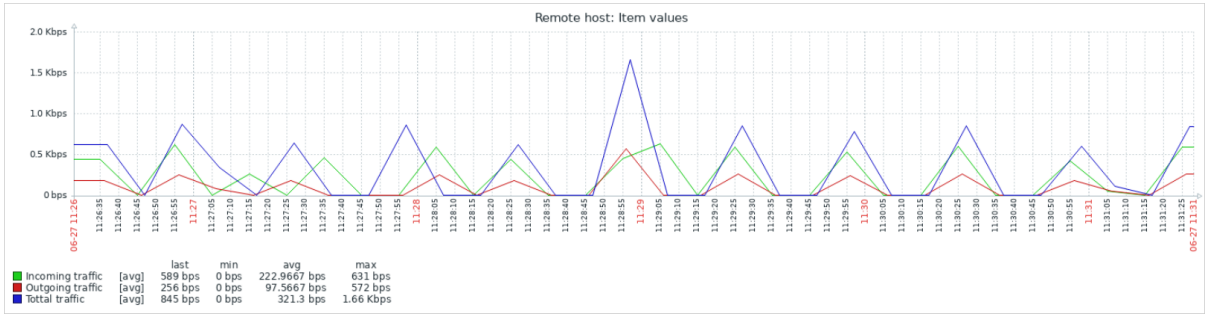
Чтобы просмотреть собранные метрики, откройте раздел меню *Мониторинг → Узлы сети* и нажмите *Последние данные* в строке узла сети.



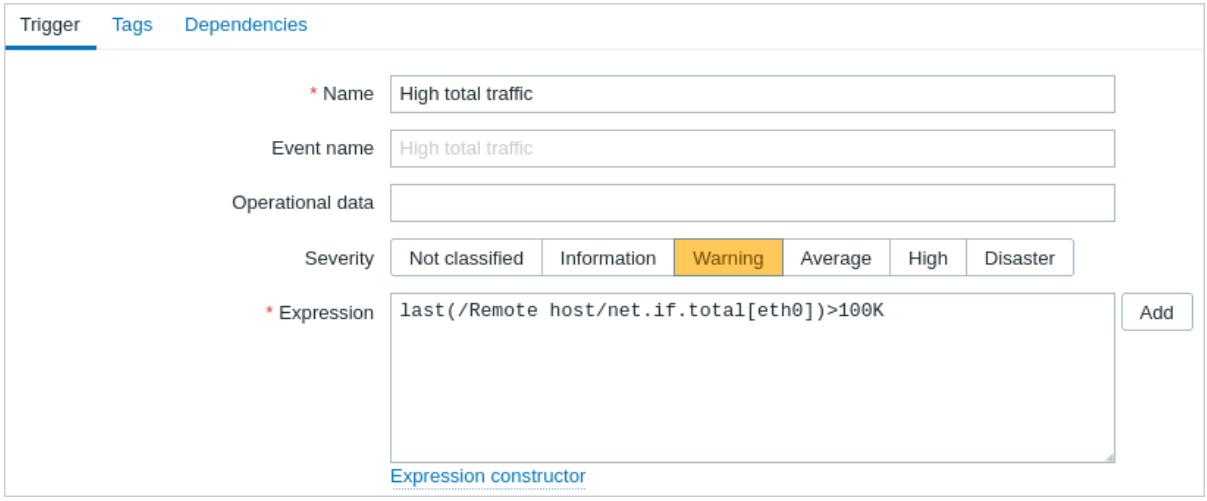
Вы увидите данные о трафике.



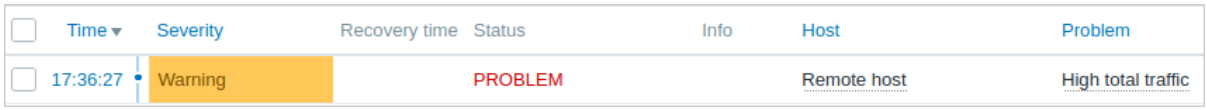
Просмотр графиков Собранные данные могут быть отображены в виде графиков. Чтобы просмотреть их, в разделе *Последние данные* нажмите *График* в строке элемента данных или выберите необходимые элементы данных и нажмите *Отобразить график* ниже.



Настройка триггеров Вы можете настроить 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 *Мониторинг* → *Проблемы*, чтобы проверить, что проблема отображается в списке.



Настройка оповещений о проблемах Существует несколько способов получения уведомлений о проблеме. Электронная почта является самым популярным из них, поэтому следуйте инструкциям по настройке уведомления о проблеме по электронной почте. Вы также можете выбрать другие типы медиа для доставки уведомлений.

См. также:

- Эскалации проблем — как создавать многошаговые сценарии оповещений (например, сначала отправить сообщение системному администратору, затем, если проблема не будет 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, привязав предварительно настроенный шаблон.

7 Мониторинг сетевого трафика с использованием 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 требуется более расширенная конфигурация, см. раздел Configuration руководства Zabbix.

Предварительные требования

Прежде чем 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 машину, для которой требуется мониторинг трафика. Это может быть как тот же узел сети, на котором установлен сервер Zabbix, так и другой узел сети.

В этом руководстве 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 смотрите в разделе Пассивные и активные проверки агента. В этом руководстве будет описан мониторинг с помощью **активных проверок**.

Настройка Zabbix агент

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

Если вы используете Zabbix агент:

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

Если вы используете Zabbix агент 2:

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

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

```
ServerActive=192.0.2.0:10051
```

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

3. Укажите 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, потому что для активных проверок Zabbix агент использует его для получения правильной конфигурации узла сети с сервера. В частности, агент инициирует соединение с сервером 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 Zabbix агент.

Если вы используете Zabbix агент:

```
systemctl restart zabbix-agent
```

Если вы используете Zabbix агент 2:

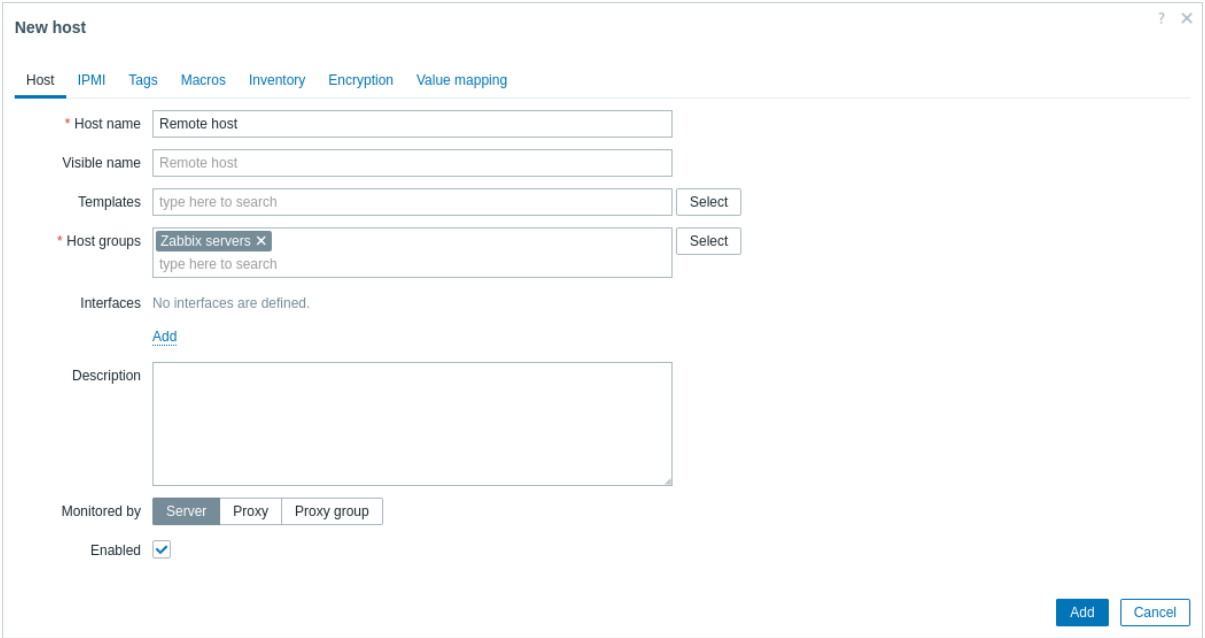
```
systemctl restart zabbix-agent2
```

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 поле *Interfaces* пустым; интерфейс не требуется для активных проверок, поскольку агент сам инициирует соединение с сервером, а не ожидает входящего соединения от сервера.

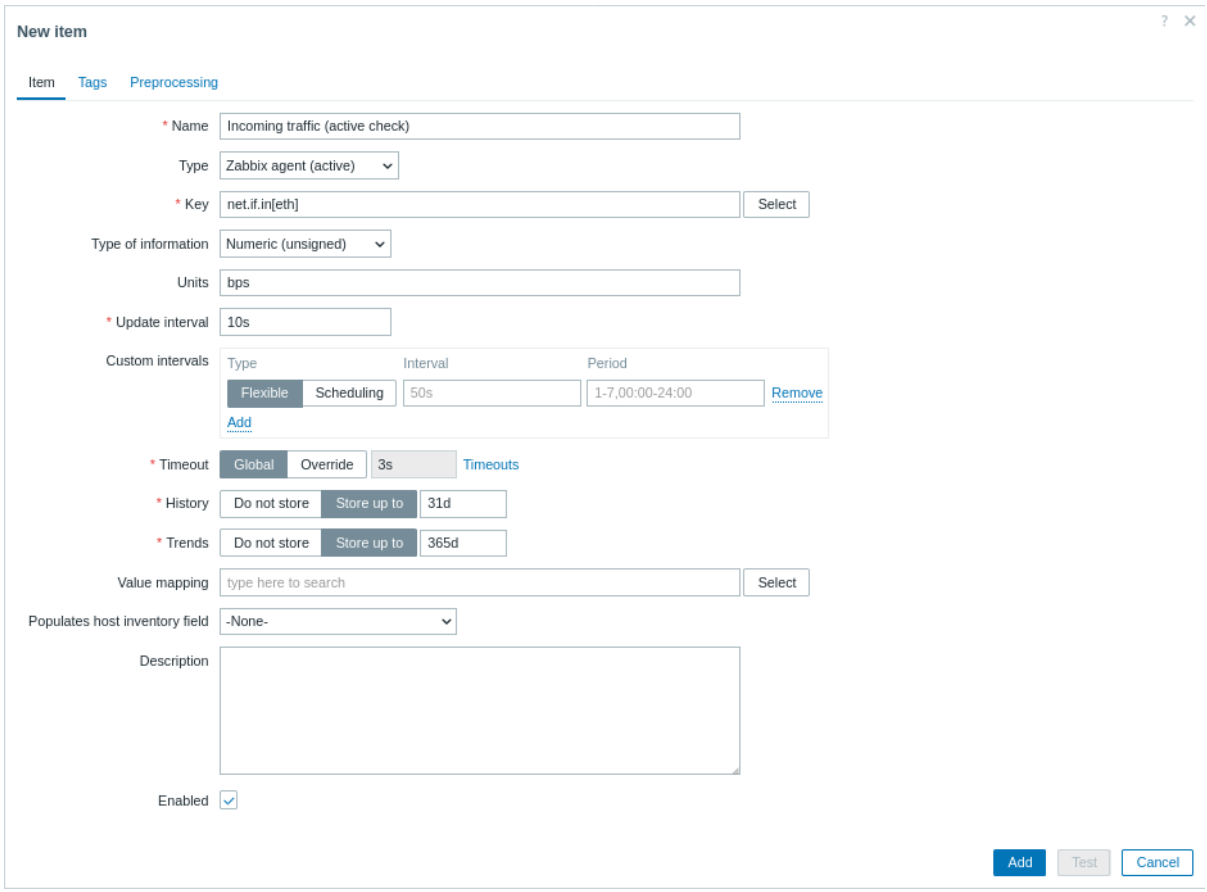


3. Нажмите *Add*, чтобы добавить узел сети. Этот узел сети будет представлять отслеживаемую Linux-машину.

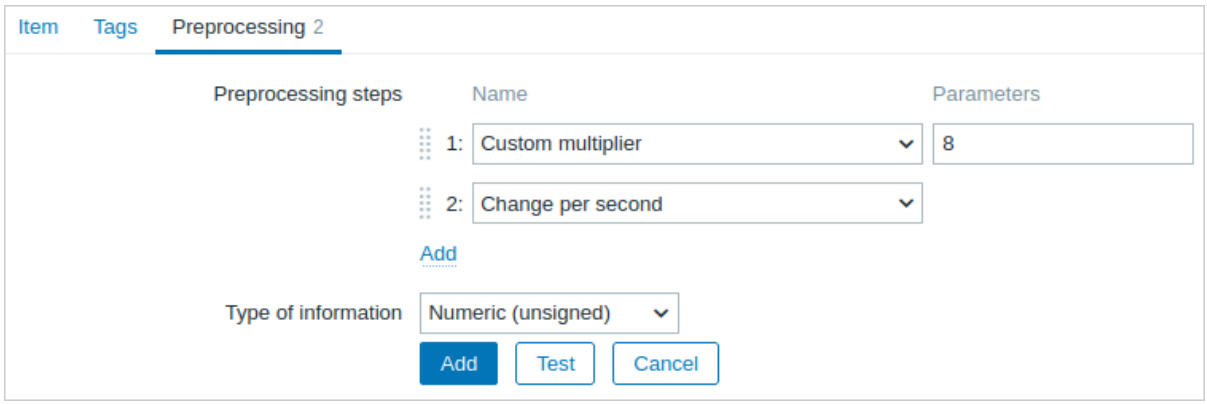
Создание элементов данных Следуйте инструкциям по созданию элемента данных, чтобы добавить элементы данных для мониторинга трафика, а именно:

- Входящий трафик
- Исходящий трафик
- Общий трафик

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

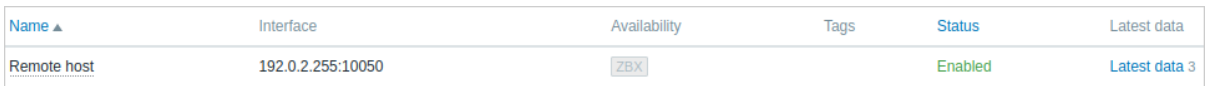


Чтобы сделать собранные данные пригодными для практического использования, при создании элементов данных можно настроить несколько шагов предобработки. В данном случае это может быть умножение на 8 (для преобразования байтов в биты) и представление в виде изменения в секунду.

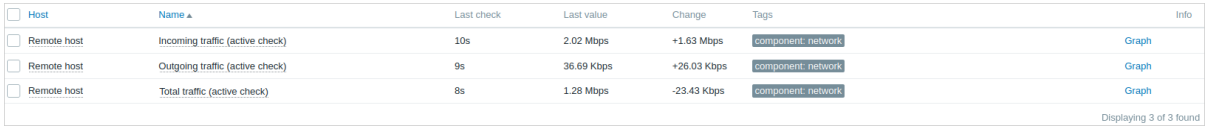


Просмотр собранных данных Поздравляем! На этом этапе Zabbix уже отслеживает трафик вашей сети.

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



Вы увидите данные о трафике.



Host	Name ▲	Last check	Last value	Change	Tags	Info
Remote host	Incoming traffic (active check)	10s	2.02 Mbps	+1.63 Mbps	component: network	Graph
Remote host	Outgoing traffic (active check)	9s	36.69 Kbps	+26.03 Kbps	component: network	Graph
Remote host	Total traffic (active check)	8s	1.28 Mbps	-23.43 Kbps	component: network	Graph

Displaying 3 of 3 found

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 будет решена в течение 45 минут, отправить сообщение менеджеру центра обработки данных).
- Подтверждение проблемы — как указать, что проблема известна, добавлять комментарии о её решении, подавлять или закрывать проблему.
- Мониторинг Linux с помощью Zabbix агент — как начать базовый мониторинг наиболее важных элементов данных, привязав предварительно настроенный шаблон.
- Установка из пакетов — как установить компоненты Zabbix с использованием официальных пакетов RPM и DEB для различных дистрибутивов Linux, обеспечивая доступ к новейшим возможностям и исправлениям ошибок.

8 Мониторинг веб-сайтов с помощью элементов данных браузера

Введение

На этой странице описаны шаги, необходимые для начала базового мониторинга веб-сайтов с помощью элементов данных браузера.

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 расширенная конфигурация, см. страницу Элементы данных Browser или раздел Конфигурация руководства Zabbix.

Предварительные требования

Прежде чем 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 немного отличаться. Это руководство основано на среде с Ubuntu.

Настройка WebDriver

Элементы данных браузера требуют наличия платформы автоматизации (либо Selenium Server, либо обычного 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 Docker-контейнере.

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

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

- **docker run --name browser** — запускает новый Docker-контейнер с именем ”browser”;
- **-p 4444:4444** — сопоставляет порт 4444 на вашей машине с портом 4444 контейнера (этот порт используется Selenium Server для приема команд);
- **-p 7900:7900** — сопоставляет порт 7900 на вашей машине с портом 7900 контейнера (этот порт используется сервером Virtual Network Computing (VNC), позволяя вам 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 во избежание сбоев);
- **-d** — запускает контейнер в фоновом режиме;
- **selenium/standalone-chrome:latest** — указывает Docker-образ, который следует использовать; в данном случае — последнюю версию 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 Docker-контейнер `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 от 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 обратитесь к документации Docker.

Настройка сервера Zabbix

Элементы данных браузера выполняются и обрабатываются процессами Zabbix *browser poller*, которые необходимо включить, настроив параметр конфигурации сервера StartBrowserPollers. Дополнительно параметр WebDriverURL должен указывать ранее настроенную конечную точку веб-тестирования.

По умолчанию параметр StartBrowserPollers имеет значение 1, поэтому вам нужно только указать 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
###HTTP[S] URL       WebDriver.      , http://localhost:4444                    Selenium WebDriver.
###
### Mandatory: no
### Default:
### WebDriverURL=
```

```
WebDriverURL=192.0.2.1:4444
```

3. Перезапустите сервер Zabbix.

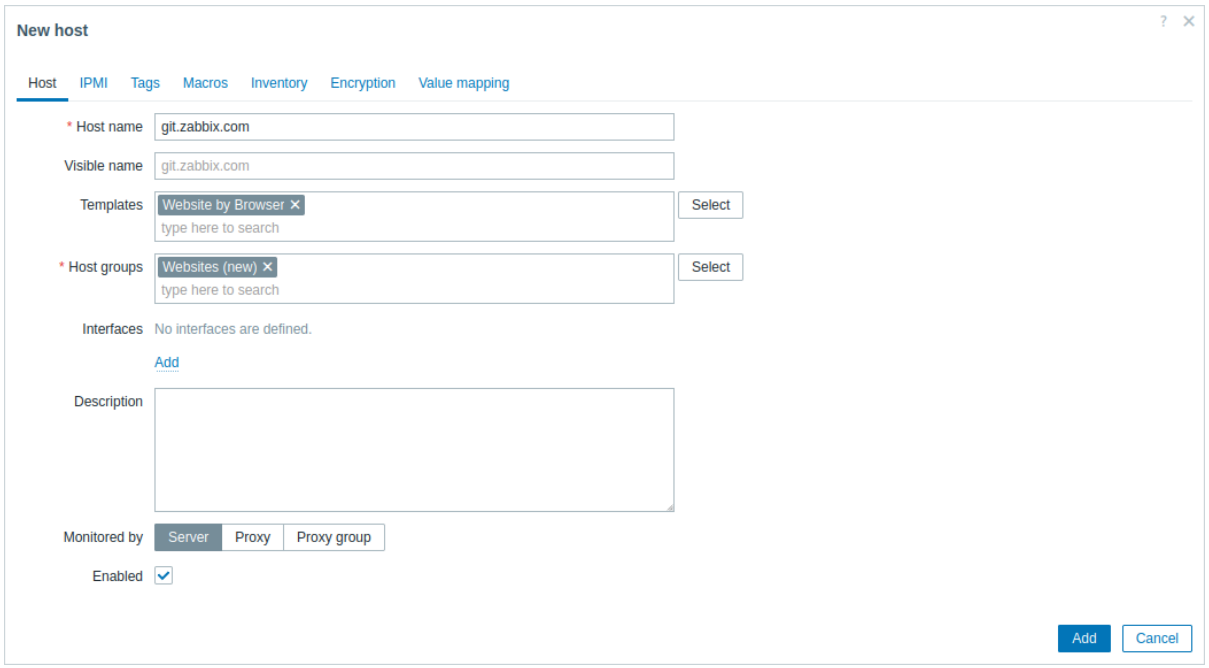
```
systemctl restart zabbix-server
```

Настройка веб-интерфейса Zabbix

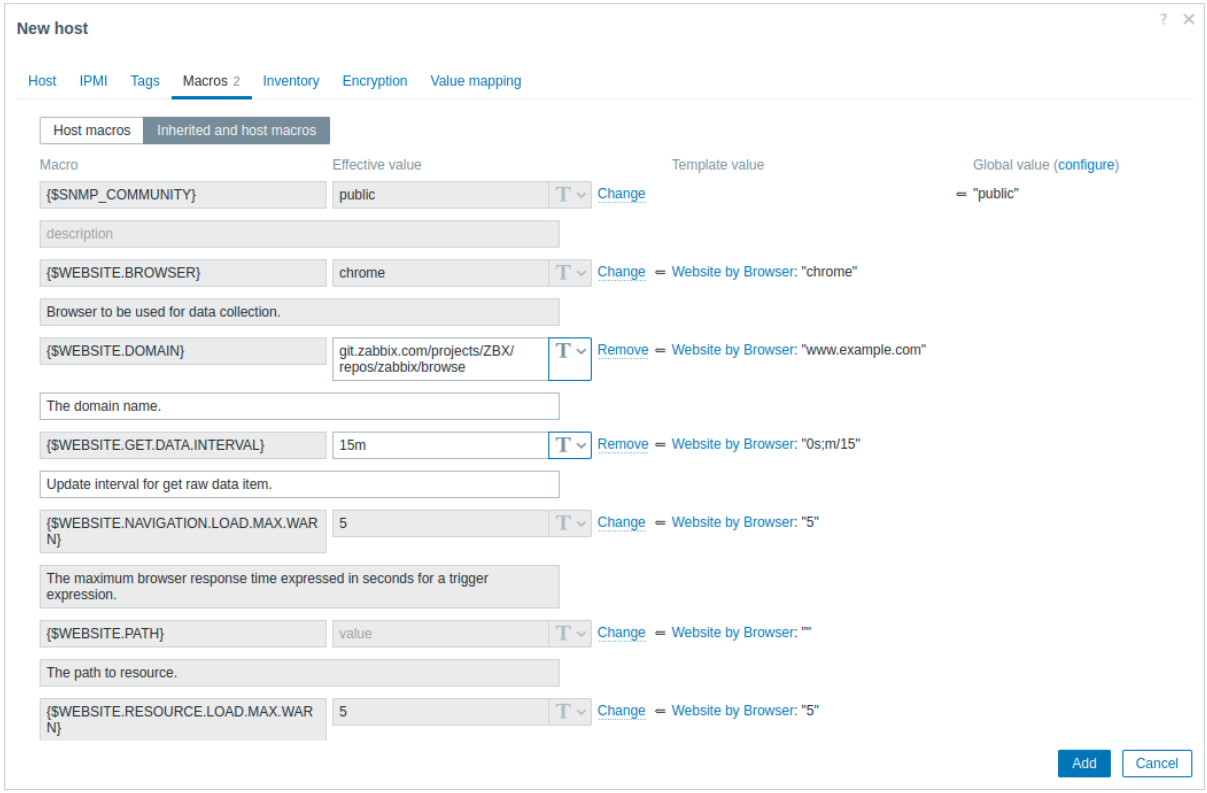
1. Войдите в веб-интерфейс Zabbix.

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”. Дополнительную информацию об этом шаблоне см. в 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)

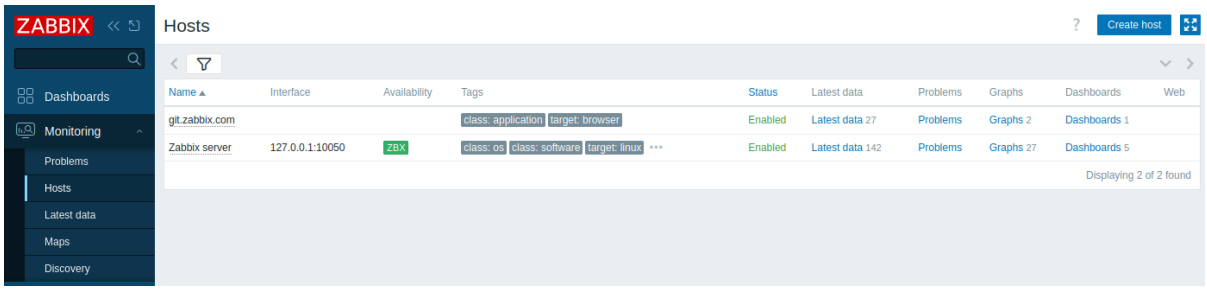


3. Нажмите кнопку *Add*, чтобы создать узел сети. Этот узел сети будет представлять веб-сайт, который вы хотите отслеживать.

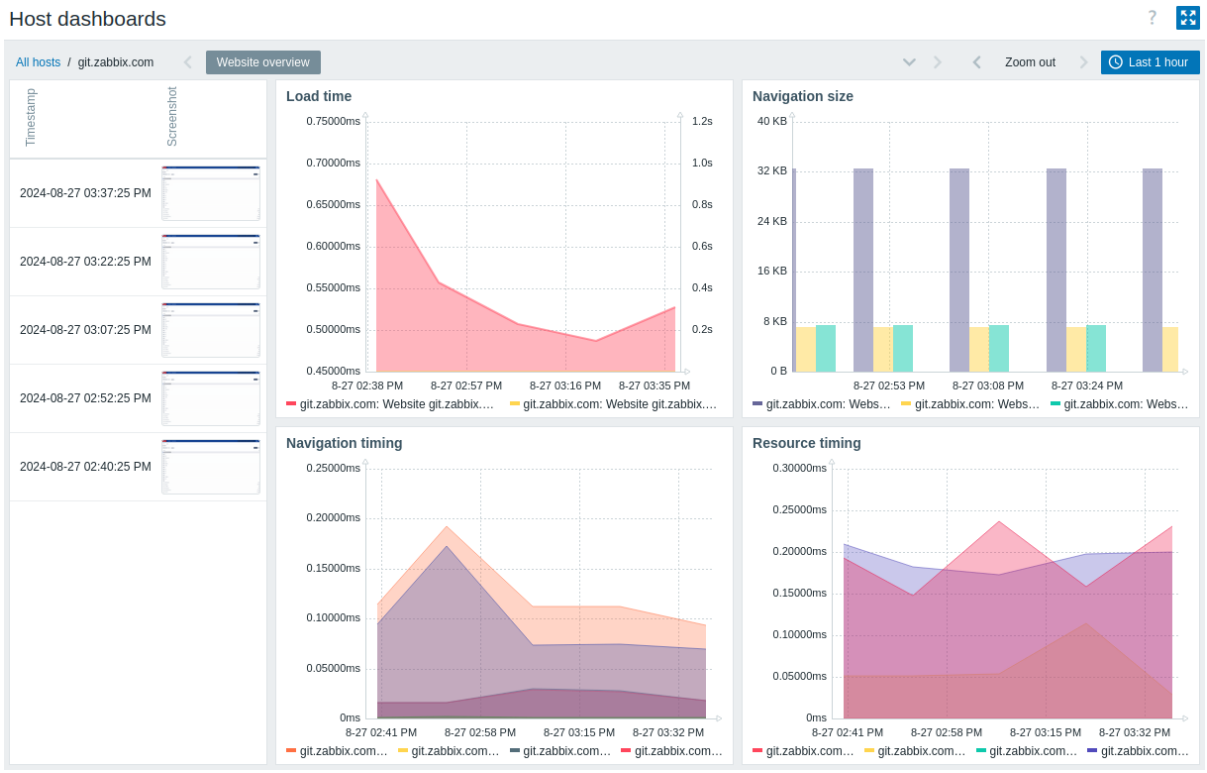
Просмотр собранных метрик

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

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



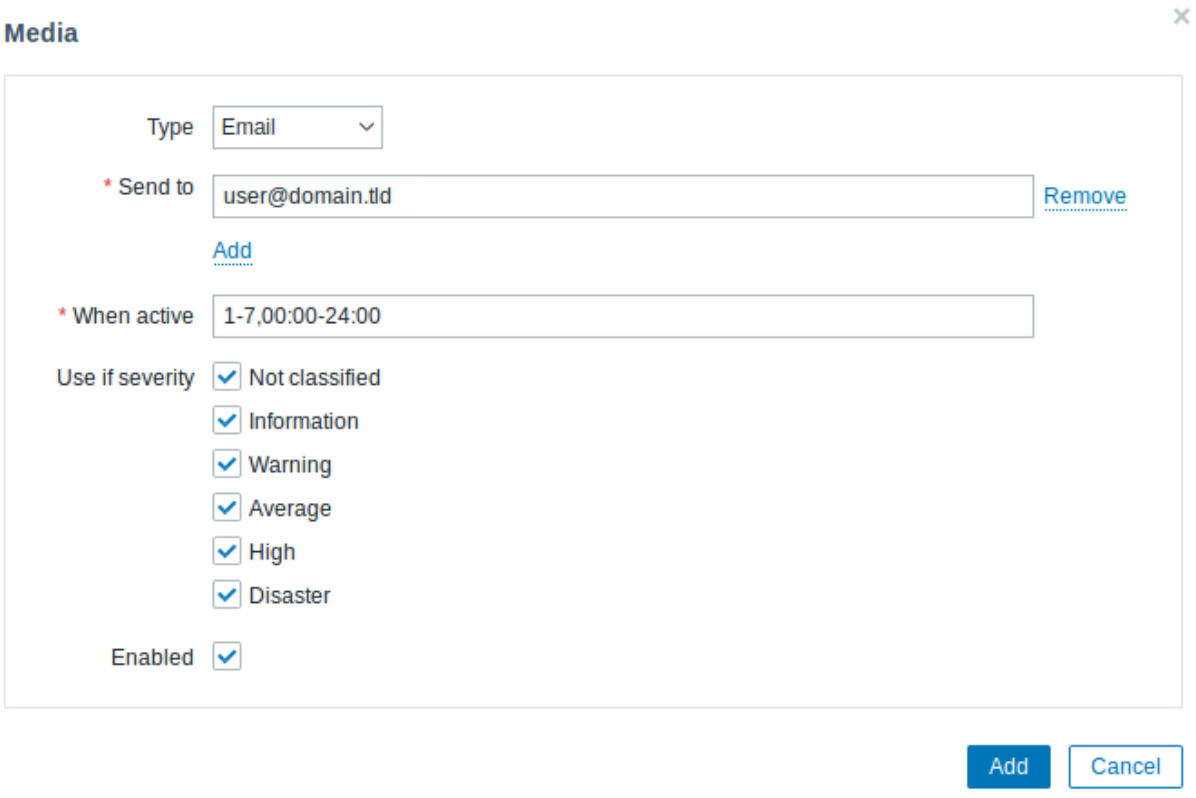
Это действие перенаправит вас на панель узла сети (настроенную на уровне шаблона) с наиболее важными метриками, собранными с веб-сайта.



Настройка оповещений о проблемах

Zabbix может уведомлять вас о проблемах в вашей инфраструктуре различными способами. В этом руководстве приведены основные шаги по настройке отправки оповещений по электронной почте.

1. Перейдите в *Настройки пользователя → Профиль*, откройте вкладку *Способы оповещения* и добавьте свой адрес электронной почты.



2. Следуйте руководству Получение 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, с ранее настроенным значением макроса {$WEBSITE.DOMAIN} и укажите неверное доменное имя (например, /git.zabbix.com/projects/ZBX/repos/zabbix/browse).

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

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



Если оповещения настроены, вы также получите уведомление о проблеме.

6. Измените значение макроса обратно на предыдущее, чтобы устранить проблему и продолжить мониторинг веб-сайта.

См. также

- Создание элемента данных — как начать мониторинг дополнительных метрик.
- Эскалации проблем — как создавать многошаговые сценарии оповещений (например, сначала отправить сообщение системному администратору, затем, если проблема не будет 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*.

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

Введение

Это руководство содержит исчерпывающий обзор того, как настроить мониторинг SSL/TLS-сертификатов с помощью ключа `web.certificate.get` в Zabbix агент 2. Оно предназначено для упрощения мониторинга сертификатов для одного или нескольких веб-сайтов, чтобы администраторы могли быстро выявлять потенциальные проблемы, такие как истекшие 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-сайта. Если вам нужны возможности глубокой настройки или более расширенная конфигурация, см. раздел Configuration руководства Zabbix.

Предварительные требования

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

Настройка Zabbix агент 2

1. Откройте файл конфигурации Zabbix агента (путь по умолчанию: `/etc/zabbix/zabbix_agent2.conf`):

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

2. Установите параметр Server в значение 127.0.0.1, так как агент и сервер запущены на одной и той же машине:

```
Server=127.0.0.1
```

3. Сохраните файл и перезапустите службу Zabbix агент 2:

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

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

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

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

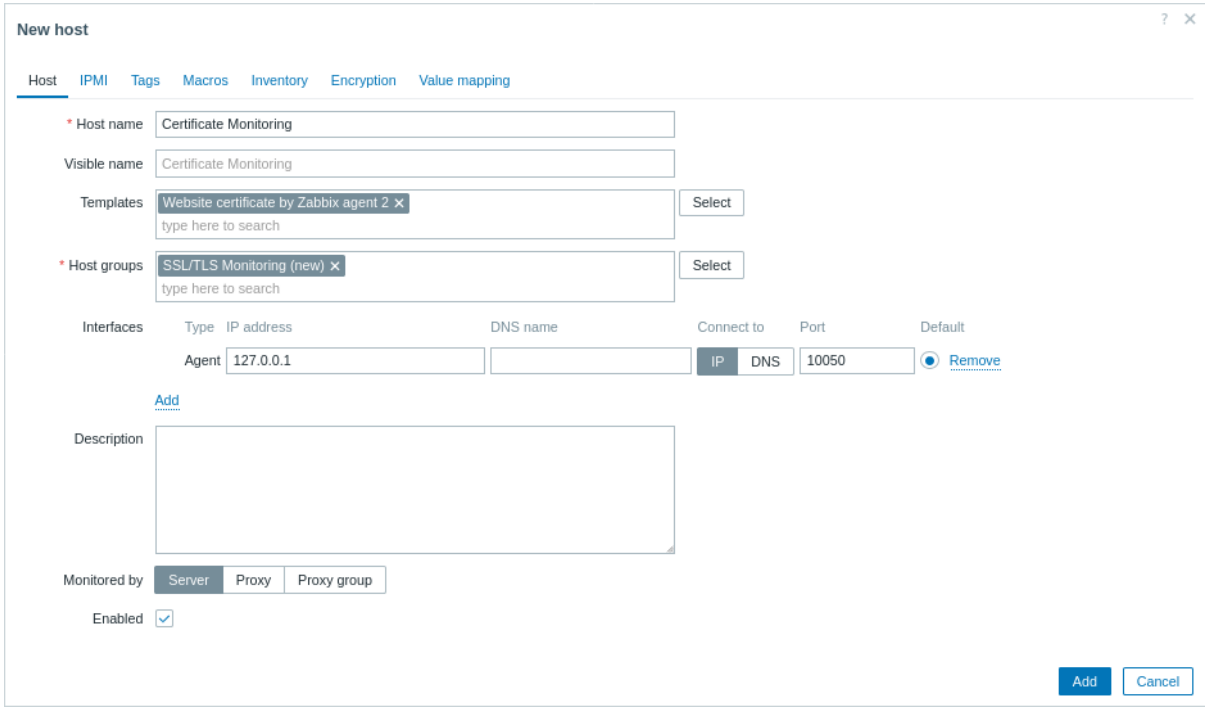
Настройка веб-интерфейса Zabbix

1. Войдите в веб-интерфейс Zabbix.

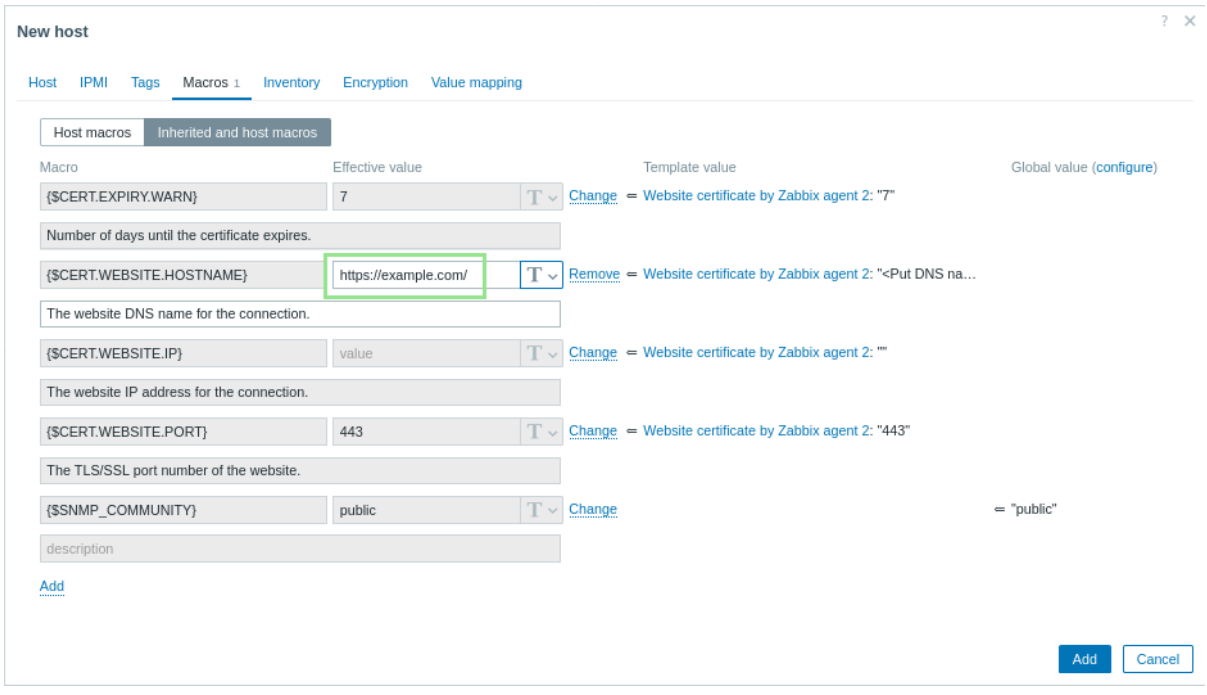
2. Перейдите в *Monitoring* > *Hosts*.

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

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



- На вкладке *Macros* переключитесь на *Inherited and host macros*, найдите следующие макросы и нажмите *Change* рядом со значением макроса, чтобы обновить его:
  - `{$CERT.WEBSITE.HOSTNAME}` — введите в качестве значения нужное DNS-имя веб-сайта.



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

Чтобы отслеживать несколько веб-сайтов, вы можете указать список их имен узлов, разделенных запятыми, в макросе {$CERT.WEBSITE.HOSTNAME}. При необходимости вы также можете указать список портов, разделенных запятыми, в макросе {$CERT.WEBSITE.PORT}, где каждый порт соответствует имени узла в том же порядке. Например, чтобы отслеживать example.com и example.org на порту 8443:

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

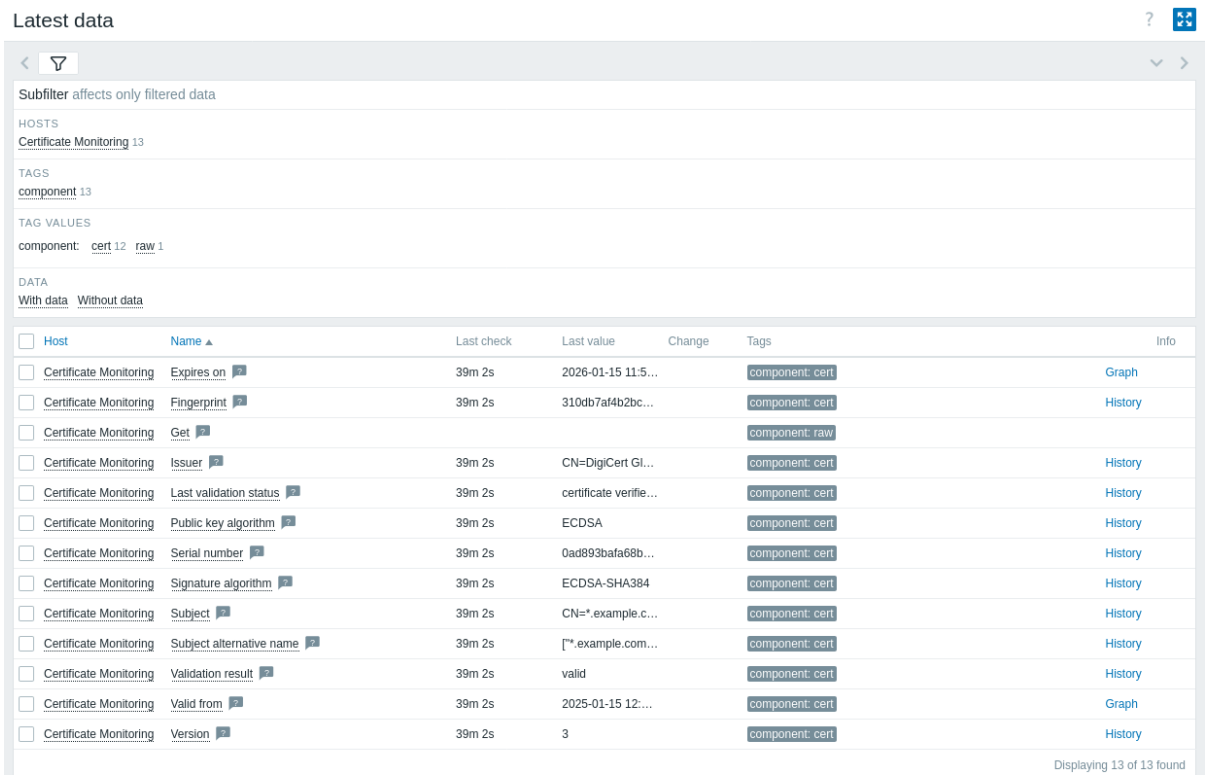
Для каждого веб-сайта, указанного в макросе {$CERT.WEBSITE.HOSTNAME}, Zabbix создаст соответствующий набор элементов данных и триггеров. Это позволяет выполнять индивидуальный мониторинг и оповещение для SSL-сертификата каждого веб-сайта.

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 выполняет мониторинг нужного веб-сертификата.

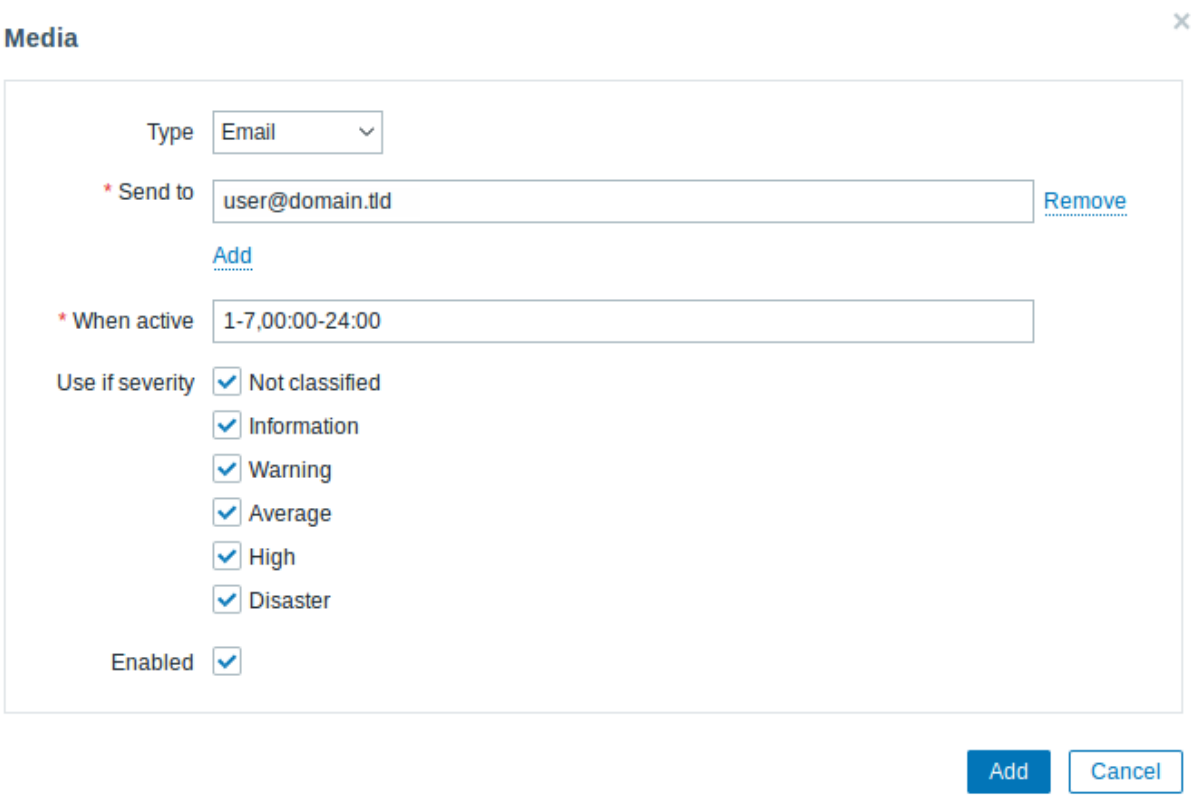
Чтобы просмотреть собранные метрики, перейдите в раздел меню *Мониторинг > Узлы сети* и нажмите *Последние данные* рядом с узлом сети, чтобы увидеть все последние собранные метрики в виде списка, такие как дата истечения срока действия, издатель и субъект.



Настройка оповещений о проблемах

Zabbix может уведомлять вас о проблемах в инфраструктуре. В этом руководстве приведены основные шаги по настройке отправки оповещений по электронной почте.

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



2. Следуйте руководству Получение 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-интерфейс Zabbix.

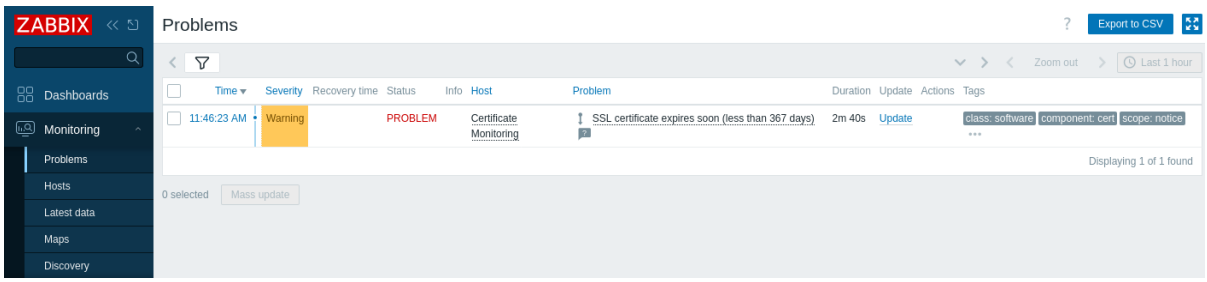
1. Откройте конфигурацию вашего узла сети "Certificate Monitoring" в Zabbix.

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

3. Нажмите *Change* рядом с ранее настроенным значением макроса `{$CERT.EXPIRY.WARN}` и установите очень большое количество дней (более 365 дней должно быть достаточно), чтобы получить предупреждение до истечения срока действия сертификата.

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

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



Если оповещения настроены, вы также получите уведомление о проблеме.

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

См. также

- Zabbix агент 2 — список ключей 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*.
- Мониторинг веб-сайтов с помощью элементов данных Browser — как начать базовый мониторинг веб-сайтов с помощью элементов данных Browser.

10 Мониторинг сетевого коммутатора или маршрутизатора с помощью Zabbix

Введение Это руководство поможет вам выполнить шаги, необходимые для запуска базового мониторинга вашего сетевого коммутатора или маршрутизатора с помощью Zabbix. В качестве примера используется маршрутизатор Cisco, однако 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 к разделу Configuration руководства Zabbix.

Предварительные требования

Перед продолжением убедитесь, что у вас есть:

- Установленные Zabbix сервер и Zabbix веб-интерфейс: выполните установку в соответствии с инструкциями для вашей операционной системы (см. Installation from packages и Web interface installation).
- Установленный Zabbix агент, если требуется мониторинг локальных сетевых метрик.
- Устройство с поддержкой SNMP: сетевой коммутатор или маршрутизатор (например, маршрутизатор Cisco) с включённым SNMP.
- Установленные MIB files: установка MIB-файлов позволяет Zabbix преобразовывать числовые OID в понятные человеку имена и описания. При отсутствии надлежащей поддержки MIB вы можете видеть только числовые значения, что затрудняет настройку элементов данных и устранение проблем.

Чтобы установить MIB-файлы в Ubuntu:

1. Установите пакет загрузчика MIB:

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

Если необходимо добавить MIB-файлы от конкретного производителя (например, Cisco, Juniper), поместите их в соответствующий каталог MIB:

- Для систем на базе Linux обычно используются каталоги /usr/share/snmp/mibs/ или /usr/local/share/snmp/mibs/.
- 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, распознаёт ли система новые MIB, используйте:

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

Подробные инструкции см. в документации вашей SNMP-библиотеки:

- Cisco MIBs
- Juniper MIBs

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

3. Выполните проверку с помощью `snmpwalk` (например, `snmpwalk -v 2c -c <your_community_string> <device_IP>`) и убедитесь, что OID отображаются с понятными 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 по SNMP необходимо настроить ваше сетевое устройство так, чтобы оно разрешало SNMP-запросы. Пример ниже приведён для SNMPv2 и не учитывает существующие настройки. Внимание: применение этих команд может переопределить текущие конфигурации 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 конфигурирования:

```
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 защищённую строку community. Примечание: параметр RO (Read-Only) позволяет SNMP получать данные с устройства, но предотвращает любые изменения конфигурации.

В целях безопасности рекомендуется ограничить доступ по SNMP только необходимыми устройствами. Дополнительные рекомендации по 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 конфигурация более безопасна, чем у SNMPv2, и должна быть проверена в соответствии с документацией для вашего устройства.

1. Создайте группу SNMP.

Настройте группу SNMPv3 с включенной privacy (шифрованием):

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

2. Создайте пользователя SNMP.

Добавьте пользователя SNMPv3 с аутентификацией и privacy. Замените заполнители на нужные вам значения:

```
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 для конкретной модели вы можете обратиться к внешним руководствам Cisco по настройке SNMP. Однако это руководство содержит основные шаги по включению мониторинга SNMP.

Настройка веб-интерфейса Zabbix

Создание узла сети в веб-интерфейсе Zabbix 1. Войдите в веб-интерфейс Zabbix.

2. Добавьте новый узел сети.

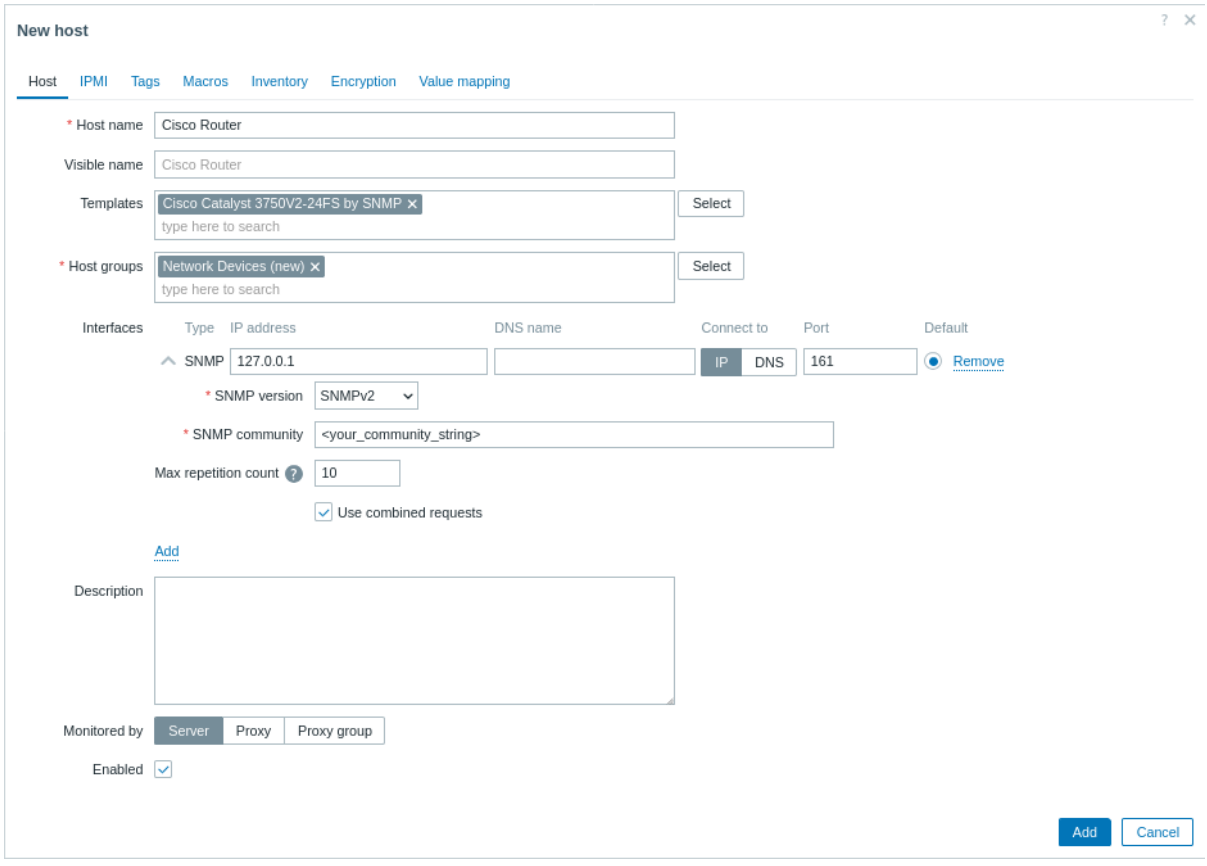
Перейдите в *Data collection* > *Hosts* и нажмите *Create host*.

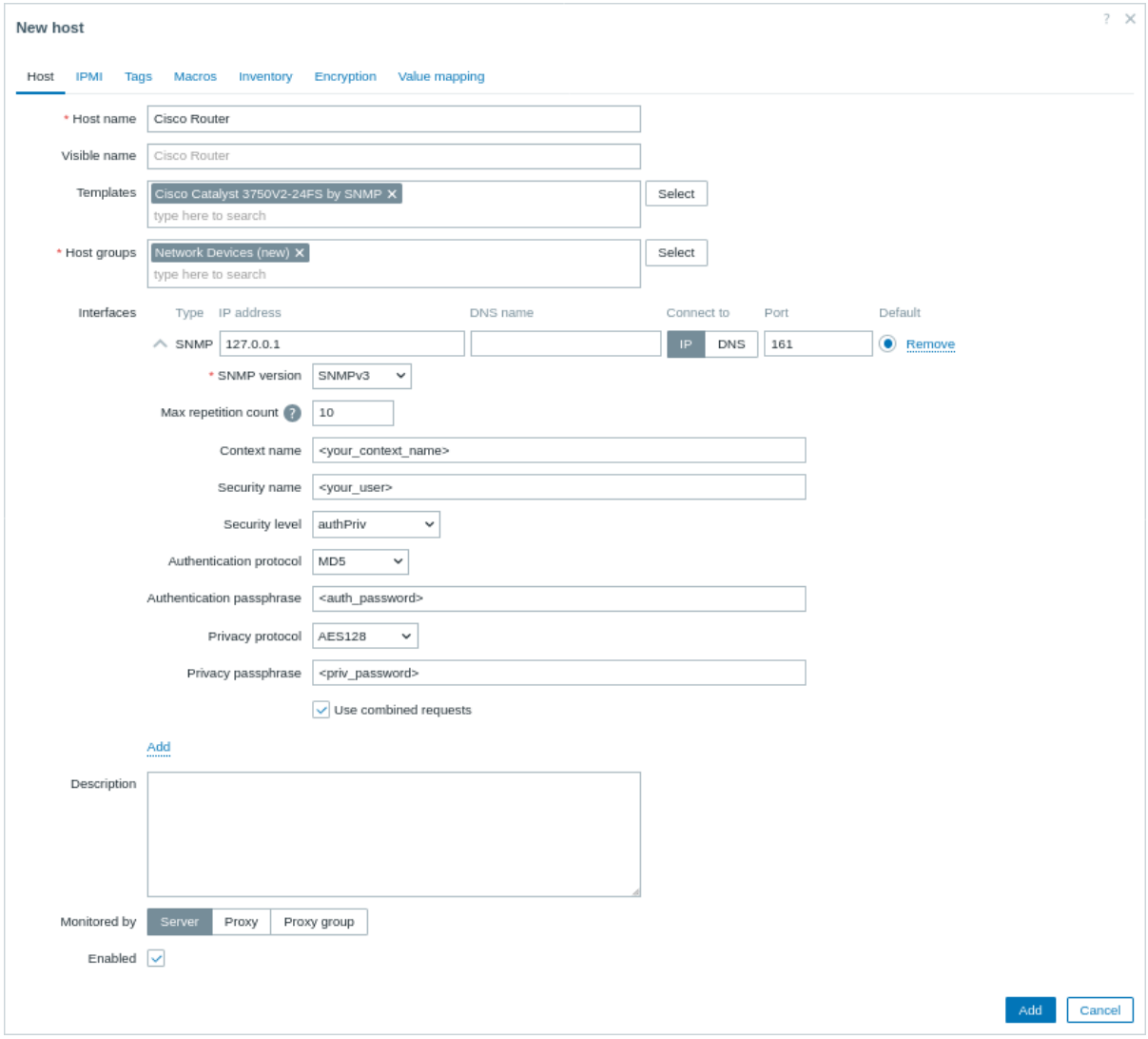
- *Host name*: введите имя вашего устройства (например, ”Cisco Router”).
- Host groups: выберите существующую группу или создайте новую, например ”Network Devices”.
- Interfaces:
  - Нажмите *Add* в разделе Interfaces.
  - Выберите *SNMP* в качестве типа интерфейса.
  - Введите IP-адрес или DNS-имя вашего маршрутизатора Cisco.
  - Укажите стандартный порт SNMP (обычно 161).
  - В раскрывающемся списке выберите подходящую версию SNMP (например, SNMPv2).
  - Для SNMPv1/v2 введите строку community в поле *SNMP community*. Для SNMPv3 будет предложено указать дополнительные учетные данные (*Context name*, *Security name* и *Security level* и т. д.).

3. Привязка шаблонов

В поле *Templates* выберите шаблон SNMP, который лучше всего соответствует вашему устройству. Zabbix предоставляет ряд 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.

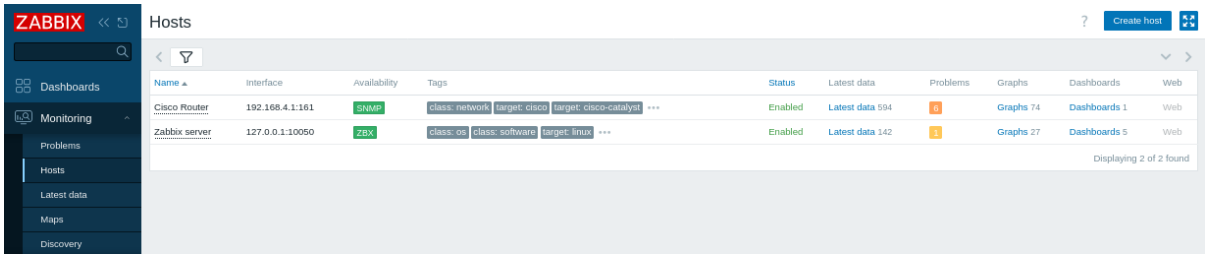




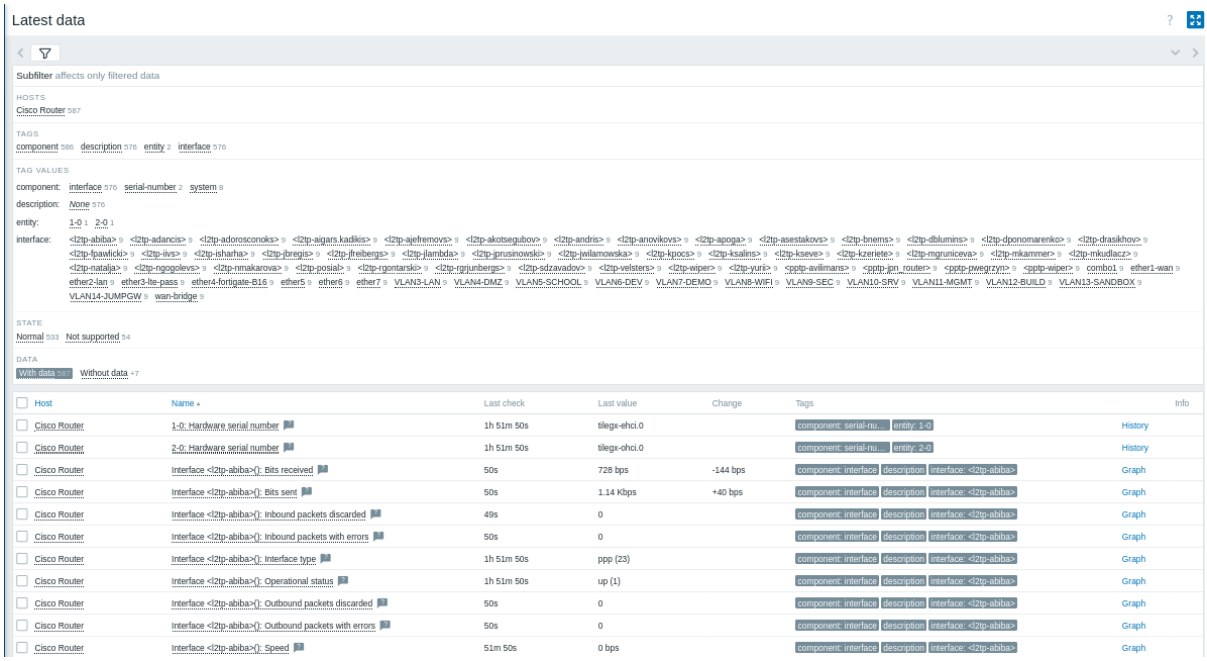
Просмотр собранных метрик Поздравляем! Теперь Zabbix настроен для мониторинга вашего сетевого устройства.

Последние данные:

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



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



- Графики и комплексные экраны:

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

В качестве следующего шага вы можете:

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

Создание SNMP-элементов данных После настройки узла сети вы можете создать элементы данных для мониторинга определённых метрик. Примечание: этот шаг необязателен, если вы используете шаблон, так как шаблоны уже содержат стандартные наборы элементов данных.

1. Определите SNMP OID:

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

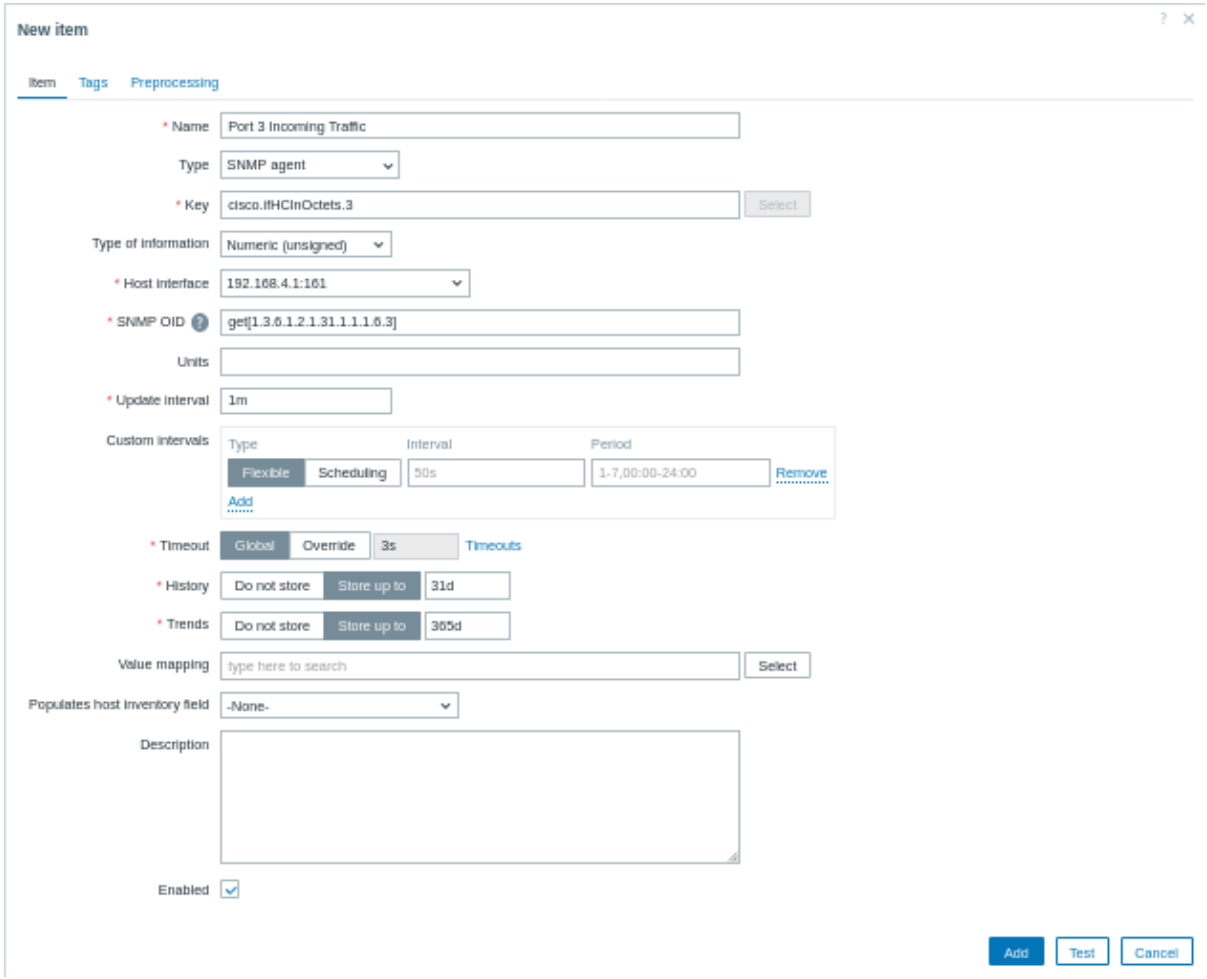
```
snmpwalk -v 2c -c <your_community_string> <device_IP> .
```

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:

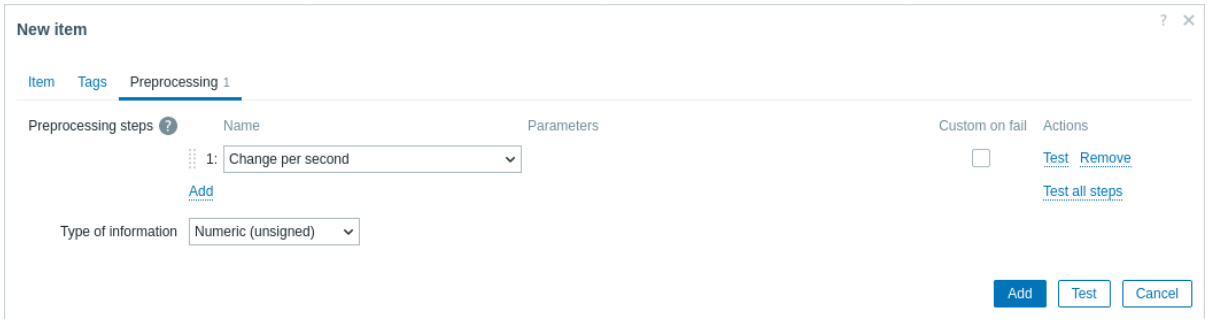
```
snmpget -v 2c -c <your_community_string> -On <device_IP> IF-MIB::ifHCInOctets.3
```

2. Создайте SNMP-элемент данных:

- Перейдите в *Data collection* > *Hosts*, откройте вкладку *Items* для вашего SNMP-узла сети и нажмите *Create item*.
- *Name*: введите понятное имя (например, «Входящий трафик порта 3»).
- *Type*: выберите *SNMP агент*.
- *Key*: укажите понятный ключ (например, `cisco.ifHCInOctets.3`).
- *Host interface*: убедитесь, что выбран SNMP-интерфейс.
- *SNMP OID*: введите OID в одном 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 поддерева значений.



- *Preprocessing* (при необходимости): если элемент данных возвращает накопительный счётчик (например, трафик интерфейса), перейдите на вкладку *Preprocessing* и добавьте шаг предобработки, например «Изменение в секунду», чтобы вычислить скорость.



Чтобы получить несколько значений в одной SNMP-транзакции, можно указать несколько OID, используя синтаксис `walk[OID1,OID2,...]`.

Преобразование OID между числовым форматом и именами MIB При работе с SNMP может потребоваться преобразование между числовыми OID и соответствующими им именами MIB. Такое преобразование помогает проще идентифицировать метрики и устранять неполадки.

- Преобразование имени MIB в числовой OID: используйте команду `snmptranslate` с опцией `-On`. Например, чтобы преобразовать 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
```

- Преобразование числового OID в имя MIB: используйте команду `snmptranslate` с опцией `-IR` (или `-m ALL`) для обратного преобразования. Например, чтобы преобразовать числовой OID `.1.3.6.1.2.1.31.1.1.1.6.3` обратно в имя MIB, выполните:

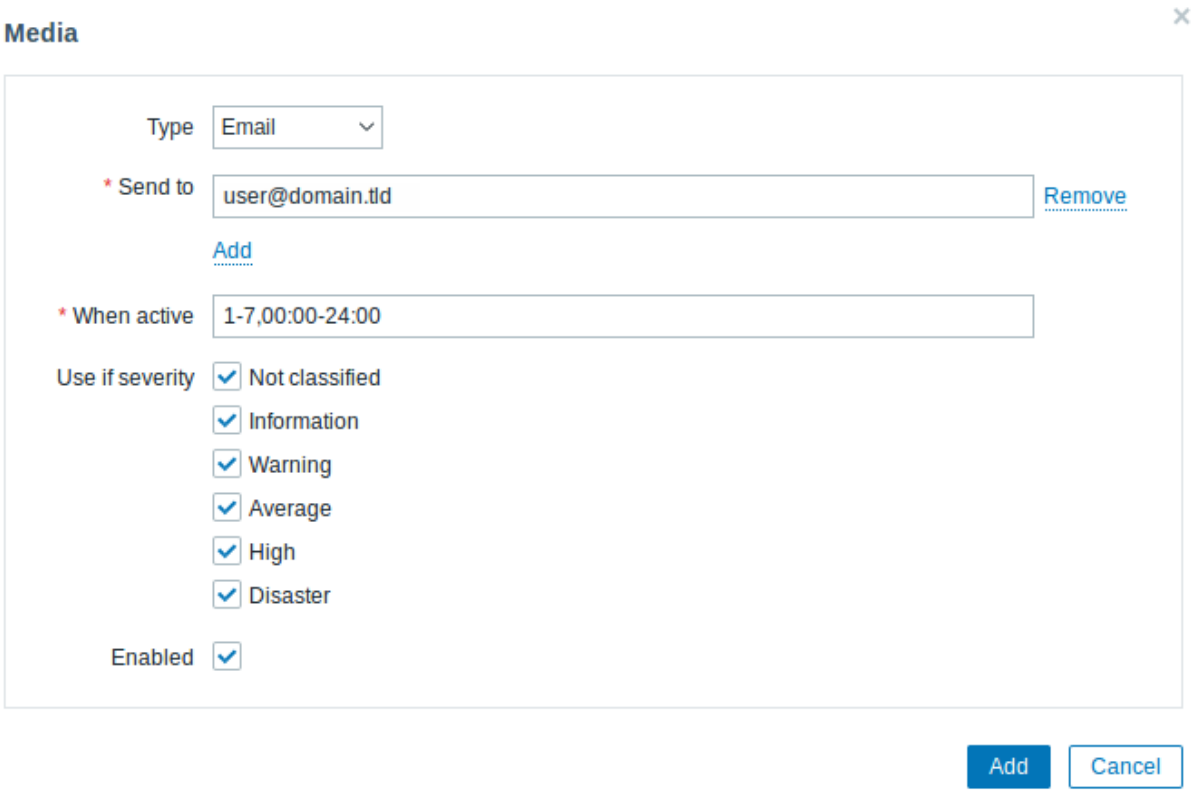
```
snmptranslate -IR -On .1.3.6.1.2.1.31.1.1.1.6.3
```

Эта команда может вывести:

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

Настройка оповещений о проблемах Это руководство содержит основные шаги по настройке отправки оповещений по электронной почте.

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



2. Следуйте руководству Получение уведомления о проблеме.

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

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

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

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

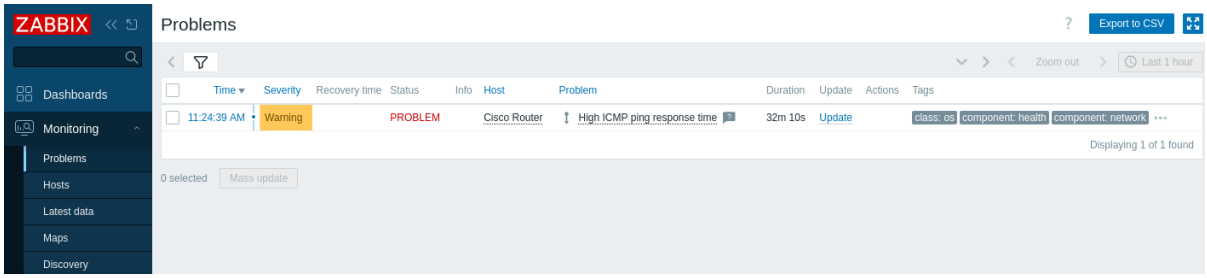
3. Найдите макрос `{$ICMP_RESPONSE_TIME_WARN}` (или похожий макрос порога времени отклика).

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 немного, 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 исходное значение макроса и нажмите *Update*, чтобы сохранить изменения.

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

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

1. Проверьте доступность SNMP.

Для SNMPv2 выполните следующую команду на вашем сервере 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 позволяет убедиться, что учетные данные SNMPv3 указаны верно и устройство безопасно отвечает на запросы.

2. Убедитесь, что файлы MIB установлены и включены, как описано в разделе предварительных требований. Чтобы это проверить, следующая команда не должна выдавать ошибку при опросе сетевого устройства:

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

В результате должны возвращаться преобразованные OID без ошибок.

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

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

Это позволяет убедиться, что строка community (для SNMPv2) или сведения о пользователе SNMPv3 указаны правильно.

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 межсетевые экраны или сетевые проблемы не блокируют SNMP-трафик (обычно на порту 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 межсетевого экрана:

```
sudo ufw status
```

Или, для iptables:

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

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

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

`tail -f` позволяет отслеживать обновления журнала в режиме реального времени.

См. также:

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

11 Мониторинг журнала событий Windows с использованием активных проверок

Введение

В этом руководстве объясняется, как отслеживать журналы событий Windows в Zabbix с помощью активных проверок. Используя специфичные для Windows ключи элементов данных Zabbix, вы можете собирать и анализировать критические события (например, неудачные попытки входа в систему, системные ошибки и т. д.) в режиме реального времени.

Для кого предназначено это руководство

Это руководство предназначено для новых пользователей Zabbix и сетевых администраторов, которые хотят отслеживать журналы событий Windows. Дополнительные параметры настройки см. в документации специфичных для Windows ключей элементов данных.

Предварительные требования

Прежде чем продолжить, необходимо загрузить и установить сервер Zabbix и веб-интерфейс Zabbix в соответствии с инструкциями для вашей ОС. Также необходимо загрузить и установить агент Zabbix на компьютере с Windows, который вы хотите отслеживать.

Настройка агента Zabbix для мониторинга журнала событий Windows

1. Откройте `zabbix_agentd.conf` (путь по умолчанию: `C:\Program Files\Zabbix Agent\zabbix_agentd.conf`) на вашем узле сети Windows и убедитесь, что параметр ServerActive установлен в IP-адрес вашего сервера Zabbix, а параметр Hostname соответствует имени узла сети, которое будет задано в веб-интерфейсе Zabbix. Это позволяет агенту запрашивать активные проверки для своего узла сети и у указанного сервера Zabbix. Например:

```
ServerActive=192.0.2.0
Hostname=MyWindowsHost
```

2. Перезапустите службу агента Zabbix, чтобы применить изменения:

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

3. Убедитесь, 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:

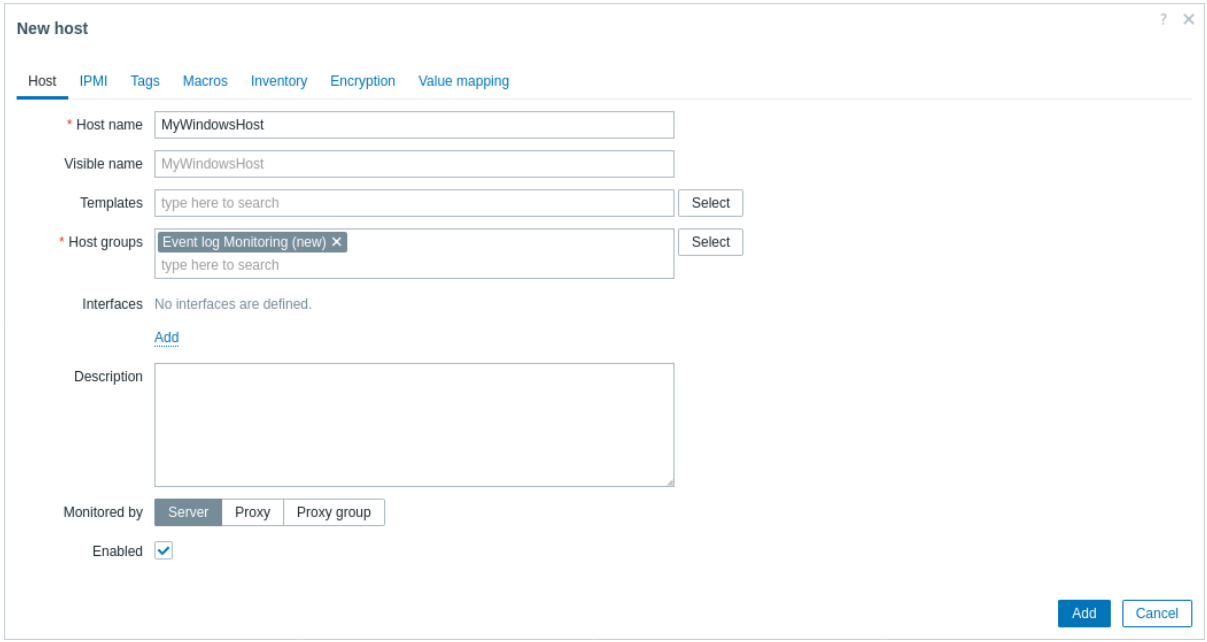
```
Test-NetConnection -ComputerName <Zabbix-server-IP> -Port 10051
```

Настройка веб-интерфейса Zabbix

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

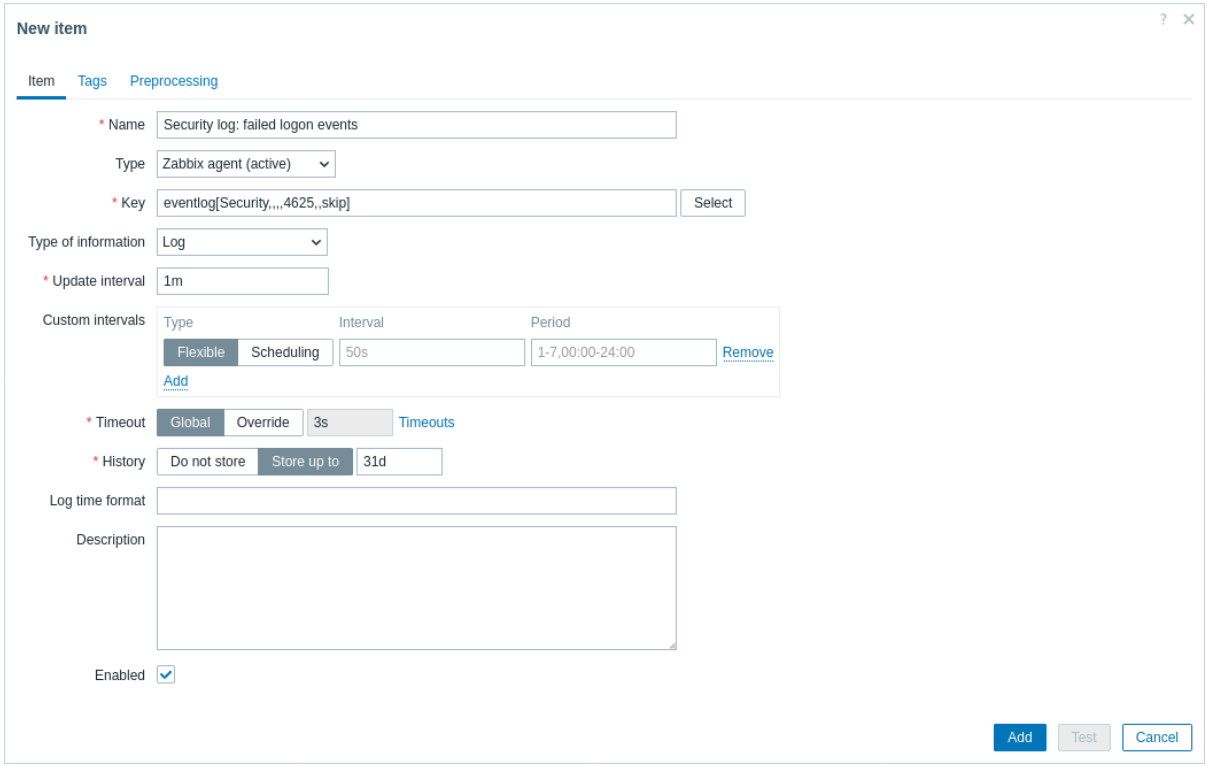
- В поле *Имя узла сети* введите имя узла сети (например, ”MyWindowsHost”).
- В поле *Группы узлов сети* введите или выберите группу узлов сети (например, ”Event log Monitoring”).
- Нажмите *Добавить*, чтобы сохранить настроенный узел сети.

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



2. Создайте новый элемент данных со следующими параметрами:

- В поле *Имя* введите понятное имя элемента данных (например, ”Security log: failed logon events”).
- В раскрывающемся списке *Тип* выберите ”Zabbix агент (активный)” (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 старше последней проверки элемента данных (с использованием параметра `skip`), введите следующий ключ элемента данных: `eventlog[Security,,,,4625,,skip]`
- В раскрывающемся списке *Тип информации* выберите ”Журнал”.



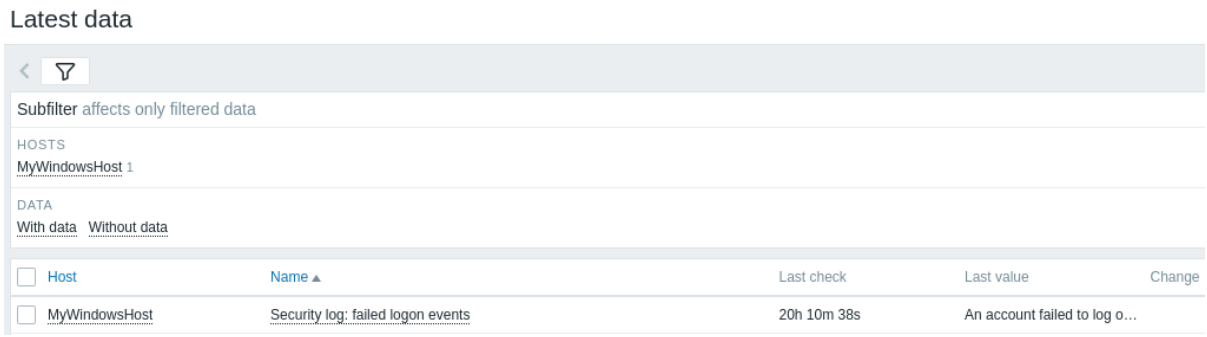
3. Нажмите *Добавить*, чтобы сохранить элемент данных.

Проверка и просмотр собранных метрик

Поздравляем! Теперь Zabbix настроен на сбор журналов событий Windows. Чтобы убедиться, что журналы событий собираются, вы можете проверить элемент данных ”Журнал безопасности: события неудачного входа”, выйдя из своей учетной записи Windows и попытавшись войти с неверными учетными данными.

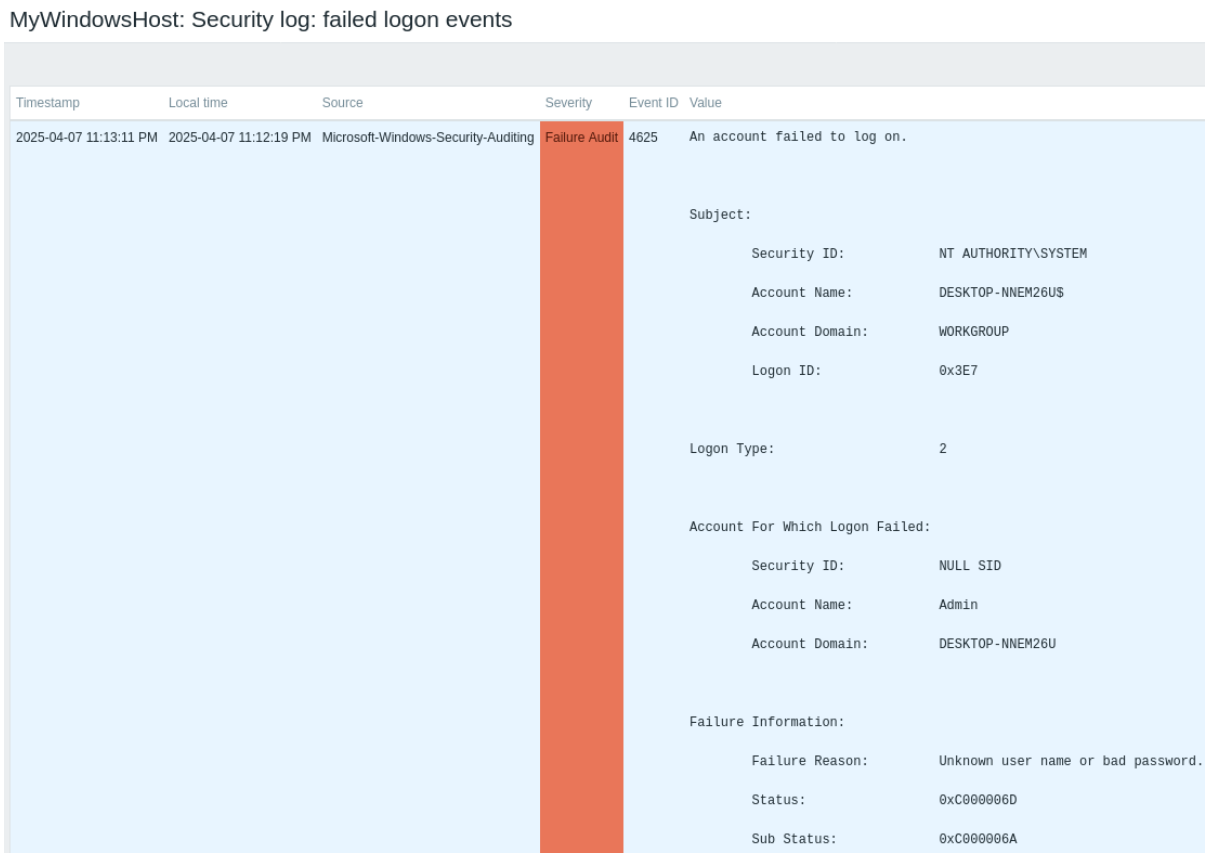
Затем просмотрите собранные журналы в веб-интерфейс Zabbix:

1. Перейдите в *Monitoring* > *Latest data* в веб-интерфейс 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 руководства Устранение неполадок.

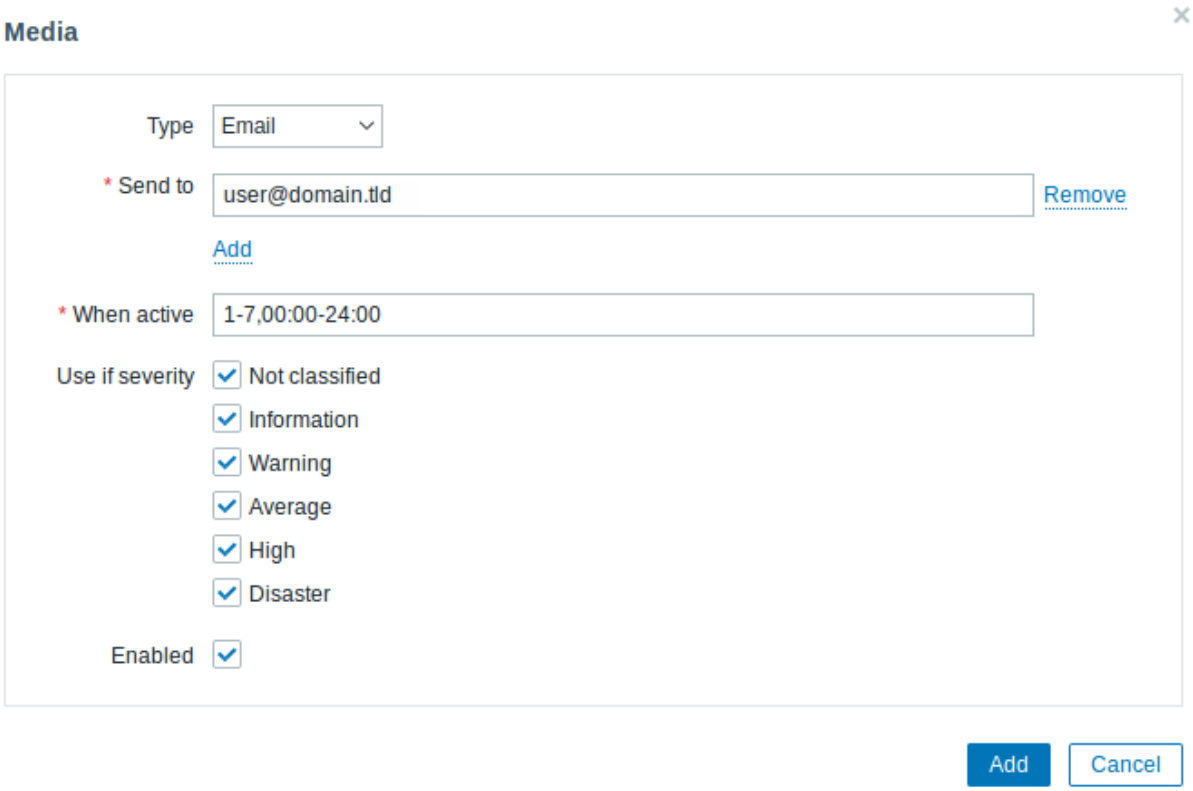
Настройка оповещений о проблемах

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

1. Перейдите в Data collection > Hosts, чтобы создать триггер, который сработает, когда ваш элемент данных журнала событий зафиксирует нужный вам 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 адрес электронной почты.



3. Следуйте руководству Получение 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 приведёнными ниже советами, чтобы выявить и устранить распространённые проблемы:

1. На сервере Zabbix (Linux) выведите список правил iptables следующей командой:

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

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

2. Убедитесь, что ваш ключ `eventlog[...]` использует точное имя журнала (с учётом регистра), ID события, режим (например, skip) и другие параметры в точности так, как показано в разделе Специфичные для Windows ключи элементов данных.

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.
- Специфичные для Windows ключи элементов данных — подробная информация о поддерживаемых агентами Zabbix ключах элементов данных для Windows, включая ключи для мониторинга журнала событий.
- Мониторинг лог-файлов — инструкции по настройке Zabbix для централизованного мониторинга и анализа лог-файлов, применимые к журналам событий Windows.

Что такое Zabbix

Zabbix был создан Алексеем Владишевым и в настоящее время активно развивается и поддерживается компанией Zabbix SIA.

Zabbix — это распределённое решение корпоративного класса для мониторинга с открытым исходным кодом.

Zabbix — это программное обеспечение, которое отслеживает многочисленные параметры сети, а также состояние и целостность серверов, виртуальных машин, приложений, сервисов, баз данных, веб-сайтов, облачной инфраструктуры и многого другого. Zabbix использует гибкий механизм уведомлений, который позволяет пользователям настраивать

оповещения по электронной почте практически для любого события. Это позволяет быстро реагировать на проблемы сервера. Zabbix предлагает отличные возможности создания отчётов и визуализации данных на основе сохранённых данных. Это делает Zabbix идеальным решением для планирования мощностей.

Zabbix поддерживает как опрос, так и получение данных в режиме trapper. Все отчёты и статистика Zabbix, а также параметры конфигурации, доступны через веб-интерфейс. Веб-интерфейс обеспечивает возможность оценивать состояние вашей сети и работоспособность ваших серверов из любого места. При правильной настройке Zabbix может играть важную роль в мониторинге IT-инфраструктуры. Это в равной степени справедливо как для небольших организаций с несколькими серверами, так и для крупных компаний с множеством серверов.

Zabbix распространяется бесплатно. Zabbix написан и распространяется по лицензии AGPL-3.0. Это означает, что его исходный код свободно распространяется и доступен широкой общественности.

Доступна коммерческая поддержка, предоставляемая компанией Zabbix и её партнёрами по всему миру.

Подробнее о возможностях Zabbix.

Пользователи Zabbix Многие организации разного масштаба по всему миру используют Zabbix в качестве основной платформы мониторинга.

Архитектура Zabbix состоит из нескольких основных программных компонентов. Их назначение описано ниже.

Сервер

Сервер Zabbix — это центральный компонент, на который агенты отправляют информацию о доступности, целостности и статистике. Сервер является центральным хранилищем, в котором сохраняются все данные конфигурации, статистики и эксплуатации.

Хранение базы 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. Он 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 разворачиваются на объектах мониторинга для активного мониторинга локальных ресурсов и приложений и передачи собранных данных на Zabbix сервер. Начиная с Zabbix 4.4, доступны два типа агентов: Zabbix agent (легковесный, поддерживается на многих платформах, написан на C) и Zabbix agent 2 (дополнительно гибкий, легко расширяется 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 перейти к другой стороне спектра Zabbix, то для создания триггера сначала необходимо иметь элемент данных. Для создания действия необходим триггер. Таким образом, если вы хотите получать оповещение о том, что загрузка CPU слишком высока на *Server X*, сначала необходимо создать запись узла сети для *Server X*, затем элемент данных для мониторинга его CPU, затем триггер, который срабатывает, если загрузка CPU слишком высока, и после этого действие, которое отправит вам электронное письмо. Хотя это может показаться большим количеством шагов, при использовании шаблонов это совсем не так. Однако благодаря такой архитектуре 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 trapper, так и polling), IPMI, JMX, мониторинга VMware
- пользовательские проверки
- сбор необходимых данных с произвольными интервалами
- выполняется сервером/прокси и агентами

Гибкое определение порогов

- вы можете определять очень гибкие пороги проблем, называемые триггерами, с использованием значений из внутренней базы данных

Широкие возможности настройки оповещений

- отправку уведомлений можно настраивать по расписанию эскалации, получателю, типу медиа
- уведомления можно сделать содержательными и полезными с помощью макропеременных
- автоматические действия включают удалённые команды

Построение графиков в реальном времени

- отслеживаемые элементы данных немедленно отображаются на графиках с использованием встроенной функциональности построения графиков

Возможности веб-мониторинга

- Zabbix может отслеживать последовательность имитируемых щелчков мышью на веб-сайте и проверять функциональность и время отклика

Широкие возможности визуализации

- возможность создавать пользовательские графики, объединяющие несколько элементов данных в одном представлении
- карты сети
- слайд-шоу в стиле панели мониторинга
- отчёты
- высокоуровневое (бизнес-) представление отслеживаемых ресурсов

Хранение исторических данных

- данные хранятся в базе данных
- настраиваемая история
- встроенная процедура очистки устаревших данных

Простая настройка

- добавление отслеживаемых устройств как узлов сети
- узлы сети начинают отслеживаться после добавления в базу данных
- применение шаблонов к отслеживаемым устройствам

Использование шаблонов

- группировка проверок в шаблонах
- шаблоны могут наследовать другие шаблоны

Обнаружение сети

- автоматическое обнаружение сетевых устройств
- авторегистрация агента
- обнаружение файловых систем, сетевых интерфейсов и SNMP OID

Быстрый веб-интерфейс

- веб-интерфейс на PHP
- доступен откуда угодно
- удобная навигация по интерфейсу
- журнал аудита

Zabbix API

- Zabbix API предоставляет программируемый интерфейс к Zabbix для массовых операций, интеграции со сторонним программным обеспечением и других целей.

Система прав доступа

- безопасная аутентификация пользователей
- для определённых пользователей можно ограничить доступ определёнными представлениями

Полнофункциональный и легко расширяемый агент

- развёртывается на целевых системах мониторинга
- может быть развёрнут как на Linux, так и на Windows

Двоичные демоны

- написаны на C для высокой производительности и малого потребления памяти

- легко портируются

Готовность к сложным средам

- удалённый мониторинг упрощается благодаря использованию 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

- логическая группировка узлов сети. Группы узлов сети используются при назначении прав доступа к узлам сети для различных групп пользователей.

элемент данных

- определённая единица данных, которую вы хотите получать от узла сети, 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 'Problem'. Когда полученные данные ниже порога, триггеры остаются в состоянии/возвращаются в состояние 'Ok'.

шаблон

- набор сущностей (элементов данных, триггеров, графиков, правил обнаружения низкого уровня, веб-сценариев), готовый к применению к одному или нескольким узлам сети.

Назначение шаблонов — ускорить развертывание задач мониторинга на узле сети, а также упростить массовое внесение изменений в задачи мониторинга. Шаблоны привязываются непосредственно к отдельным узлам сети.

группа шаблонов

- логическая группировка шаблонов. Группы шаблонов используются при назначении прав доступа к шаблонам для различных групп пользователей.

событие

- единичный случай чего-либо, заслуживающего внимания, например изменение состояния триггера или выполнение обнаружения/авторегистрации агента.

тег события

- предопределённая метка для события. Она может использоваться в корреляции событий, детализации прав доступа и т. д.

корреляция событий

- метод гибкого и точного сопоставления проблем с их устранением.

Например, вы можете определить, что проблема, о которой сообщил один триггер, может быть устранена другим триггером, который может даже использовать другой метод сбора данных.

проблема

- триггер, находящийся в состоянии "Problem".

обновление проблемы

- параметры управления проблемой, предоставляемые Zabbix, такие как добавление комментария, подтверждение, изменение важности или ручное закрытие.

действие

- предопределённый 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 удалённых команд.

макрос

- заполнитель переменной, который разрешается в конкретное значение в зависимости от контекста.

способ оповещения

- средство доставки уведомлений; канал доставки.

уведомление

- сообщение о каком-либо событии, отправленное пользователю через выбранный канал оповещения.

удалённая команда

- предопределённая команда, которая автоматически выполняется на контролируемом узле сети при выполнении некоторого условия.

веб-сценарий

- один или несколько HTTP-запросов для проверки доступности веб-сайта.

веб-интерфейс

- веб-интерфейс, поставляемый с Zabbix.

панель мониторинга

- настраиваемый раздел веб-интерфейса, отображающий сводки и визуализации важной информации в визуальных блоках, называемых виджетами.

виджет

- визуальный блок, отображающий информацию определённого вида и источника (сводку, карту, график, часы и т. д.), используемый на панели мониторинга.

Zabbix API

- Zabbix API позволяет использовать протокол JSON RPC для создания, обновления и получения объектов Zabbix (таких как узлы сети, элементы данных, графики и другие) или выполнения любых других пользовательских задач.

сервер Zabbix

- центральный процесс программного обеспечения Zabbix, который выполняет мониторинг, взаимодействует с прокси Zabbix и агентами, вычисляет триггеры, отправляет уведомления; центральное хранилище данных.

прокси Zabbix

- процесс, который может собирать данные от имени сервера Zabbix, снимая часть нагрузки по обработке с сервера.

агент Zabbix

- процесс, развернутый на объектах мониторинга для активного мониторинга локальных ресурсов и приложений.

агент Zabbix 2

- новое поколение агента Zabbix для активного мониторинга локальных ресурсов и приложений, позволяющее использовать пользовательские плагины для мониторинга.

> **Attention:**
> Поскольку агент Zabbix 2 разделяет значительную часть функциональности с агентом Zabbix, термин ”агент Zabbix” в документации обозначает оба варианта — агент Zabbix и агент Zabbix 2, если функциональное поведение одинаково. Агент Zabbix 2 отдельно упоминается только там, где его функциональность отличается.

шифрование

- поддержка шифрованного обмена данными между компонентами Zabbix (сервер, прокси, агент, утилиты zabbix_sender и zabbix_get) с использованием протокола Transport Layer Security (TLS).

авторегистрация агента

- автоматизированный процесс, при котором сам агент Zabbix регистрируется как узел сети и начинает мониторинг.

обнаружение сети

- *автоматическое обнаружение сетевых устройств*.

обнаружение низкого уровня

- *автоматическое обнаружение низкоуровневых сущностей на конкретном устройстве* (например, файловых систем, сетевых интерфейсов и т. д.).

правило обнаружения низкого уровня

- *набор определений для автоматического обнаружения низкоуровневых сущностей* на устройстве.

прототип элемента данных

- *метрика с определёнными параметрами в качестве переменных, готовая для обнаружения низкого уровня*. После обнаружения низкого уровня переменные автоматически подставляются реальными обнаруженными параметрами, и метрика автоматически начинает собирать данные.

прототип триггера

- *триггер с определёнными параметрами в качестве переменных, готовый для обнаружения низкого уровня*. После обнаружения низкого уровня переменные автоматически подставляются реальными обнаруженными параметрами, и триггер автоматически начинает оценивать данные.

Прототипы некоторых других сущностей Zabbix также используются в обнаружении низкого уровня — прототипы графиков, прототипы узлов сети, прототипы групп узлов сети.

Zabbix Cloud

Zabbix Cloud

Ознакомьтесь с доступными параметрами конфигурации и элементами управления доступом для организаций и узлов в Zabbix Cloud.

Настройка и управление Zabbix Cloud

Начало работы с Zabbix Cloud Пошаговые инструкции, которые помогут быстро запустить ваш экземпляр Zabbix Cloud, включая первоначальную настройку и подключение отслеживаемых 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 сети.

Узнайте о Zabbix Cloud

Zabbix Cloud и локальное развертывание Сравните различия в управлении, масштабируемости и обслуживании между двумя вариантами развертывания и узнайте, какой из них сможет наиболее органично поддержать ваши рабочие процессы.

Знакомство с Zabbix Cloud Получите общее представление о Zabbix Cloud, включая его ключевые возможности, преимущества и то, что входит в подписку. Ответы на распространенные вопросы по настройке и использованию см. в FAQ.

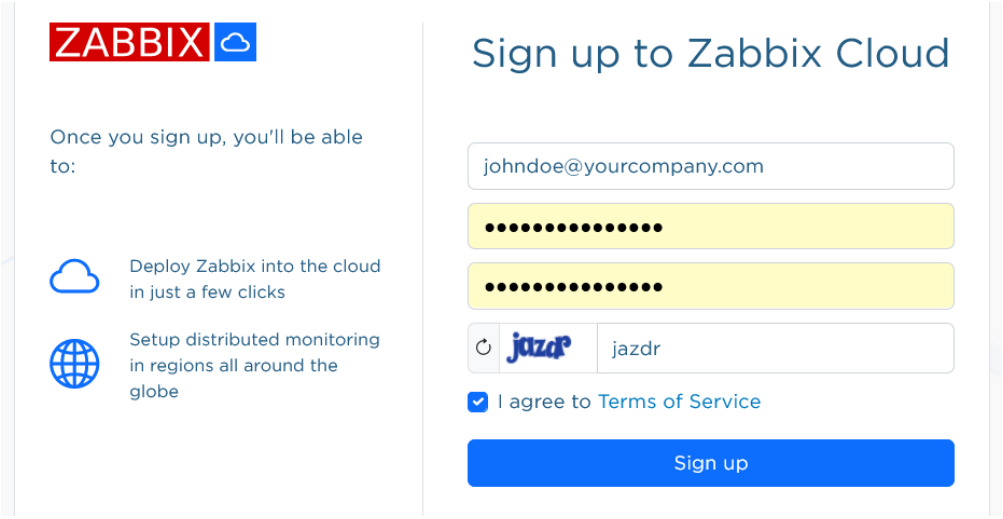
Развертывание Zabbix в облаке

Чтобы начать использовать Zabbix в облаке, выполните следующие шаги:

- Войдите или зарегистрируйтесь на cloud.zabbix.com
- Создайте облачный узел
- Начните использовать Zabbix в облаке

Вход или регистрация Вы можете войти на cloud.zabbix.com, используя существующую учетную запись Google, GitHub или Microsoft.

Либо перейдите на страницу регистрации, чтобы создать учетную запись.



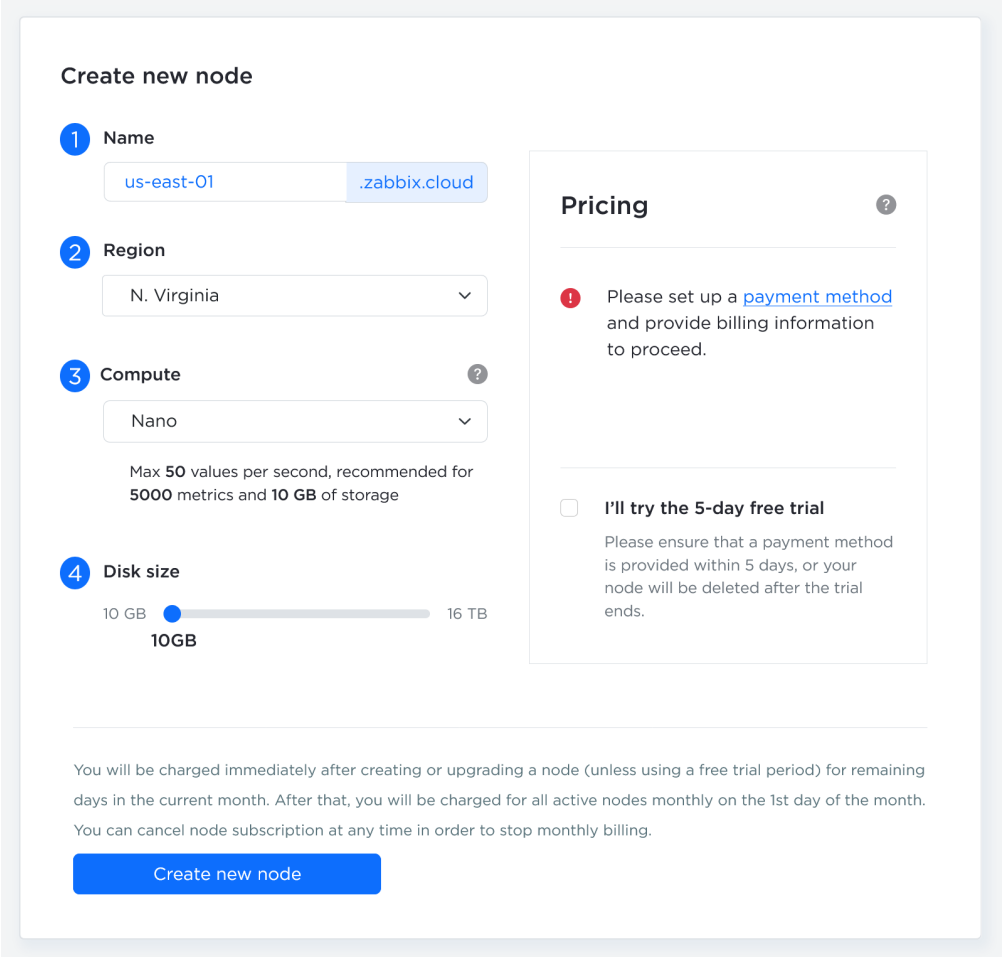
После регистрации вы получите подтверждение создания новой учетной записи по электронной почте.

Перейдите на cloud.zabbix.com, чтобы войти. При входе у вас будет запрошен одноразовый пароль. Этот пароль будет отправлен вам по электронной почте.

Создание облачного узла Облачный узел — это экземпляр Zabbix, размещенный в облаке. Он включает сервер Zabbix, базу данных и веб-интерфейс.

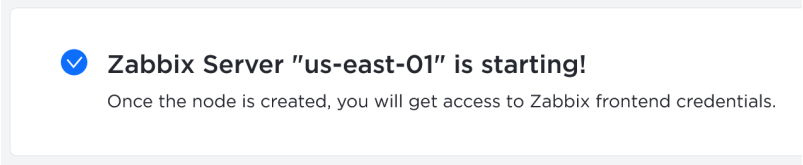
После входа вы увидите, что для вашей учетной записи по умолчанию уже существует организация *Private*. Позже вы сможете переименовать организацию по умолчанию.

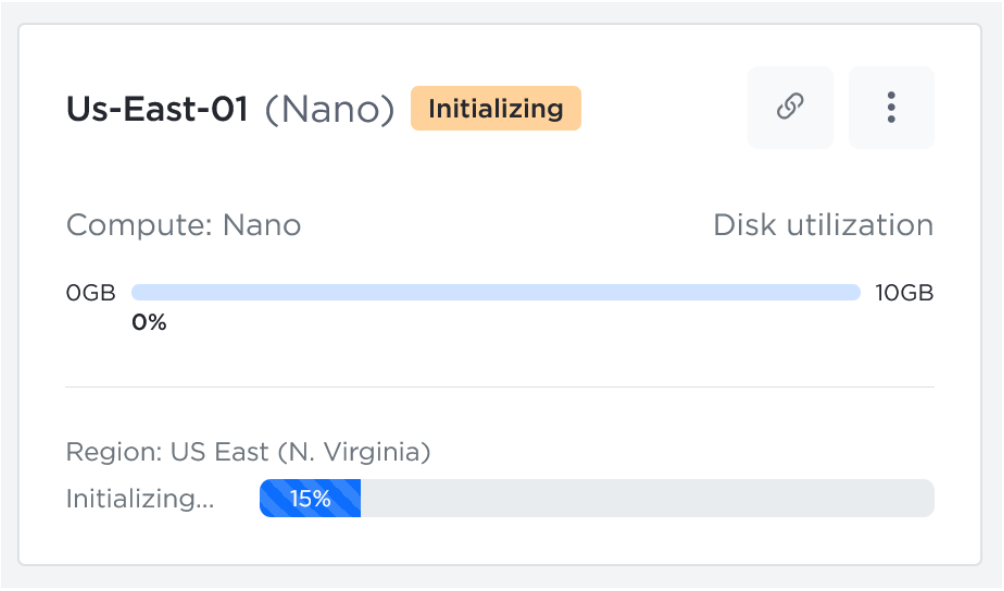
Нажмите *Create New Node*. Откроется форма настройки узла.

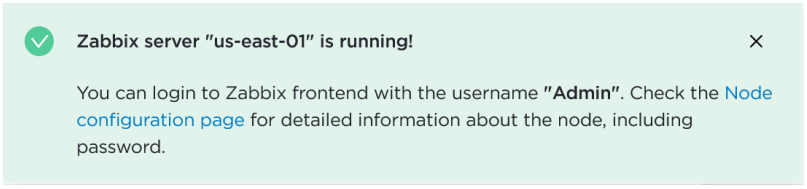


Параметр	Описание
Name	Выберите уникальное имя узла. Имя узла формирует уникальный URL для облачного сервера/веб-интерфейса в формате `<your-name>.zabbix.cloud`.
Region	Выберите ближайшее расположение дата-центра для минимальной сетевой задержки.
Compute	Выберите вычислительную мощность для узла. Чтобы определить, какой уровень ресурсов вам нужен, см. подробности о ценах. При необходимости вы всегда сможете выполнить обновление позже. Перед созданием нетестового узла потребуется указать способ оплаты. Добавьте платежную информацию через меню *Billing*. Для ознакомления с узлом уровня *Nano* доступен **бесплатный пробный период** на 5 дней. В течение пробного периода с вас не будет взиматься плата, однако после его окончания узел будет удален, если его не перевести на платный тариф.
Disk size	По умолчанию ползунок установлен на рекомендуемый минимальный размер в зависимости от уровня. Увеличьте его, если вы храните большие объемы истории или трендов. Обратите внимание, что диски большего размера увеличивают ежемесячную стоимость, а уменьшить размер диска самостоятельно нельзя — для этого потребуется обратиться в службу поддержки.

После заполнения формы нажмите *Create New Node*. Пока узел инициализируется, отображается соответствующее сообщение, а сам узел уже появляется в разделе Organizations.





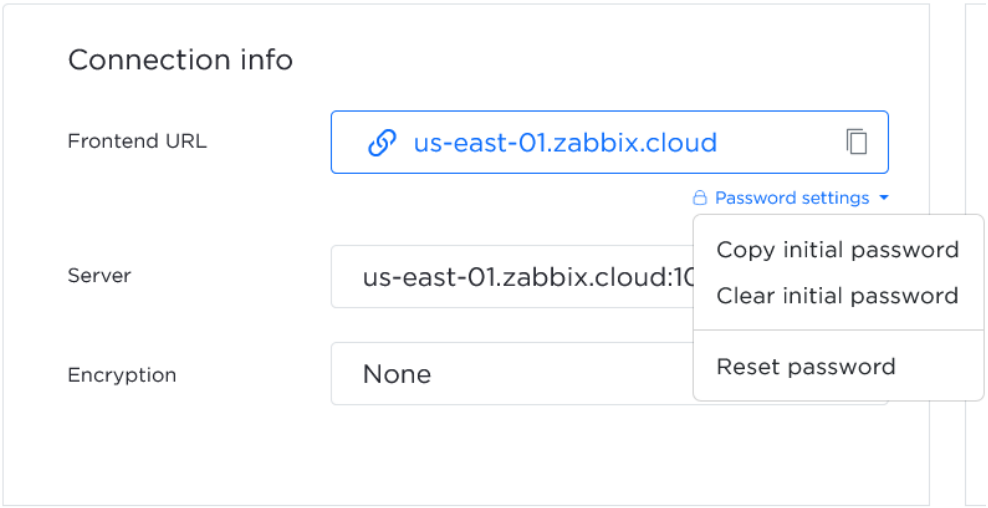


Начало работы с Zabbix

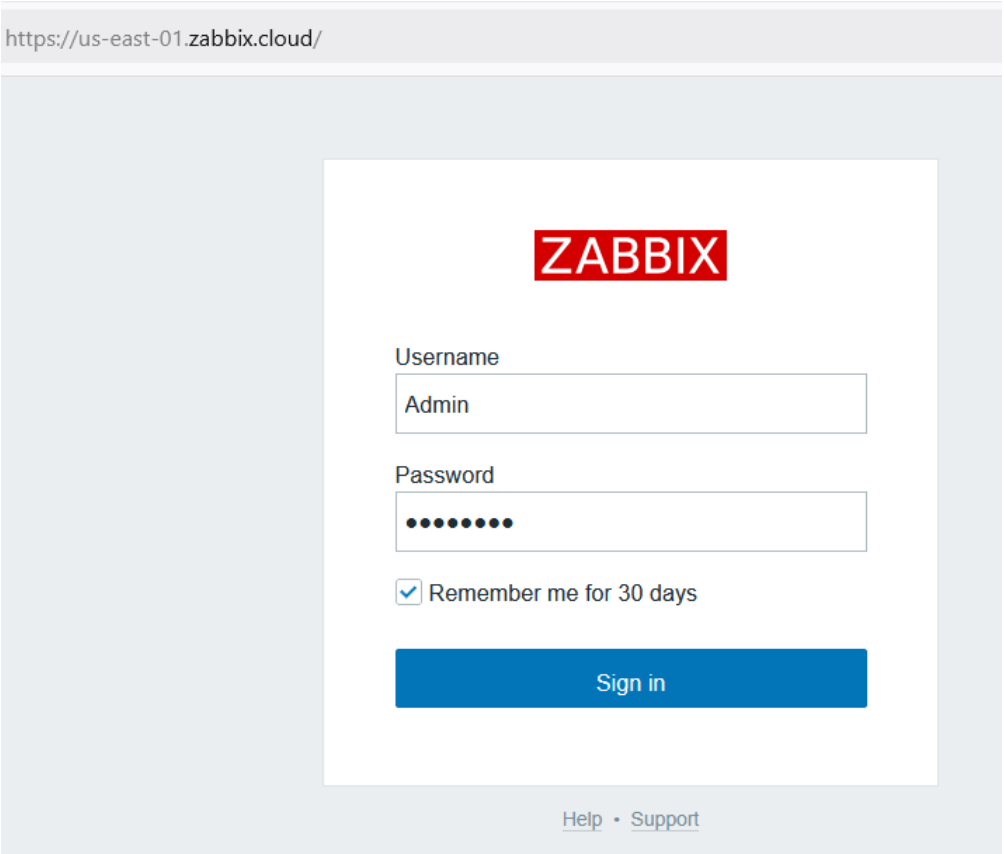
Когда статус узла изменится с *Initializing* на *Running*, вы сможете войти в веб-интерфейс Zabbix.

Чтобы получить начальный пароль веб-интерфейса, нажмите на созданный узел.

Вы будете перенаправлены в настройки узла (обзор, фильтры доступа, шифрование, резервные копии и т. д.). В разделе **Connection info** нажмите *Password settings* и выберите *Copy initial password* из выпадающего списка.



Нажмите URL веб-интерфейса, чтобы открыть веб-интерфейс Zabbix (`us-east-01.zabbix.cloud`).



Войдите, используя следующие учетные данные:

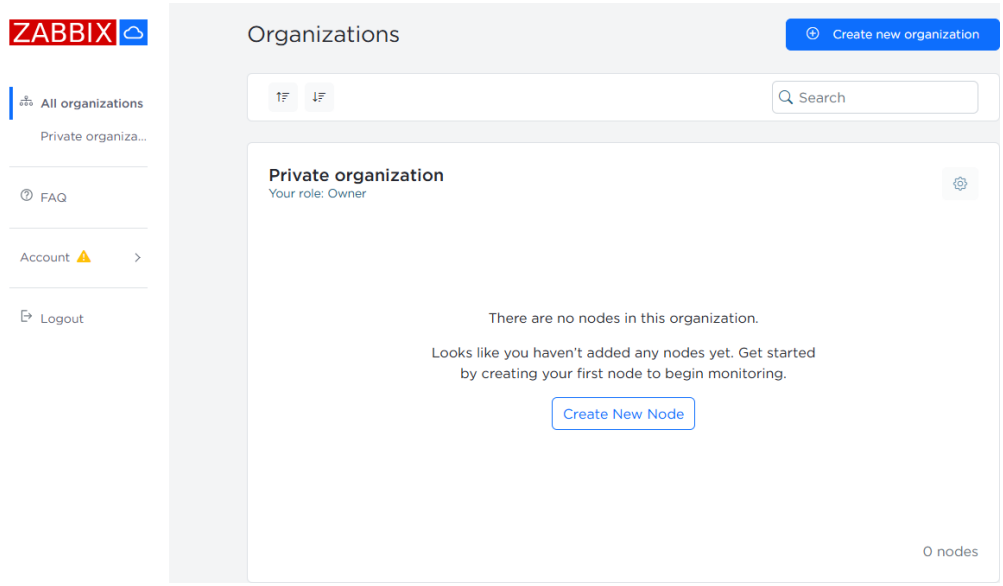
- *Username* - `Admin`
- *Password* - укажите начальный пароль (например, HzTG9t7Y)

Чтобы начать мониторинг, необходимо определить фильтры доступа — IP-адрес или адреса, которым разрешено подключаться к облачному серверу Zabbix. Необходимо определить как минимум один.

Управление организациями Организация — это контейнер для одного или нескольких узлов. Организации также помогают структурировать ваши ресурсы и делегировать обязанности, такие как:

- Управление узлами
- Биллинг

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 и названием организации. Нажмите кнопку настроек , чтобы получить доступ к параметрам управления.

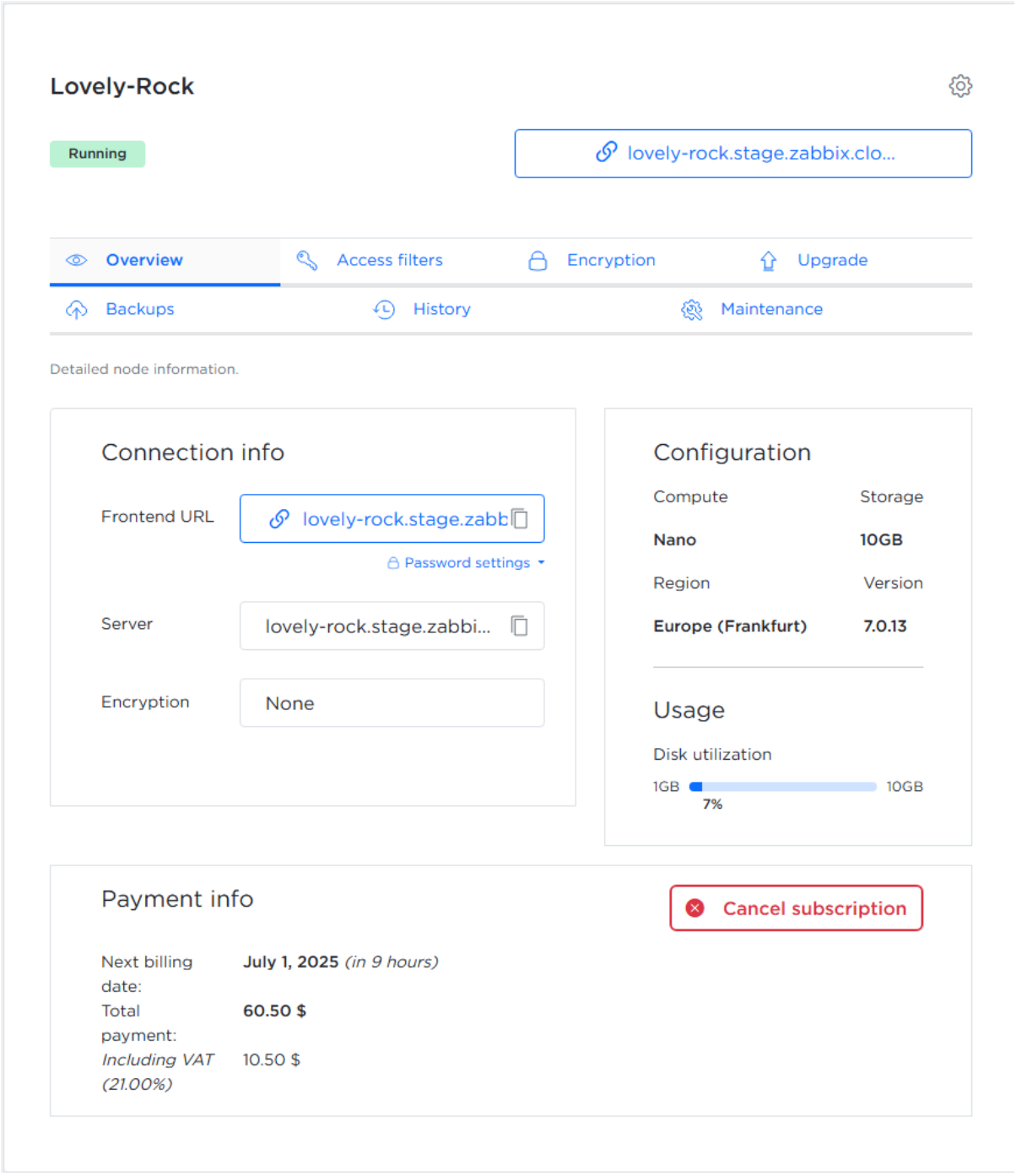
Конфигурация узла

Обзор

Чтобы открыть страницу конфигурации узла, нажмите на узел сети либо в разделе *All organizations*, либо из вашего меню организации, либо по ссылке *Node configuration page*, которая отображается после запуска узла сети.

В представлении конфигурации узла сети каждая вкладка позволяет управлять определённым аспектом вашего узла сети:

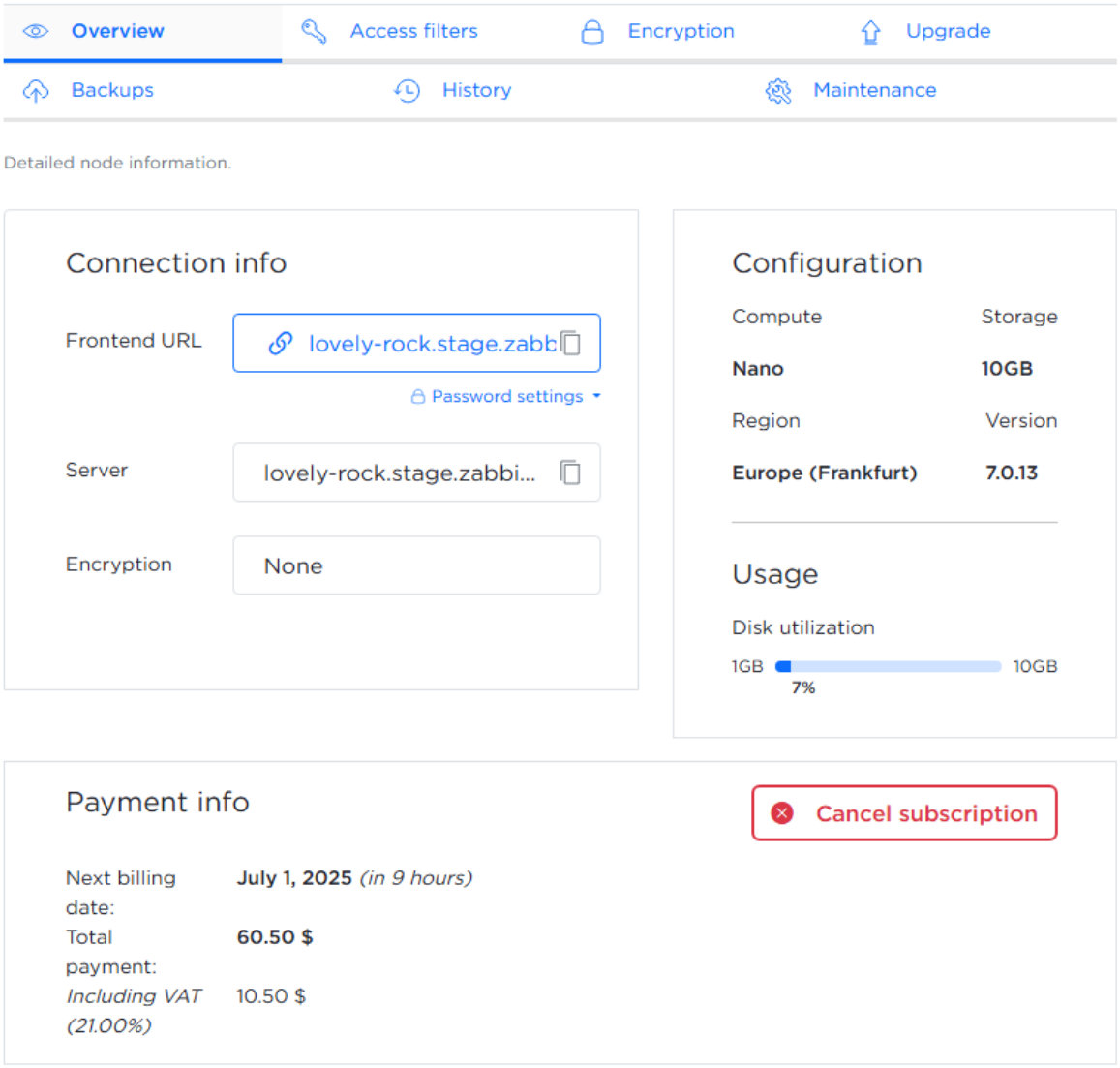
- Overview — просмотр сведений об узле сети, открытие его веб-интерфейса и управление пользователями узла сети.
- Access filters — управление доступом по IP-адресу или подсети.
- Encryption — настройка TLS путём загрузки сертификатов, ключей и необязательных CRL.
- Backups — создание, восстановление и удаление снимков узла сети; просмотр времени следующего запланированного резервного копирования.
- History — настройка сроков хранения журналов аудита, истории и трендов с предустановленными единицами измерения и ограничением обновления 3 раза в день.
- Maintenance — определение еженедельных окон обслуживания по часовому поясу, дню и часу.
- Upgrade — масштабирование вычислительных ресурсов/хранилища или управление статусом пробного периода/оплаты.



Note:
Некоторые вкладки могут быть недоступны в зависимости от типа пользователя, который просматривает страницу, и статуса узла сети (пробный/непробный).

Вкладка «Обзор»

Вкладка *Обзор* предоставляет подробную информацию об узле сети.



Она разделена на три раздела:

Информация о подключении:

- URL веб-интерфейса, по которому можно перейти, с возможностью копирования в буфер обмена
- Настройки пароля
- Текущий метод шифрования

Конфигурация и использование:

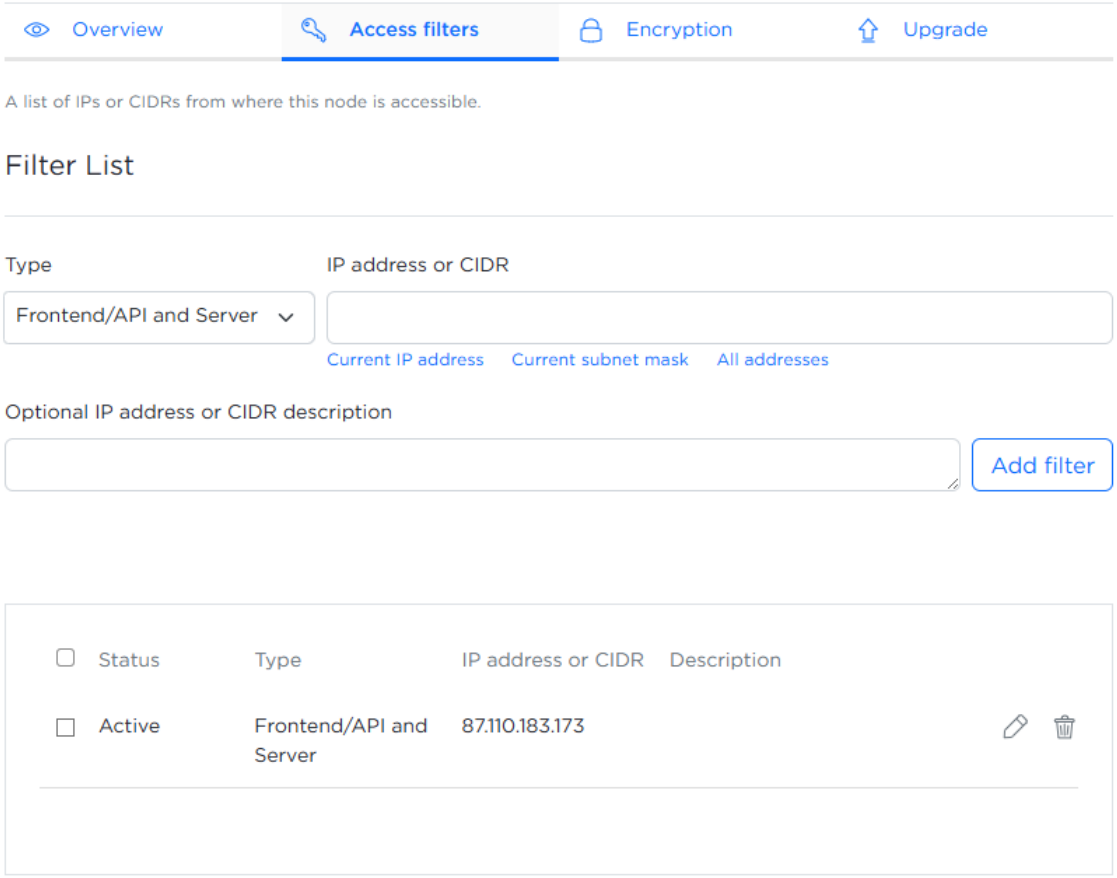
- Уровень вычислительных ресурсов, максимальный объем хранилища, регион
- Версия компонентов Zabbix
- Использование диска

Информация об узле сети/Платежная информация:

- Дата истечения срока действия (для пробных и предоплаченных узлов сети)
- После завершения инициализации узла сети в зависимости от его типа отображается соответствующая кнопка: - *Перейти на платный тариф* — доступно для узлов сети на бесплатном пробном периоде - *Отменить подписку* — доступно для платных узлов сети с активной подпиской - *Повторно оформить платную подписку* — отображается, если подписка была отменена - Для предоплаченных узлов сети кнопка действия не отображается, так как они имеют длительный срок действия и не могут быть отменены

Вкладка «Фильтры доступа»

Вкладка *Фильтры доступа* позволяет ограничить доступ к вашему узлу Zabbix по IP-адресу или подсети.



Чтобы настроить фильтр доступа, заполните следующие поля:

- *Тип* — используйте выпадающий список, чтобы выбрать, какой компонент Zabbix нужно ограничить: только веб-интерфейс/API, только сервер или оба.
- *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, где показаны статус, тип, IP-адрес или подсеть и описание.

Warning:
Во время инициализации узла сети для использованного вами IP-адреса автоматически создается фильтр доступа. Если позже 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 с помощью флажков.

Вкладка Encryption

Вкладка *Encryption* позволяет настроить TLS-шифрование между компонентами Zabbix, загрузив необходимые файлы сертификатов.

Чтобы включить безопасное взаимодействие с использованием сертификатов, выберите *CERTIFICATE* и загрузите следующие файлы с вашего компьютера:

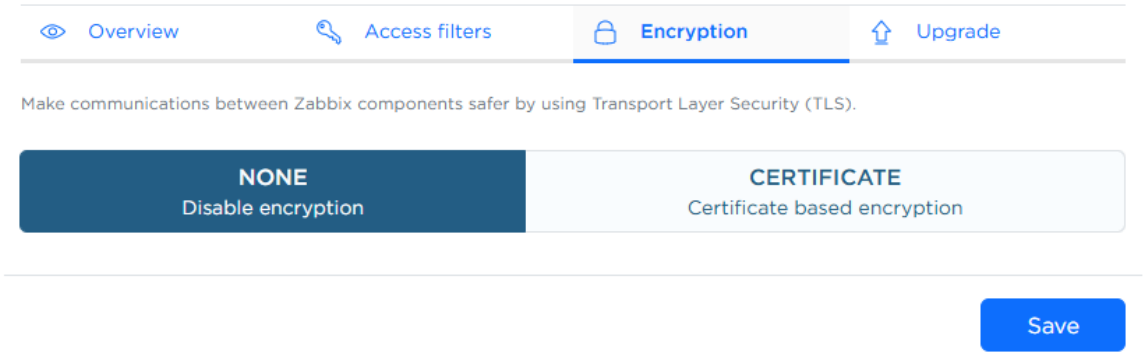
- *Root CA certificates* — используются для проверки подлинности сертификатов другой стороны.
  Нажмите *Choose file* и выберите файл, содержащий доверенные корневые сертификаты CA (обычно файл `.crt` или `.pem` в кодировке PEM).

- *Certificate chain* — представляет цепочку доверия для вашего собственного сертификата.
  Нажмите *Choose file* и выберите файл в кодировке PEM (например, `.crt` или `.pem`), который включает ваш сертификат и все промежуточные сертификаты CA.

- *Private key* — закрытый ключ, соответствующий вашему сертификату.
  Нажмите *Choose file* и выберите файл `.key` в кодировке PEM.

- *Revoked certificates* (необязательно) — если вы используете список отозванных сертификатов (CRL), нажмите *Choose file* и выберите файл CRL, чтобы указать сертификаты, которым больше не следует доверять.

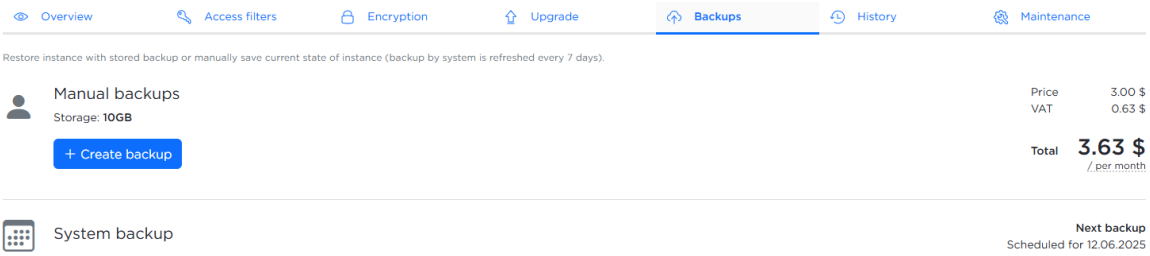
После выбора файлов нажмите *Save*. Файлы будут загружены, проверены и применены для настройки TLS-шифрования между компонентами Zabbix.



Вкладка «Резервные копии»

Вкладка *Резервные копии* позволяет просматривать, создавать, восстанавливать и удалять резервные копии вашего узла Zabbix.

Вы можете восстановить свой экземпляр из любой сохранённой резервной копии или в любое время создать новую резервную копию вручную.
На вкладке также отображается дата следующего запланированного резервного копирования.



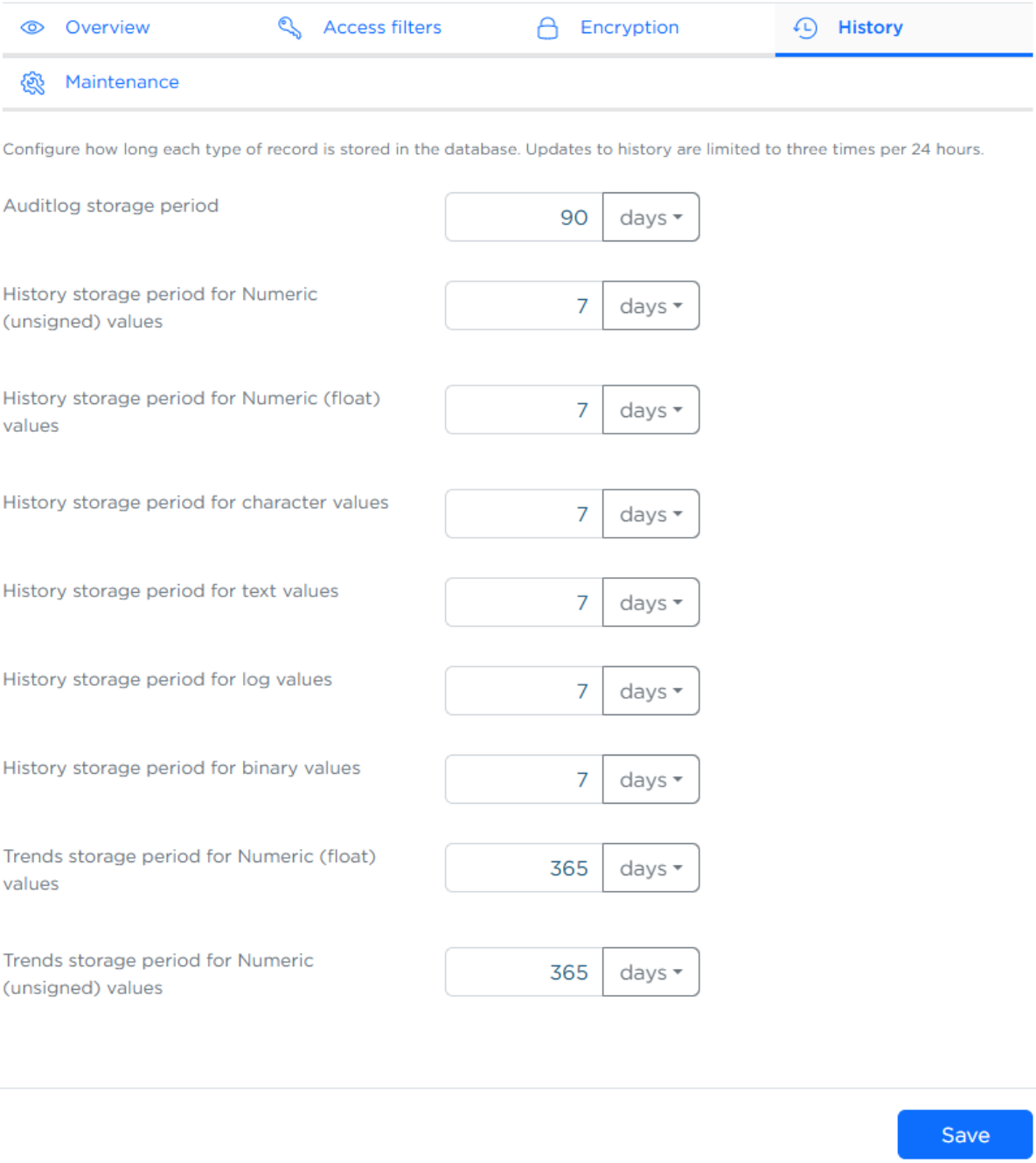
Чтобы создать резервную копию:

1. Нажмите кнопку *Create backup*.
2. Введите необязательное описание во всплывающем окне.
3. Нажмите *Create* для подтверждения.

Warning:
Дополнительные резервные копии оплачиваются ежемесячно в зависимости от объёма используемого хранилища.

Вкладка History

Вкладка *History* позволяет настроить, как долго хранятся различные типы исторических данных.



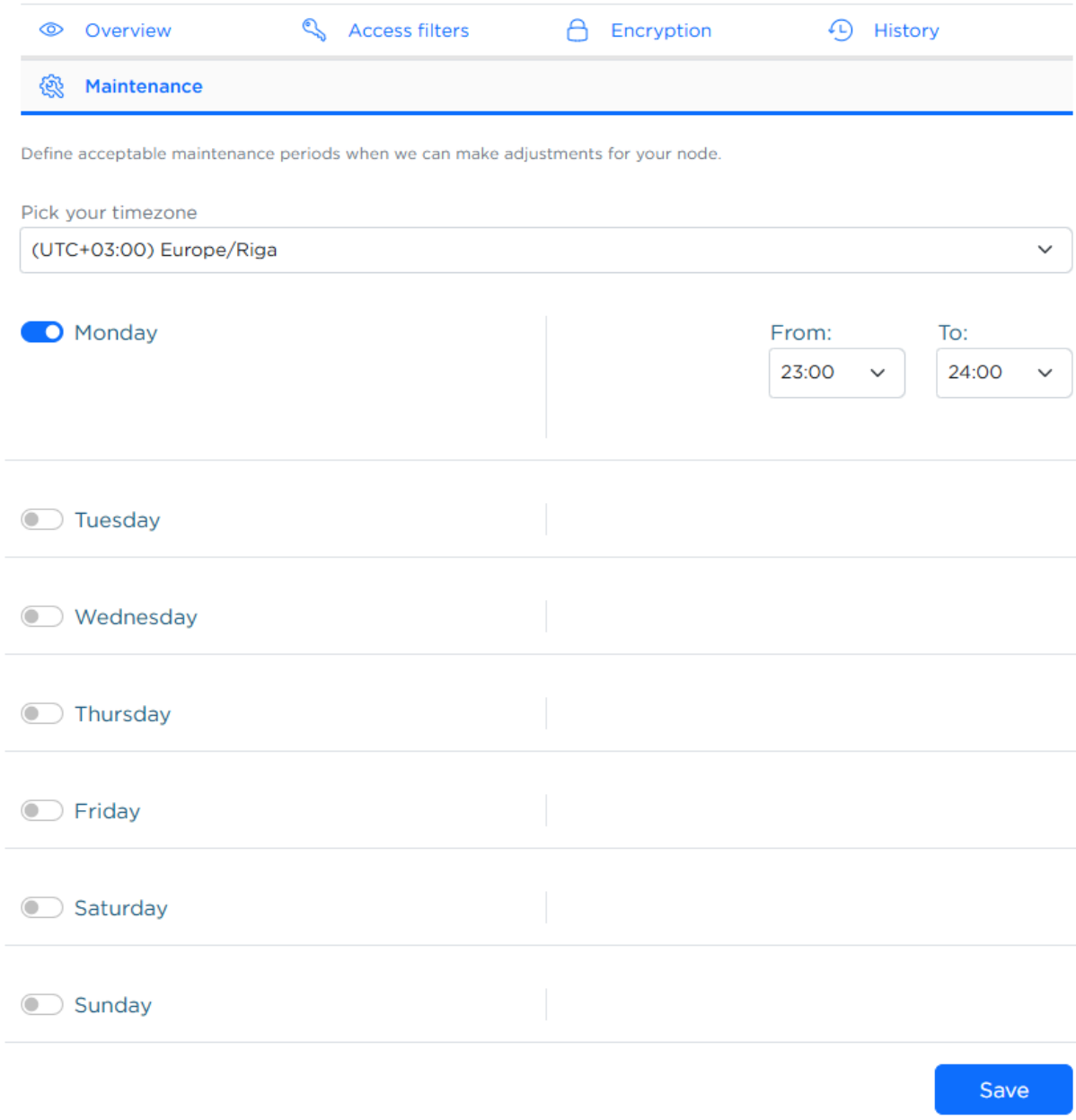
Вы можете задать период хранения для каждого типа данных (например, числовых, текстовых, журналов) и выбрать единицу измерения (дни, месяцы, годы). Изменения можно вносить не более трёх раз в течение каждых 24 часов.

После обновления значений нажмите *Save*, чтобы применить изменения.

Warning:
Более длительные периоды хранения могут увеличить использование дискового пространства и повлиять на производительность. Рекомендуется настраивать периоды хранения в соответствии с вашими фактическими данными и потребностями анализа.
 Zabbix Cloud использует секционирование базы данных для управления хранением данных вместо традиционной очистки устаревших данных Zabbix. Это обеспечивает более высокую производительность и более предсказуемую обработку сроков хранения.

Вкладка «Обслуживание»

Вкладка *Обслуживание* позволяет задать допустимые окна обслуживания, в течение которых на вашем узле могут выполняться обновления или корректировки.

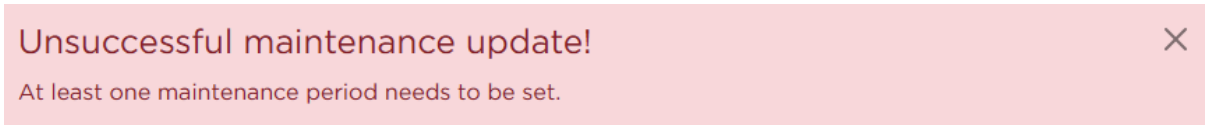


Необходимо:

- Выбрать ваш часовой пояс из выпадающего списка.
- Выбрать как минимум один день и один почасовой интервал для обслуживания.

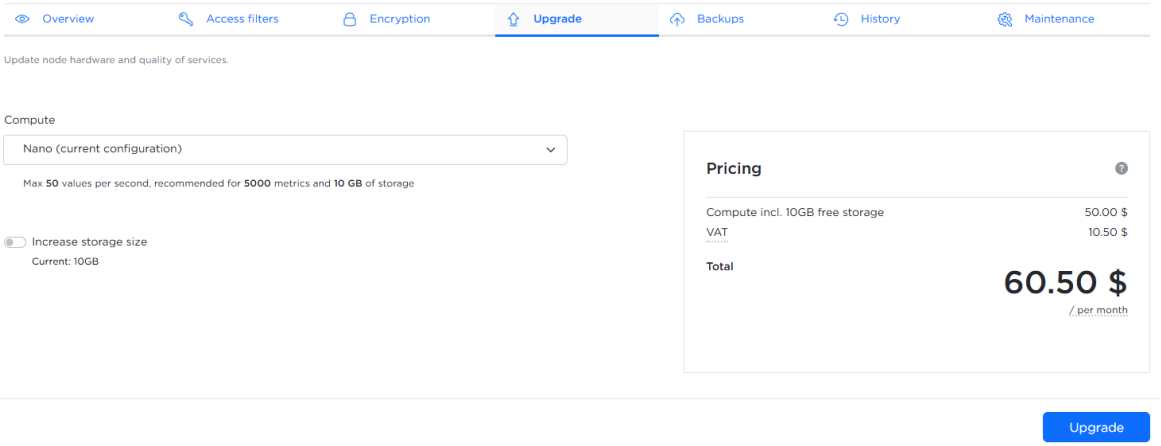
Нажмите *Сохранить*, чтобы применить изменения.

Если не выбран ни день, ни временной интервал, будет отображено сообщение об ошибке.



Вкладка Upgrade

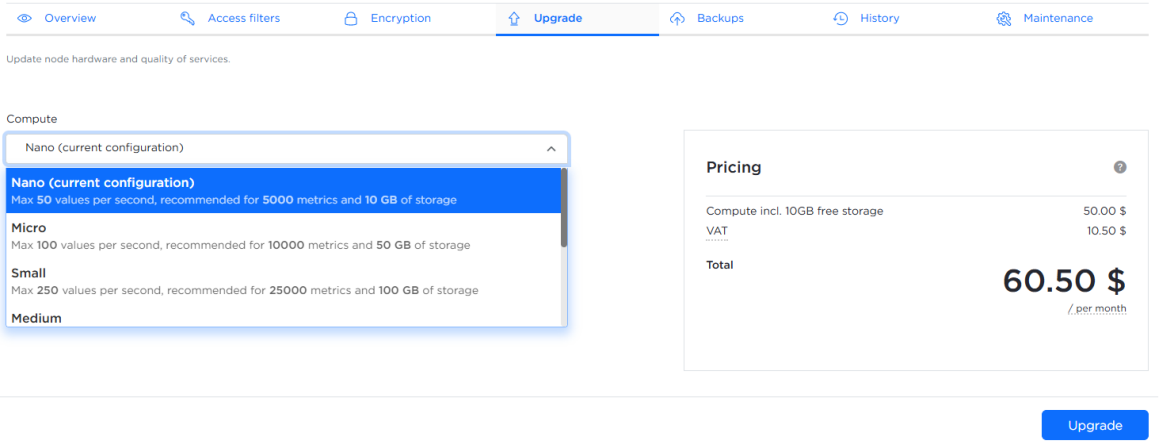
Вкладка *Upgrade* позволяет масштабировать ресурсы вашего узла сети или управлять его подпиской.



Вы можете увеличить уровень вычислительных ресурсов или размер диска — обратите внимание, что размер диска можно увеличивать только один раз в 24 часа, и его нельзя уменьшить через пользовательский интерфейс (для любых запросов на уменьшение обратитесь в службу поддержки Zabbix).
Узлы сети с бесплатным пробным периодом можно перевести на платную подписку.

Чтобы увеличить уровень вычислительных ресурсов:

1. Нажмите на выпадающий список *Compute*.



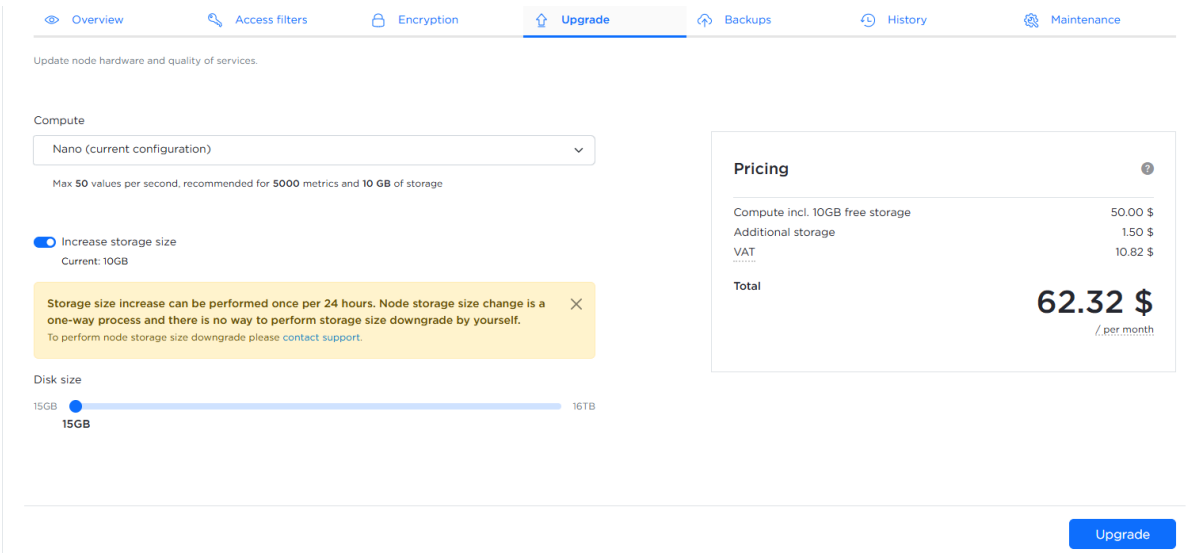
2. Выберите нужную конфигурацию.

3. Нажмите кнопку *Upgrade*.

Чтобы увеличить размер диска:

1. Нажмите переключатель *Increase storage size*.

2. Переместите ползунок *Disk size* на нужное значение.



3. Нажмите кнопку *Upgrade*.

Note:
Обновлённая общая ежемесячная цена отображается в таблице *Pricing* перед подтверждением изменений.

Добавление пользователей

Введение

Следующие примеры описывают, как добавлять пользователей в вашу организацию и узлы.

Note:
Копирование, очистка и сброс пароля веб-интерфейса могут выполняться только владельцем.

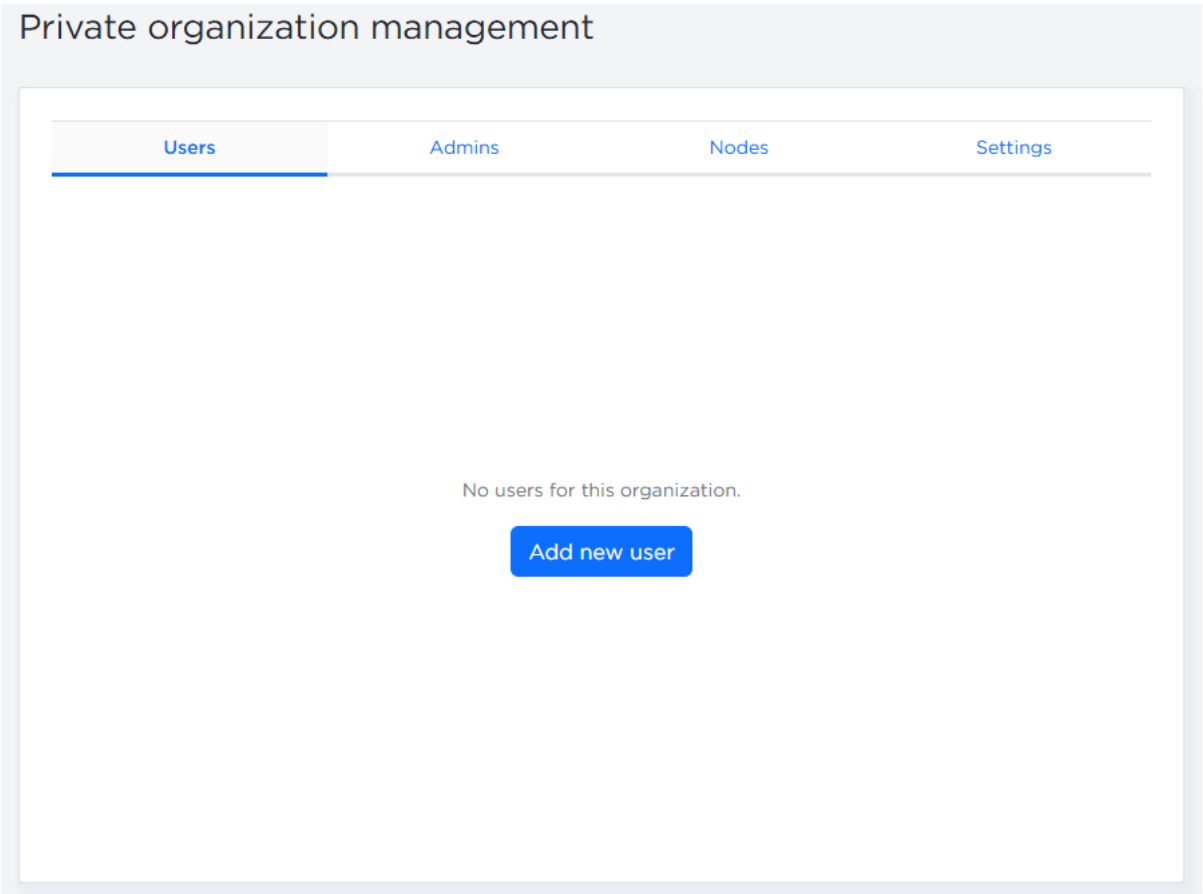
Добавление пользователей в организации

Аналогично пользователям в Zabbix, в Zabbix Cloud можно создавать пользователей и назначать им роли, чтобы управлять тем, что они могут делать в рамках организации и ее узлов сети.

Чтобы добавить пользователей в вашу организацию:

1. Откройте страницу *All organizations*.

2. Нажмите кнопку настроек нужной организации, чтобы перейти в меню управления ею.



3. На вкладке *Users* или *Admins* нажмите *Add new user*.

4. Введите email пользователя и выберите роль: *Administrator* или *User*.

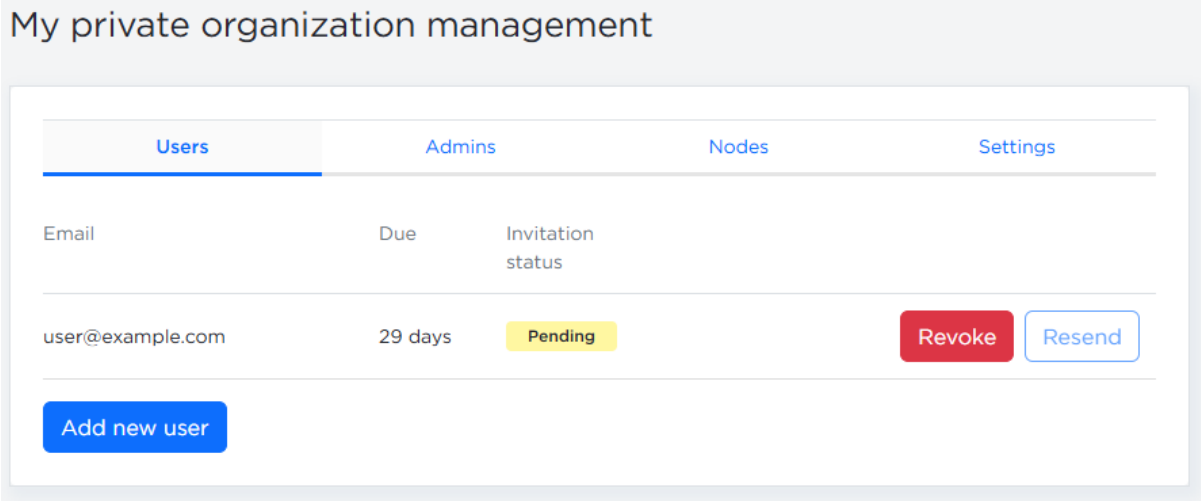
Обратите внимание, что выбор роли *User* добавляет пользователя в организацию, но не предоставляет доступ ни к каким узлам сети.

Каждая роль предоставляет разный уровень доступа:

- **User** — может получать доступ к назначенным ему узлам сети и выполнять задачи настройки, которые не приводят к дополнительным расходам, например:
  - Управлять фильтрами доступа
  - Настраивать шифрование
  - Задавать периоды хранения history/trend
  - Планировать обслуживание
- **Administrator** — имеет все разрешения роли User, а также может выполнять действия, которые могут привести к дополнительным расходам, например:
  - Создавать и восстанавливать резервные копии
  - Увеличивать или уменьшать ресурсы узла сети (уровень вычислительных ресурсов и размер диска)
  - Отменять подписки
- **Owner** — имеет полный контроль над организацией. Помимо всех разрешений роли Administrator, Owner может:
  - Создавать и удалять узлы сети
  - Добавлять или удалять пользователей и администраторов
  - Переименовывать или удалять организацию
  - Управлять платежными данными

> **Attention:**
> Со способа оплаты Owner списывается плата за все услуги. Если требуется подтверждение банком, Owner должен авторизовать транзакцию, даже если она была инициирована Administrator.

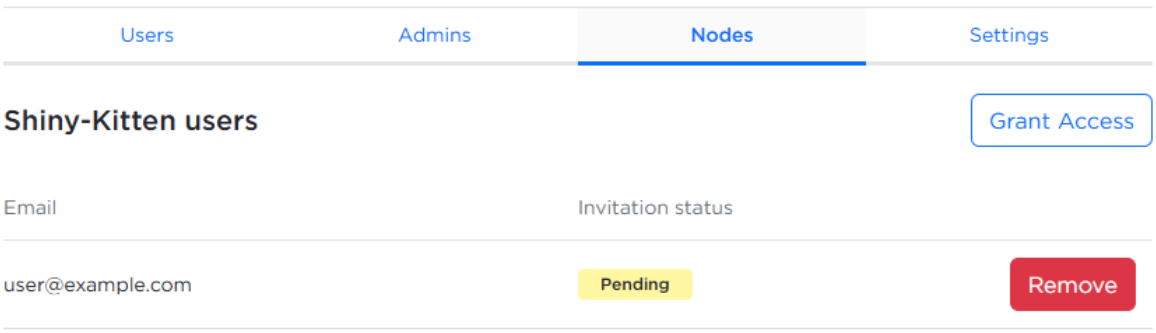
Приглашенный пользователь будет добавлен со статусом *Pending*.
Вы можете отозвать приглашение или отправить его повторно (повторная отправка доступна не чаще одного раза в 10 минут — кнопка будет отключена, если этот лимит достигнут).



Добавление пользователей к узлам сети

Чтобы добавить пользователя к узлу сети:

1. Перейдите на вкладку *Nodes*.
2. Нажмите *Grant Access*.
3. Выберите пользователя, добавленного в организацию, из выпадающего списка.
4. Нажмите *Grant access*.



Также можно предоставить доступ из окна *Node configuration*:

1. Нажмите кнопку настроек  .
2. Выберите *Add user*.
3. Выберите пользователя и нажмите *Grant access*.

Приглашённые пользователи должны принять приглашение, создав учётную запись Zabbix Cloud с использованием того же адреса электронной почты, на который было отправлено приглашение. Если у пользователя уже есть учётная запись Zabbix Cloud для этого адреса электронной почты, он должен войти в систему и подтвердить приглашение. Приглашения привязаны к конкретному адресу электронной почты и не могут быть приняты из другой учётной записи.

Ключевые отличия Zabbix Cloud

Сравнительная таблица Хотя большая часть функциональности пересекается между локальными установками Zabbix и облачными узлами, существуют и различия. Ниже приведена сводная сравнительная таблица. В последующих разделах отдельные элементы данных описаны подробнее.

Функция / Аспект	Zabbix Cloud	Локальный Zabbix
Цикл релизов	Синхронизирован с LTS-релизами	Вы сами выбираете версию Zabbix (стабильная, бета, пользовательские сборки)
Простота использования	Готовая к использованию платформа, полностью управляемая Zabbix	Требуются установка, настройка и обслуживание
Установка исправлений уязвимостей	Автоматическая установка исправлений ОС/Zabbix	Ручная установка исправлений пользователем
HTTPS-сертификат	Действительный сертификат «из коробки» (без Let’s Encrypt)	По умолчанию самоподписанный (можно настроить Let’s Encrypt или другой CA)
Производительность БД (INSERT/UPDATE/SELECT)	Автоматическое партиционирование для максимальной скорости; без длительных DELETE	Ручное партиционирование и очистка (возможны длительные транзакции DELETE)
Настройка производительности	Предварительно настроено командой Zabbix Cloud (без журналов изменений)	Полная видимость и контроль над параметрами настройки
Управление межсетевым экраном	Управляется командой Zabbix Cloud; порты GUI/trapper не доступны публично	Полный контроль над правилами межсетевого экрана и доступностью портов
Изоляция и защита БД	Высокая степень изоляции, без прямого доступа к БД/SSH	Зависит от вашей сети и безопасности узла сети
Высокая доступность (HA)	Работает на высокодоступной облачной платформе, для повышения отказоустойчивости сбора данных можно использовать прокси Zabbix	HA под управлением пользователя
SNMP traps	Только через выделенный прокси (без HA/балансируемого получателя trap-сообщений)	Напрямую или через прокси, возможны HA/балансировка нагрузки
SNMP polling	Сложно без прокси (каждому устройству нужен NAT/пользовательский порт)	Встроенный SNMP polling; прокси не обязателен
Настройки хранения	History/Trends/Audit только через UI (без API или конфигурационного файла)	Настраивается в конфигурационном файле сервера Zabbix или через API
Пользовательские скрипты	`AlertScriptsPath`, `ExternalScripts`, веб-интерфейс и модули сообщества не поддерживаются	Полностью поддерживается (путь к скриптам, модули, интеграции)
ODBC-мониторинг	Только PostgreSQL (драйвер `{postgresql}`); плагин MariaDB присутствует как заглушка	ODBC для PostgreSQL, MySQL, Oracle и т. д. — полностью настраивается
Ограничение ODBC-вызовов	Невозможно (только `StartODBCPollers=1`; тяжёлые синхронные запросы блокируют другую работу)	Полностью настраиваемое количество poller-процессов и расписание
Загрузка SAML-сертификата	Не поддерживается	Поддерживается через UI или API

Функция / Аспект	Zabbix Cloud	Локальный Zabbix
Плановые отчёты	Необходимо создать собственный тип оповещения Email (скриптовый тип оповещения не поддерживается)	Поддерживаются как скриптовый тип оповещения, так и Email «из коробки»
Интерфейс узла сети для активных проверок	IP-адрес, назначаемый платформой (создаётся автоматически)	Интерфейсы управляются вами; IP-адреса находятся под вашим контролем

Различия функций 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 для SNMP-опроса требуется, чтобы каждое устройство было доступно через NAT и пользовательские порты.

Настройки хранения

Периоды хранения истории, трендов и журнала аудита в облаке можно настраивать только через веб-интерфейс. Эти настройки нельзя задать через `zabbix_server.conf` или API.

Ручное переопределение хранения истории для каждого элемента данных не поддерживается (разделение на партиции управляется глобально). URL-адреса облачных узлов не принимают те же параметры запроса, что и локальные установки.

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, не могут быть установлены.

ODBC-мониторинг

Zabbix Cloud поддерживает ODBC-мониторинг только для PostgreSQL. Используйте официальный ODBC-шаблон Zabbix и определите строку подключения в шаблоне, используя:

```
Driver={postgresql}
```

Плагин MariaDB также установлен, но в настоящее время не работает. При использовании укажите:

```
Driver={mysql}
```

Attention:
Для Oracle MySQL 8.0 в Zabbix Cloud нет известной рабочей конфигурации. Хотя простые запросы (например, `SELECT 1`) могут выполняться успешно, более сложные запросы возвращают `SQL_ERROR`.

 Ограничить ODBC-вызовы невозможно (`StartODBCPollers=1` only). Тяжелая 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 сети, а также не разрешает прямые подключения к базе данных (например, к порту 3306). Все действия по настройке, мониторингу и устранению неполадок должны выполняться через облачный пользовательский интерфейс или API, что обеспечивает изоляцию и защиту уровней ОС и базы данных.

Интерфейс активных проверок

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 создания.

Загрузка сертификата SAML

Аутентификация SAML не поддерживается, поскольку немногие провайдеры принимают незашифрованные или неподписанные сообщения SAML.

Запланированные отчеты

Медиа-тип *Cloud Email* по умолчанию использует скриптовый транспорт и не может использоваться для запланированных отчетов. Чтобы отправлять отчеты по электронной почте:

1. Создайте новый медиа-тип Email.
2. Назначьте медиа электронной почты своему пользователю в *Настройки пользователя* > *Медиа*.
3. Выберите этот медиа-тип при настройке запланированных отчетов.

Центр разработчика

Центр разработчика

Начните разрабатывать пользовательские расширения для Zabbix с руководствами по модулям, виджетам и плагинам.

Создание на основе Zabbix

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 Описание структуры и логики виджетов, а также инструкции по созданию пользовательских элементов панели мониторинга, адаптированных под ваши потребности.

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

Плагины

Обзор Пользовательские загружаемые плагины расширяют функциональность Zabbix agent 2. Они компилируются отдельно, но используют общий пакет с Zabbix agent 2.

Каждый плагин является *Go*-пакетом, который определяет структуру и реализует один или несколько интерфейсов плагина (*Exporter*, *Configurator*, *Runner*).

Перейти к:

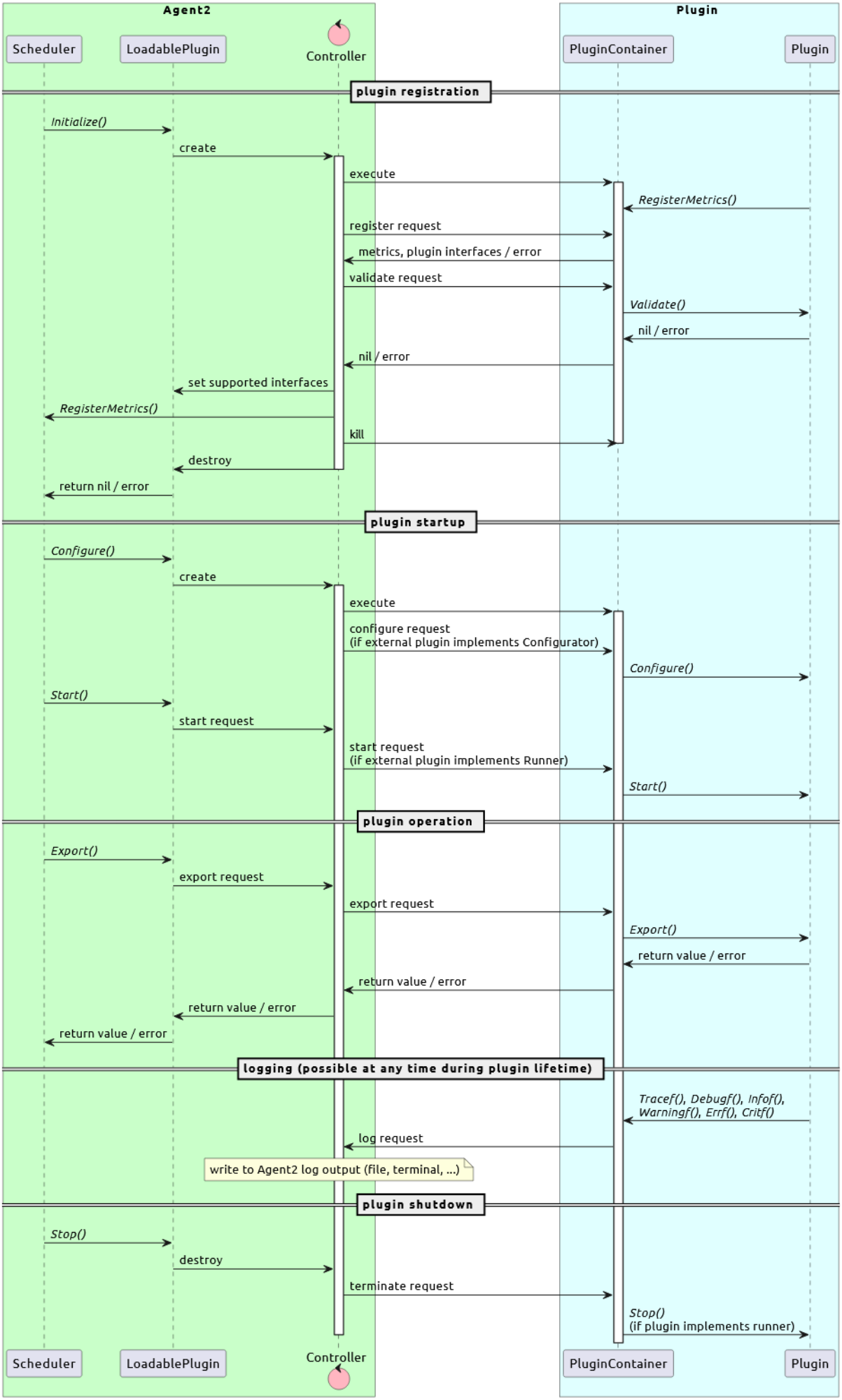
- Написание вашего первого плагина
- Интерфейсы плагинов

См. также:

- Пример плагина для Zabbix agent 2

Диаграмма подключения Zabbix agent 2 подключается к плагинам двунаправленно, используя UNIX-сокеты на Linux и именованные каналы (Named Pipes) на Windows.

Диаграмма подключения ниже иллюстрирует процесс связи между Zabbix agent 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

What you'll create This is ste-by-step tutorial for creating a simple loadable plugin named **MyIP**. This plugin will implement a single item key (`myip`) returning the external IP address of the Zabbix agent host.

Step 1: Setup

1. A plugin is a standard Go module. Start by initializing the `go.mod` file in the plugin directory to track plugin dependencies:

```
cd path/to/plugins/myip # Switch to your plugin directory
go mod init myip
```

2. Install the mandatory dependency Zabbix Go SDK (`golang.zabbix.com/sdk`):

```
go get golang.zabbix.com/sdk@$LATEST_COMMIT_HASH
```

Replace `$LATEST_COMMIT_HASH` with the latest `HEAD` commit hash from the `golang.zabbix.com/sdk` repository in the appropriate release branch. For example:

```
go get golang.zabbix.com/sdk@af85407
```

Note that `golang.zabbix.com/sdk` versioning is currently not supported, but this may change in the future.

Additional dependencies can be installed as needed using `go get`.

3. Create an empty `main.go` file for the plugin source code:

```
touch main.go
```

Now the initial setup is complete, and the plugin is ready for development.

Step 2: Plugin structure The `golang.zabbix.com/sdk` module, installed in the previous step, provides the necessary framework for plugin development and ensures all plugins have a consistent structure.

1. Set up basic execution flow.

Start by defining the main execution flow of the plugin. Add the following code to `main.go`:

```
package main

func main() {
    err := run()
    if err != nil {
        panic(err)
    }
}

func run() error {
    return nil
}
```

This establishes the basic execution flow for the plugin. The run function will later contain the core logic of the plugin.

2. Explore the plugin interfaces.

A Zabbix agent 2 plugin shall be represented by a struct that implements interfaces from the `golang.zabbix.com/sdk/plugin` package:

- *Accessor* - defines essential methods all plugins must implement, such as setting the plugin name and handling item key timeouts.
- One or more of the following functional plugin interfaces:
  - *Exporter* - performs a poll and returns a value (or values), nothing, or an error; often used alongside the *Collector* interface.
  - *Collector* - manages the periodic collection of data.
  - *Runner* - defines plugin startup and shutdown procedures.
  - *Watcher* - allows to implement independent metric polling, bypassing the agent's internal scheduler; useful for trap-based or event-driven monitoring.
  - *Configurator* - defines how the plugin reads and applies its configuration settings.

You can either implement these interfaces yourself or use the default ones provided by the Zabbix Go SDK, modifying them as needed. This tutorial uses the default implementations.

3. Create the plugin struct.

Now, import the *plugin* package and create a `myIP` struct that embeds the `plugin.Base` struct:

```
import "golang.zabbix.com/sdk/plugin"

type myIP struct {
    plugin.Base
}
```

The myIP struct currently satisfies the Accessor interface. A method for implementing one of the functional plugin interfaces, the `Exporter`, will be added later in the tutorial.

Step 3: Define item keys Your plugin needs the item keys to collect data and provide it to Zabbix server or proxy.

1. Import *errs* package for error handling:

```
import "golang.zabbix.com/sdk/errs"
```

2. Register item keys using the `plugin.RegisterMetrics()` function within the `run()` function:

```
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
}
```

To register several item keys, repeat the parameters *metric name* and *description* for each metric. For example:

```
plugin.RegisterMetrics(&impl, "Myip", "metric.one", "Metric one description.", "metric.two", "Metric two d
```

Step 4: Set up the handler The handler facilitates communication between the agent and the plugin.

1. Import the *container* package:

```
import "golang.zabbix.com/sdk/plugin/container"
```

2. Indise the `run()` function add code to create and set up a handler:

```
func run() error {
    p := &myIP{}

    // Register the `myip` item key.
    err := plugin.RegisterMetrics(
        p,
        "MyIP",                          // Plugin name
        "myip",                          // Item key name
        "Returns the host's IP address.", // Item key description
    )
    if err != nil {
        return errs.Wrap(err, "failed to register metrics")
    }

    // Create a new handler.
    h, err := container.NewHandler("MyIP") // Plugin name
    if err != nil {
        return errs.Wrap(err, "failed to create new handler")
    }
```

```
    // Setup logging to forward logs from the plugin to the agent.
    // Available via p.Logger.Infof, p.Logger.Debugf, etc.
    p.Logger = h

    // Start plugin execution.
    // Blocks until a termination request is received from the agent.
    err = h.Execute()
    if err != nil {
        return errs.Wrap(err, "failed to execute plugin handler")
    }

    return nil
}
```

Step 5: Implement data collection Data collection is done via the Exporter interface, which describes the `Export` method:

```
func Export(
  key string,             // The item key to collect.
  params []string,        // Arguments passed to the item key (`myip[arg1, arg2]`).
  context ContextProvider // Metadata about the item key data collection.
) (any, error)
```

1. Import the required packages for HTTP requests and response reading:

```
import (
    "io"
    "net/http"
)
```

2. Implement the `Export` method for the `myIP` struct:

```
func (p *myIP) Export(
    key string, params []string, context plugin.ContextProvider,
) (any, error) {
    // The plugin can use different data collection logic based on the `key` parameter.
    // This implementation only verifies that the provided `key` is supported.
    if key != "myip" {
        return nil, errs.Errorf("unknown item key %q", key)
    }

    // The log will get forwarded to the agent 2 log.
    p.Infof(
        "received request to handle %q key with %d parameters",
        key,
        len(params),
    )

    // Collect the data and return it.

    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
}
```

Step 6: Build and configure the plugin

1. To build the plugin, run:

```
go mod tidy
go build
```

This should create an executable `myip` in the current directory.

2. Configure Zabbix agent 2 to use the plugin:

```
echo "Plugins.MyIP.System.Path=$PATH_TO_THE_MYIP_PLUGIN_EXECUTABLE" > /etc/zabbix_agent2.d/plugins.d/myip.
```

Replace `$PATH_TO_THE_MYIP_PLUGIN_EXECUTABLE` with the path to the `myip` created in step 5.

The plugin name in the configuration parameter name (*MyIP* in this tutorial) must match the plugin name defined in the *plugin.RegisterMetrics()* function.

3. To test the plugin and its `myip` item, run:

```
zabbix_agent2 -c /etc/zabbix_agent2.conf -t myip
```

The output should contain an external IP address of your host and look similar to this:

```
myip                                          [s|192.0.2.0]
```

With that, you have created a simple loadable plugin for Zabbix agent 2. Congrats!

```go
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{}

    // 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 from the agent is received.
    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) {
    // The plugin can use different data collection logic based on the `key` parameter.
    // This implementation only verifies that the provided `key` is supported.
    if key != "myip" {
        return nil, errs.Errorf("unknown item key %q", key)
    }

    // The log will get forwarded to the agent 2 log.
    p.Infof(
        "received request to handle %q key with %d parameters",
        key,
        len(params),
    )

    // Collect the data and return it.

    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
}
```

Complete source code

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

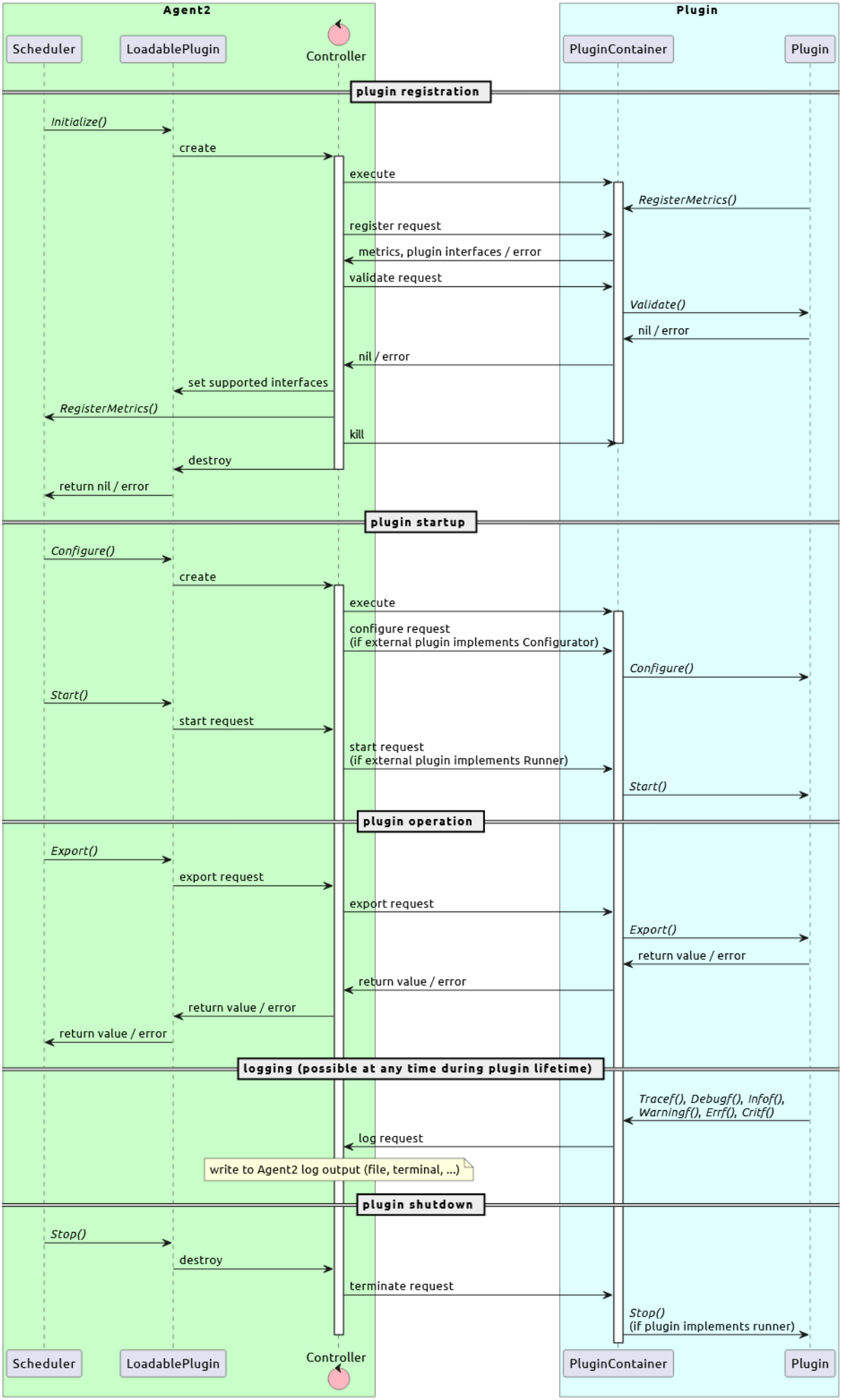
задачу. Кроме того, существует ограничение в 100 одновременных вызовов *Export()* на каждый плагин, которое можно уменьшить в соответствии с требованиями каждого плагина.

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.

Connection diagram Zabbix agent 2 connects bidirectionally to the plugins using UNIX sockets on Linux and Named Pipes on Windows.

The connection diagram below illustrates the communication process between Zabbix agent 2 and a loadable plugin and the metrics collection process.


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 Zabbix API для управления объектами Zabbix (создавать узлы сети, обновлять элементы данных, получать события и т. д.).
- Собирать данные с Zabbix агент (аналогично Zabbix get).
- Отправлять данные на Zabbix сервер или прокси (аналогично Zabbix sender).

Библиотека поддерживает как синхронное, так и асинхронное выполнение скриптов, что позволяет использовать её 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 Zabbix сервер и современными выпусками Python.

Чтобы начать, установите библиотеку, а затем следуйте руководству Быстрый старт.

Установка

zabbix_utils можно установить любым из описанных ниже способов.

Требования:

- Zabbix 6.0 или новее (протестировано на 6.0, 7.0, 7.2, 7.4)
- Python 3.8 или новее (протестировано на 3.8–3.13)
- асинхронный HTTP-фреймворк aiohttp (требуется только для асинхронного режима)

Python Package Index (PyPI)

Это наиболее распространённый метод для большинства сред:

```
pip install zabbix_utils
pip install zabbix_utils[async] #             async
```

Репозиторий Zabbix

Используйте этот метод, если вы предпочитаете управлять зависимостями через пакетный менеджер вашей системы.

Загрузите 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 разрабатываемую версию:

```
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.

Краткое руководство

После того как вы установили zabbix_utils, вы можете использовать его в своем скрипте.

Это краткое руководство показывает, как:

- Получить имя узла сети от Zabbix агент.
- Создать узел сети и элемент данных trapper с помощью Zabbix API.
- Отправить значение в элемент данных.

В руководстве скрипт представлен пошагово, с пояснением каждой части по мере ее добавления. Полный скрипт приведен в конце страницы.

В руководстве также предполагается, что ваш Zabbix сервер, агент и API запущены локально.

Получение имени узла сети от Zabbix агент

Начните с получения имени системы, на которой запущен Zabbix агент. Это имя узла сети потребуется для создания узла сети с помощью Zabbix API.

1. Импортируйте класс `Getter` из zabbix_utils. Этот класс работает как Zabbix get и позволяет запрашивать данные у Zabbix агент.

2. Создайте экземпляр `Getter`, который подключается к локальному Zabbix агент по адресу `127.0.0.1` на порту 10050.

3. Вызовите метод `get()` у экземпляра `Getter`, чтобы запросить имя узла сети у Zabbix агент.

- Метод `get()` принимает ключ элемента данных в качестве параметра и отправляет запрос Zabbix агент для этого элемента данных.
- Метод `get()` возвращает объект, а имя узла сети хранится в атрибуте `value` этого объекта.
- `system.hostname` — это встроенный ключ Zabbix агент, 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 ваше имя пользователя и пароль. Учетная запись должна иметь разрешение на доступ к Zabbix API.

4. Вызовите метод `logout()`, чтобы закрыть сеанс после 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 от Zabbix агент и установлено подключение к Zabbix API, вы можете создать новый узел сети в Zabbix.

1. Вызовите метод API `host.create()` для экземпляра `ZabbixAPI` и укажите сведения об узле сети:

- `host` — задайте значение переменной `hostname`, содержащей имя узла сети, полученное от Zabbix агент.
- `interfaces` — содержит сведения о подключении для Zabbix агент, запущенного на узле сети.
- `groups` — содержит как минимум одну группу узлов сети, к которой должен принадлежать узел сети.

2. Поскольку Zabbix API возвращает ID только что созданного узла сети, сохраните этот ID в переменной `host_id` для дальнейшего использования.

3. Выведите сообщение для подтверждения того, что узел сети был создан.

```
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}")
```

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()` у экземпляра `ZabbixAPI` и укажите сведения об элементе данных:

- `hostid` — установите значение переменной `host_id`, содержащей ID только что созданного узла сети.
- `name` — имя элемента данных.
- `key_` — установите значение переменной `item_key`, которую вы только что определили.
- `type` — установите значение 2 (элемент данных trapper); это необходимо для получения значений, отправляемых из вашего скрипта на следующих шагах.
- `value_type` — установите значение 3 (numeric unsigned), тип данных, которые хранит этот элемент данных.

3. Поскольку Zabbix API возвращает ID только что созданного элемента данных, сохраните этот ID в переменной `item_id` для дальнейшего 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 свой узел сети и элемент данных, вы можете отправить в 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 сведения:

- `hostname` — установите в переменную `hostname`, содержащую имя узла сети, полученное от Zabbix агент.
- `item_key` — установите в переменную `item_key`, которую вы только что определили ранее в своем скрипте.
- `1` — значение для отправки.

4. Выведите сообщение, подтверждающее, 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 успехе, указывающее, что Zabbix получил ваше значение:

```
Sender response: {"processed": 1, "failed": 0, "total": 1, "time": "0.000151", "chunk": 1}
```

Теперь вы также можете проверить значение в веб-интерфейс Zabbix (*Мониторинг* > *Последние данные*).

Полный скрипт

Ниже приведён полный скрипт, который объединяет все шаги: получение имени узла сети, создание узла сети и элемента данных с помощью Zabbix API, а также отправку данных в Zabbix.

```
import time
from zabbix_utils import Getter, ZabbixAPI, Sender

### Retrieve host name from Zabbix agent
agent = Getter(host='127.0.0.1', port=10050)
hostname = agent.get('system.hostname').value

### Connect and log in to Zabbix API
api = ZabbixAPI(url='127.0.0.1/zabbix')
api.login(user='Admin', password='zabbix')

### Create host in Zabbix
host = api.host.create(
    host=hostname,
    interfaces=[{
        'type': 1,
        'main': 1,
        'useip': 1,
        'ip': '127.0.0.1',
        'dns': '',
        'port': '10050',
    }],
    groups=[{'groupid': '2'}]
)

host_id = host['hostids'][0]

print(f"Host '{hostname}' created with ID {host_id}")

### Create item in Zabbix
item_key = 'app.myservice.heartbeat'

item = api.item.create(
    hostid=host_id,
    name='App heartbeat',
    key_=item_key,
    type=2,
    value_type=3
```

```
)

item_id = item['itemids'][0]

print(f"Item '{item_key}' created with ID {item_id}")

### Log out from API
api.logout()

### Wait for Zabbix to process new host
time.sleep(10)

### Send value to host
sender = Sender(server='127.0.0.1', port=10051)
response = sender.send_value(hostname, item_key, 1)

print(f"Sender response: {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.

API-запросы могут выполняться в синхронном или асинхронном режиме:

- В синхронном режиме ваш Python-скрипт отправляет запрос и ожидает ответа перед продолжением работы, что подходит для простых, последовательных и предсказуемых операций.
- В асинхронном режиме скрипт отправляет запросы, не дожидаясь каждого ответа, что позволяет параллельно выполнять другие операции; это более эффективно для медленных запросов или больших пакетов данных.

Примеры на этой странице сосредоточены на синхронном режиме, хотя асинхронный режим использует схожие подходы. Дополнительные примеры доступны в GitHub-репозитории 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 Python-библиотеку сообщества, обычно можно заменить этот импорт на ZabbixAPI из zabbix_utils.

Вход в систему

Перед выполнением API-запросов необходимо создать экземпляр ZabbixAPI и войти в Zabbix API.

Выберите метод, который лучше всего соответствует тому, как вы управляете учетными данными. Вы можете использовать имя пользователя и пароль или API-токены.

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, используя синтаксис именованных аргументов Python:

```
ZABBIX_AUTH = {
    "url": "127.0.0.1/zabbix",
    "user": "Admin",
```

```
    "password": "zabbix"
}

api = ZabbixAPI(**ZABBIX_AUTH)
```

Использование `login()`

Сначала вы можете создать экземпляр `ZabbixAPI`, указав только URL веб-интерфейса Zabbix, а затем использовать метод `login()` со своими учетными данными:

```
###                 :
api = ZabbixAPI(url="127.0.0.1/zabbix")
api.login(user="Admin", password="zabbix")

### API    :
api = ZabbixAPI(url="127.0.0.1/zabbix")
api.login(token="your_api_token")
```

Использование переменных окружения

Сначала вы можете сохранить свои учетные данные как переменные окружения, например, с помощью терминала:

```
###                 :
export ZABBIX_USER="Admin"
export ZABBIX_PASSWORD="zabbix"

###    :
export ZABBIX_TOKEN="your_api_token"
```

Затем, если вы запускаете свой скрипт из того же сеанса терминала, экземпляру `ZabbixAPI` в скрипте требуется только URL веб-интерфейса Zabbix:

```
api = ZabbixAPI(url="127.0.0.1/zabbix")
```

Вы также можете сохранить URL в переменной окружения:

```
###                 :
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 окружения, экземпляру `ZabbixAPI` в скрипте вообще не требуются никакие аргументы:

```
api = ZabbixAPI()
```

Выход из системы

Если вы вошли в систему, используя имя пользователя и пароль, вызовите метод `logout()` после завершения операций API:

```
from zabbix_utils import ZabbixAPI

api = ZabbixAPI(url="127.0.0.1/zabbix")
api.login(user="Admin", password="zabbix")

###                 API

api.logout()
```

Note:
Вызов метода `logout()` не требуется при использовании токенов API.

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 увидеть список узлов сети с помощью `host.get`:

```
### 1. Import ZabbixAPI from zabbix_utils:
from zabbix_utils import ZabbixAPI

### 2. Create the ZabbixAPI instance and log in:
api = ZabbixAPI(url="127.0.0.1/zabbix")
api.login(user="Admin", password="zabbix")

### 3. Retrieve hosts matching a filter:
hosts = api.host.get(
    filter={
        "host": [
            "Zabbix server",
            "Linux server"
        ]
    }
)

### 4. Iterate over returned hosts and print each host's details:
for host in hosts:
    print(host)

### 5. Log out from the API to close the session:
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, если ему нужно выполнять много API-запросов или если выполнение некоторых запросов 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 `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 увидеть список узлов сети с помощью `host.get` в асинхронном режиме:

```
### 1.         asyncio                 AsyncZabbixAPI    zabbix_utils:
import asyncio
from zabbix_utils import AsyncZabbixAPI

### 2.             async-    ,                          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 распространённые задачи Zabbix API с использованием библиотеки zabbix_utils.

Дополнительные примеры доступны в каталоге `examples` GitHub-репозитория zabbix_utils.

Сбор данных с помощью агента Zabbix

Обзор

zabbix_utils позволяет собирать данные с Zabbix агент (аналогично Zabbix get).

Данные можно собирать в синхронном или асинхронном режиме:

- В синхронном режиме ваш Python-скрипт запрашивает данные и ожидает их получения перед продолжением работы, что подходит для простых, последовательных и предсказуемых операций.
- В асинхронном режиме скрипт запрашивает данные, не ожидая каждый ответ, что позволяет параллельно выполнять другие операции; это более эффективно для медленных запросов или больших пакетов данных.

Примеры на этой странице сосредоточены на синхронном режиме, хотя асинхронный режим использует схожие подходы. Дополнительные примеры доступны в GitHub-репозитории zabbix_utils.

Импорт

Чтобы использовать zabbix_utils для сбора значений элементов данных, импортируйте класс `Getter` в ваш Python-скрипт:

```
from zabbix_utils import Getter
```

Данные запроса

Чтобы запросить значение элемента данных:

1. Создайте экземпляр `Getter`, указав IP-адрес и порт вашего Zabbix агента.
2. Вызовите метод `get()` у экземпляра `Getter`, указав ключ элемента данных, который вы хотите получить.

Например, чтобы запросить данные для элемента данных `system.uname`:

```
agent = Getter(host='192.0.2.0', port=10050)
response = agent.get('system.uname')
```

Использование нестандартного IP-адреса

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='192.0.2.0',
    port=10050,
    source_ip='10.10.7.1'
)
```

Использование timeout

Вы можете задать `timeout` ответа для `Getter`, чтобы указать, как долго ваш скрипт должен ждать ответа от Zabbix агент, прежде чем прекратить ожидание:

```
agent = Getter(
    host='192.0.2.0',
    port=10050,
    timeout=30
)
```

Использование шифрования

`Getter` не включает встроенную поддержку шифрования, однако вы можете обеспечить её, создав обёртку с использованием сторонних библиотек:

```
def psk_wrapper(sock, tls):
    # ...
    #            TLS PSK
    # ...

agent = Getter(
    host='192.0.2.0',
    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, если ему нужно собрать много значений или если сбор некоторых значений занимает много времени.

При использовании асинхронного режима есть важные отличия по сравнению с синхронным режимом:

- Импортируйте модуль Python `asyncio` (сначала необходимо install требуемые зависимости).
- Импортируйте `AsyncGetter` вместо `Getter`.
- Пишите код внутри функции `async`.
- Используйте `await` при вызове метода `get()`.

Например, чтобы получить одно значение в асинхронном режиме:

```
### 1. Import asyncio for asynchronous mode, and AsyncGetter from zabbix_utils:
import asyncio
from zabbix_utils import AsyncGetter
```

```
### 2. Define the main async function where all data requests will be executed:
async def main():
    agent = AsyncGetter(host='192.0.2.0', port=10050)

    # 3. Fetch the system.uname value from Zabbix agent (must await):
    response = await agent.get('system.uname')

    # 4. Print the value returned by Zabbix agent:
    print(response.value)

### 5. Run the async main() function using asyncio's event loop:
asyncio.run(main())
```

Отправка данных на сервер Zabbix или прокси

zabbix_utils позволяет отправлять значения элементов данных в trapper item на сервер Zabbix или прокси (аналогично Zabbix sender).

Вы можете отправить одно значение, несколько значений или даже данные сразу в несколько кластеров Zabbix.

Данные можно отправлять в синхронном или асинхронном режиме:

- В синхронном режиме ваш Python-скрипт 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 параллельно выполнять другие операции; это более эффективно для медленных запросов или больших пакетов данных.

Примеры на этой странице сосредоточены на синхронном режиме, хотя асинхронный режим использует схожие подходы. Дополнительные примеры доступны в GitHub-репозитории zabbix_utils.

Импорт

Чтобы использовать zabbix_utils для отправки значений элементов данных, импортируйте класс `Sender` в ваш 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 или прокси Zabbix.
2. Вызовите метод `send_value()` для экземпляра `Sender`, используя следующий формат:

```
sender_instance.send_value('host', 'item.key', 'value', optional_timestamp, optional_nanoseconds)
```

Например, чтобы отправить 1 в элемент данных trapper `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 нестандартного IP-адреса

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 timeout

Вы можете задать `timeout` ответа для `Sender`, чтобы определить, как долго ваш скрипт должен ожидать 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 таких случаях вам не нужно указывать параметры подключения при создании экземпляра Sender:

```
sender = Sender(
    use_config=True,
    config_path='/etc/zabbix/zabbix_agent2.conf'
)
```

Attention:
Если ServerActive содержит один или несколько кластеров Zabbix с несколькими экземплярами сервера, Sender отправляет данные на первый доступный сервер в каждом кластере. Если ServerActive не задан, используется адрес из Server с портом по умолчанию (10051).

Использование шифрования

Sender не включает встроенную поддержку шифрования, но вы можете обеспечить её, создав обёртку 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 Zabbix или прокси, обрабатывается библиотекой и возвращается как объект 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 Python-скрипту отправлять значения без ожидания ответа от сервера Zabbix или прокси. Это может сделать ваш скрипт более эффективным, когда ему нужно отправить 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 install требуемые зависимости).
- Импортируйте AsyncSender вместо Sender.
- Пишите код внутри функции async.
- Используйте await при вызове метода send_value().

Например, чтобы отправить одно значение в асинхронном режиме:

```
### 1. Import asyncio for asynchronous mode, and AsyncSender from zabbix_utils:
import asyncio
from zabbix_utils import AsyncSender

### 2. Define the main async function where all data sending operations (must await) will be executed:
async def main():
    sender = AsyncSender(server='127.0.0.1', port=10051)
    response = await sender.send_value('Linux server', 'service.status', 1)

    # 3. Print the response returned by Zabbix server or proxy:
    print(response)

### 4. Run the async main() function using asyncio's event loop:
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()`) для 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 чанка

Если вам нужно отправить больше значений, чем элемент данных trapper может принять в одном запросе, вы можете разделить их на чанки.

По умолчанию размер чанка составляет 250 значений. Вы можете изменить его, задав параметр `chunk_size` при создании экземпляра `Sender`.

Например, чтобы отправить пять значений в трёх чанках (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 отправлять значения в несколько кластеров Zabbix:

1. Подготовьте массив кластеров Zabbix. Если кластер содержит несколько узлов, значение будет отправлено на первый **доступный** узел каждого кластера.
2. Создайте `Sender`, указав ваш массив кластеров Zabbix.
3. Вызовите метод `send_value()` экземпляра `Sender`, используя тот же формат, что и для метода `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 блока 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 журналирование

Для устранения проблем при работе с Zabbix API, сборе данных с Zabbix агент или отправке данных на Zabbix сервер или прокси вы можете включить отладочное журналирование.

Библиотека zabbix_utils использует стандартный модуль Python `logging`.

Чтобы увидеть подробные отладочные сообщения из библиотеки:

1. Импортируйте модуль Python `logging` в ваш скрипт.
2. Настройте систему журналирования так, чтобы выводить сообщения в удобочитаемом формате, и установите уровень журналирования 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)
```

Если отладочное журналирование включено, выводится подробная информация о запросах и ответах, например:

```
[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/                                                    ,                    .
       js/                                        JavaScript,                .
           example.something.view.js.php
       css/                                    CSS-   ,                .
           example.something.css
        image.png                                    ,                .
        example.something.file                                          .
```

Дерево файлов модуля

Написание модуля Пример процесса написания модуля состоит из следующих шагов (если возможно, щелкните имя файла или папки, чтобы просмотреть дополнительные сведения об этом 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 (поместите в *assets/js*), стилей CSS (поместите в *assets/css*) или любых других дополнительных файлов.
7. Обязательно укажите необходимые представления, действия и файлы ресурсов в файле Manifest.json.
8. Зарегистрируйте модуль в веб-интерфейсе Zabbix и начните его использовать.

Note:
Вы можете зарегистрировать и включить модуль, как только создадите файл manifest.json. Как только модуль включен, вы можете сразу просмотреть все изменения, внесенные в файлы модуля, обновив веб-интерфейс Zabbix.

manifest.json

Каждому модулю необходим файл manifest.json. Файл должен располагаться в главной директории модуля (например, *zabbix/ui/modules/module_name/manifest.json*).

Как необходимый минимум, файл manifest.json должен содержать следующие поля:

```
{
    "manifest_version": 2.0,
    "id": "my_ip_address",
    "name": "My IP Address",
    "namespace": "MyIPAddress",
    "version": "1.0"
}
```

Параметры, поддерживаемые **manifest.json** (нажмите на параметр для более подробной информации):

Параметр	Описание	Обязателен
manifest_version	Версия манифеста модуля.	Да
id	Уникальный ID модуля.	
name	Имя модуля, которое будет отображаться в разделе Администрирование.	
namespace	Пространство имён PHP для классов модуля.	
version	Версия модуля.	
type	Тип модуля. Для виджетов необходимо указать *widget*	Да для виджетов, для остальных нет
widget	Конфигурация виджета. Используется только для виджетов.	
actions	Действия для регистрации модуля.	
assets	Стили CSS и файлы JavaScript, которые необходимо включить.	Нет
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 откроется при нажатии на значок справки ? в окне *Добавить виджет* или *Редактировать виджет*. Если **url** не указан, то при нажатии на значок справки откроется общая страница Виджеты информационной панели.

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. Проверить, что все параметры, переданные в HTTP-запросе, являются допустимыми:

- Вызвать метод контроллера *checkInput()*;
- Использовать правила валидации, определенные в CNewValidator.php;
- Вызвать метод *validateInput()*.

2. Проверить права пользователя.
3. Подготовить данные в соответствии с переданными параметрами: если *checkInput()* возвращает true, Zabbix вызывает метод контроллера *doAction()*.
4. Подготовить 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 в Zabbix можно посмотреть в исходном коде.

Представления

Файл представления получает данные от контроллера, а затем формирует их HTML-вид.

Note:
Определение представления(й) для модуля веб-интерфейса необязательно, если только модуль не является виджетом.
Виджетам панели мониторинга требуется как минимум два представления: одно для режима редактирования и одно для режима просмотра (они должны храниться в каталоге *views*).

В представлении можно использовать предопределённые 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 *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"]
}
```

CSS стили

Файлы CSS могут содержать настраиваемый атрибут «тема», определяющий другой стиль для определенных тем внешнего интерфейса.

Доступные темы и значения их атрибутов:

- **Синий** - [theme='blue-theme']
- **Темный** - [theme='dark-theme']
- **Высококонтрастный свет** - [theme='hc-light']
- **Высококонтрастный темный** - [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 «Суперадминистратор»).

> **Note:**
> Интерфейс не будет устанавливать или даже распознавать несовместимые модули.

Добавление модуля Откройте страницу *Администрирование→Основные→Модули* и нажмите *Сканировать каталог*.



Найдите свой модуль в списке и активируйте его.

Чтобы активировать модуль, нажмите на гиперссылку *Отключено* - состояние модуля изменится на *Включено*.

Нажмите на имя модуля, чтобы просмотреть дополнительную информацию о модуле, например автора, версию или краткое описание (если оно определено в манифесте).

Предпросмотр виджета Модули виджетов после добавления сразу становятся видимыми в списке виджетов панели мониторинга.

Вы можете открыть панель мониторинга, переключиться в режим редактирования и добавить виджет на панель мониторинга обычным способом.

Если вы внесли какие-либо изменения в виджет, обновите панель мониторинга, чтобы увидеть, как виджет выглядит с учетом последних обновлений.

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

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

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 определения логики представления полей, определенных в классе *WidgetForm*, а также для задания их внешнего вида и поведения при отображении в представлении конфигурации.

Класс *CWidgetFormView* поддерживает следующие методы:

- *addField()* — получает в качестве параметра экземпляр класса *CWidgetFieldView*; каждый класс *CWidgetField* имеет соответствующий класс *CWidgetFieldView* для использования в представлении конфигурации виджета.
- *includeJsFile()* — позволяет добавить файл JavaScript в представление конфигурации виджета.
- *addJavaScript()* — позволяет добавить встроенный JavaScript, 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 Zabbix\Widgets\Fields\CWidgetFieldGraphDataSet;

(new CWidgetFormView($data))
    ->addField(
        new CWidgetFieldMultiSelectItemView($data['fields']['itemid'], $data['captions']['items']['itemid'
    )
    ->addField(
        new CWidgetFieldTextBoxView($data['fields']['description']),
        'js-advanced-configuration'
    )
    ->addField(
        new CWidgetFieldColorView($data['fields']['chart_color']),
        'js-advanced-configuration'
    )
    ->includeJsFile('widget.edit.js.php')
    ->addJavaScript('my_widget_form.init('.json_encode([
        'color_palette' => CWidgetFieldGraphDataSet::DEFAULT_COLOR_PALETTE
    ], JSON_THROW_ON_ERROR).');')
    ->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* — это базовый класс, 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	Один флажок.
CWidgetFieldCheckBoxList	массив int32	Несколько флажков в одном поле конфигурации.
CWidgetFieldColor	string	Поле выбора цвета.
CWidgetFieldDatePicker	строка	Поле выбора даты.
CWidgetFieldHostPatternSelect	строка	Поле множественного выбора, которое позволяет выбрать одного или нескольких узлов сети. Поддерживает определение шаблонов имен узлов сети (будут выбраны все соответствующие узлы сети).
CWidgetFieldIntegerBox	int32	Поле для ввода целого числа. Можно использовать для настройки минимальных и максимальных значений.
CWidgetFieldLatLng	строка	Поле для ввода широты, долготы и уровня увеличения карты через запятую.
CWidgetFieldMultiSelectAction	ID	Поле множественного выбора для выбора действий (из списка действий, определенных в разделе *Alerts* → *Actions*).
CWidgetFieldMultiSelectGraph	ID	Поле множественного выбора для выбора пользовательских графиков.
CWidgetFieldMultiSelectGraphPrototype	ID	Поле множественного выбора для выбора пользовательских прототипов графиков.
CWidgetFieldMultiSelectGroup	ID	Поле множественного выбора для выбора групп узлов сети.
CWidgetFieldMultiSelectHost	ID	Поле множественного выбора для выбора узлов сети.
CWidgetFieldMultiSelectItem	ID	Поле множественного выбора для выбора элементов данных.
CWidgetFieldMultiSelectItemPrototype	ID	Поле множественного выбора для выбора прототипов элементов данных.
CWidgetFieldMultiSelectMediaType	ID	Поле множественного выбора для выбора типов способов оповещения.
CWidgetFieldMultiSelectService	ID	Поле множественного выбора для выбора сервисов.
CWidgetFieldMultiSelectSla	ID	Поле множественного выбора для выбора соглашений об уровне обслуживания (SLA).
CWidgetFieldMultiSelectUser	ID	Поле множественного выбора для выбора пользователей.
CWidgetFieldNumericBox	строка	Поле для ввода числа с плавающей запятой.
CWidgetFieldRadioButtonList	int32	Группа переключателей, состоящая из одного или нескольких радиокнопок.

Класс CWidgetField	Тип поля в базе данных	Описание
CWidgetFieldRangeControl	int32	Ползунок для выбора значения целочисленного типа.
CWidgetFieldSelect	int32	Поле выбора раскрывающегося списка.
CWidgetFieldSeverities	массив int32	*CWidgetFieldCheckBoxList* с заданными уровнями важности триггеров.
CWidgetFieldTags	массив (строка, int32, строка)	Позволяет настроить одну или несколько строк фильтра тегов.
CWidgetFieldTextArea	строка	Текстовая область для ввода многострочного текста.
CWidgetFieldTextBox	строка	Текстовое поле для ввода однострочного текста.
CWidgetFieldTimeZone	строка	Раскрывающийся список с часовыми поясами.
CWidgetFieldUrl	строка	Текстовое поле, позволяющее вводить URL-адреса.

Следующие классы *CWidgetField* были созданы для определенных виджетов. Эти классы имеют очень специфические случаи использования, но их также можно использовать повторно, если это необходимо.

Класс CWidgetField	Тип поля в базе данных	Описание
CWidgetFieldColumnsList	массив (multiple mixed)	Для виджета *Топ узлов сети*. Создайте таблицу с настраиваемыми столбцами разрешенных типов.
CWidgetFieldGraphDataSet	массив (multiple mixed)	Для виджета *График*. Настройка конфигурации набора данных и всех связанных с ней опций.
CWidgetFieldGraphOverride	массив (multiple mixed)	Для виджета *График*. Переопределения настройки для конкретных узлов сети/элементов данных. Любую конфигурацию набора данных можно переопределить.
CWidgetFieldNavTree	строка	Для виджета *Дерево навигации по карте*. Заменяет вид виджета в режиме редактирования на дерево выбора карты.
CWidgetFieldReference	строка	Для виджета *Дерево навигации по карте*. Создает уникальный идентификатор для этого виджета на панели. Он используется для ссылки на этот виджет из других виджетов.
CWidgetFieldSelectResource	ID	Для виджета *Карта*. Позволяет выбрать карту сети Zabbix.
CWidgetFieldThresholds	массив (строка, строка)	Для виджета *Топ узлов сети*. Позволяет настроить цвет и числовые пары.
CWidgetFieldWidgetSelect	строка	Для виджета *Карта*. Позволяет выбрать дерево навигации карты из текущей панели. Должен использоваться в сочетании с *CWidgetFieldReference* в виджете *Дерево навигации по карте*.

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 мониторинга переходит в режим редактирования. Этот метод вызывается, когда панель мониторинга переходит в режим редактирования, обычно когда пользователь нажимает кнопку *Edit* виджета или кнопку *Edit dashboard* на панели мониторинга.

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

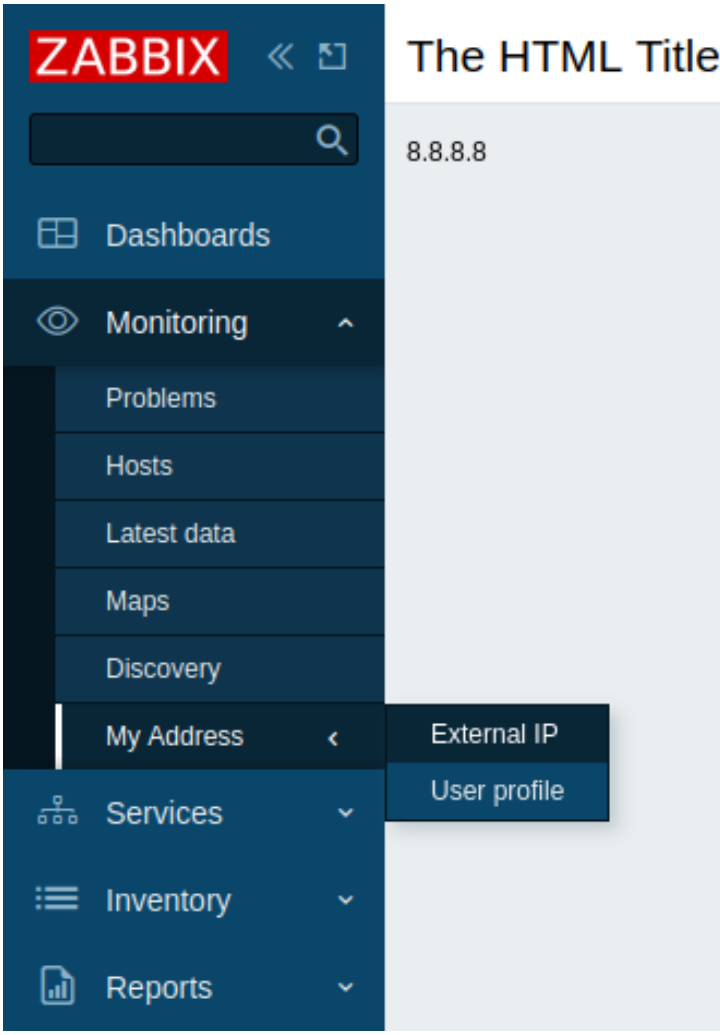
Этот раздел содержит практические пошаговые руководства, иллюстрирующие, как создать собственный модуль и виджет в Zabbix.

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

С помощью этой пошаговой инструкции вы сначала создадите модуль веб-интерфейса, который добавляет новый раздел меню *Мой адрес*, а затем преобразуете его в более продвинутый модуль веб-интерфейса, который отправляет HTTP-запрос на *https://api.seeip.org* и отображает ответ — IP-адрес вашего компьютера — на новой странице во вновь созданном



разделе меню *Мой адрес*. 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).

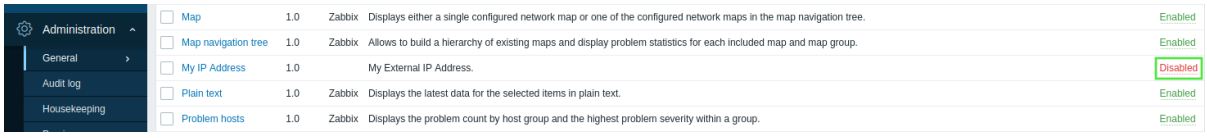
ui/modules/MyAddress/manifest.json

```
{
    "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 *Администрирование* → *Общее* → *Модули* и нажмите кнопку *Сканировать каталог*.



4. Найдите в списке новый модуль *My IP Address* и нажмите на гиперссылку "Disabled", чтобы изменить статус модуля с "Disabled" на "Enabled" (если модуль отсутствует в списке, см. раздел устранения проблем).



Теперь модуль зарегистрирован в веб-интерфейсе. Однако он пока не отображается, поскольку вам всё ещё необходимо

определить функциональность модуля. После добавления содержимого в каталог модуля вы сразу увидите изменения в веб-интерфейсе Zabbix после обновления страницы.

Создайте раздел меню

1. Создайте файл *Module.php* в каталоге *MyAddress*.

Этот файл реализует новый класс *Module*, который расширяет класс *CModule* по умолчанию. Класс *Module* вставит новый раздел меню *My Address* в главное меню.

Метод *setAction()* определяет действие, которое будет выполнено при нажатии на раздел меню. Для начала вы можете 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;

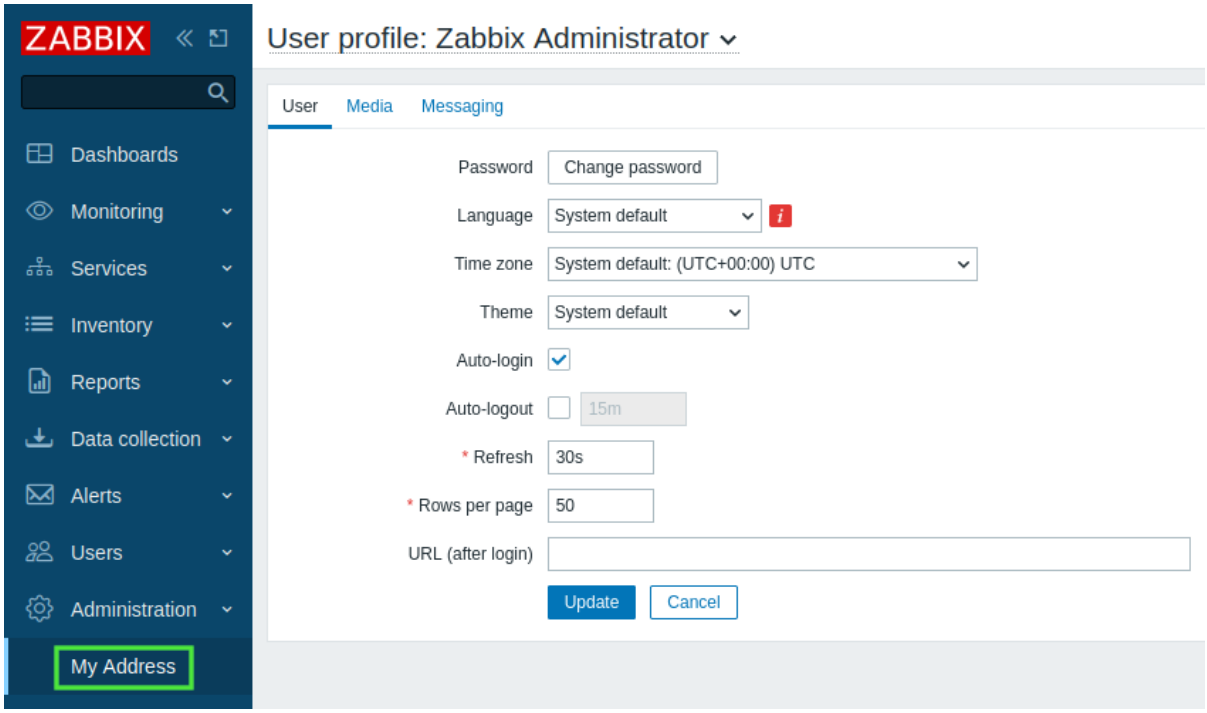
use Zabbix\Core\CModule,
    APP,
    CMenuItem;

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' другими действиями, например, 'charts.view' (открывает пользовательские графики), 'problems.view' (открывает *Мониторинг* → *Проблемы*) или ' .status' (открывает отчет *Информация о системе*).

3. Обновите интерфейс Zabbix. Теперь в нижней части главного меню Zabbix появился новый раздел *Мой адрес*. Нажмите *Мой адрес*, чтобы открыть страницу *Профиль пользователя*.



Часть II - Изменение расположения раздела меню

В этой части вы переместите раздел меню *Мой адрес* в раздел *Мониторинг*, а затем добавите в него вложенное меню. В результате пользователи смогут получить доступ к двум страницам подменю из раздела меню *Мониторинг* → *Мой адрес*.

1. Откройте и отредактируйте файл *Module.php*.

ui/modules/MyAddress/Module.php

```
<?php

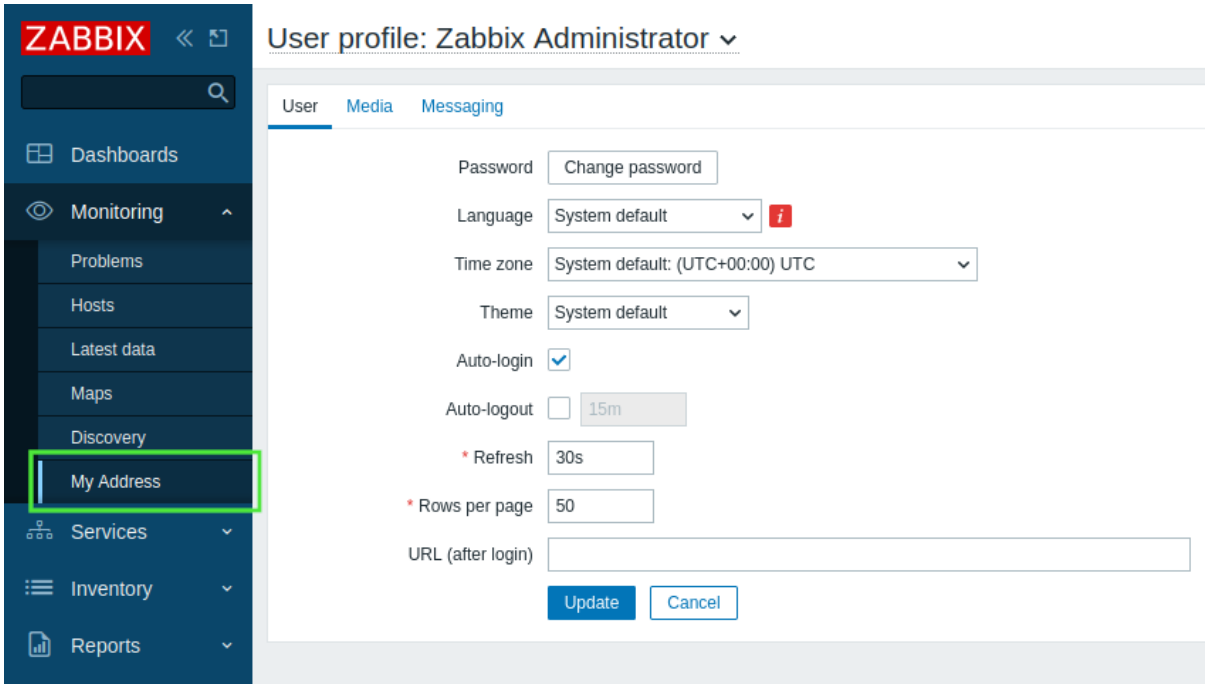
namespace Modules\MyAddress;

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 интерфейс Zabbix. Разверните раздел меню *Мониторинг* и обратите внимание, что раздел *Мой адрес* теперь расположен под разделом *Обнаружение*.



3. Чтобы добавить вложенные страницы в раздел меню *Мой адрес*, снова откройте и отредактируйте файл *Module.php*.

На этом этапе создаются два подраздела:

- *Внешний IP*, который выполняет новое действие «my.address», которое будет определено на следующих шагах;
- *Профиль пользователя*, который выполняет предопределенное действие «userprofile.edit», чтобы открыть страницу *Профиль пользователя*.

Обратите внимание, что для вложенного меню вам необходимо использовать класс *CMenu* в дополнение к классам, использованным на предыдущих шагах.

ui/modules/MyAddress/Module.php

```
<?php

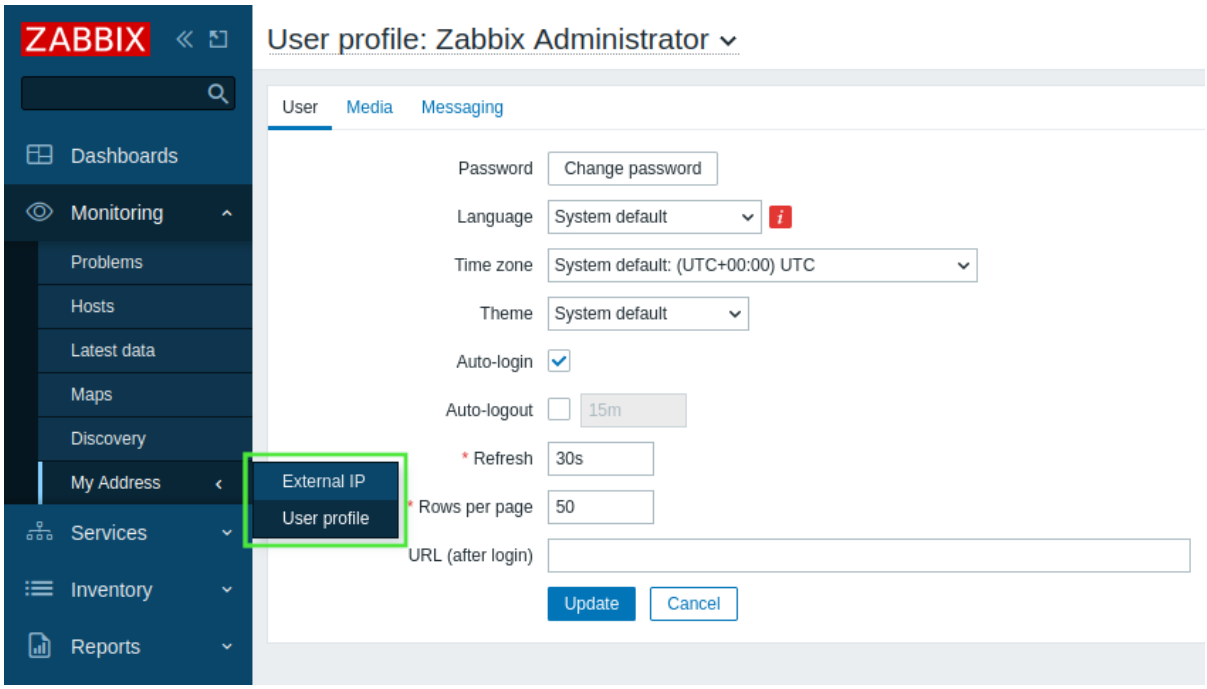
namespace Modules\MyAddress;
```

```
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 интерфейс Zabbix. Обратите внимание, что раздел меню *Мой адрес* теперь содержит меню третьего уровня с двумя страницами — *Внешний IP* и *Профиль пользователя*.



Часть III — Действие модуля

Действие реализовано 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*** отвечает за представление.

1. Создайте каталог *actions* в каталоге *MyAddress*.
2. Создайте файл *MyAddress.php* в каталоге *actions*.

Логика действий будет определена в классе *MyAddress*. Этот класс действий будет реализовывать четыре функции: *init()*, *checkInput()*, *checkPermissions()* и *doAction()*. Интерфейс Zabbix вызывает функцию *doAction()*, когда запрашивается действие. 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 ее специального определения. Платформа 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:

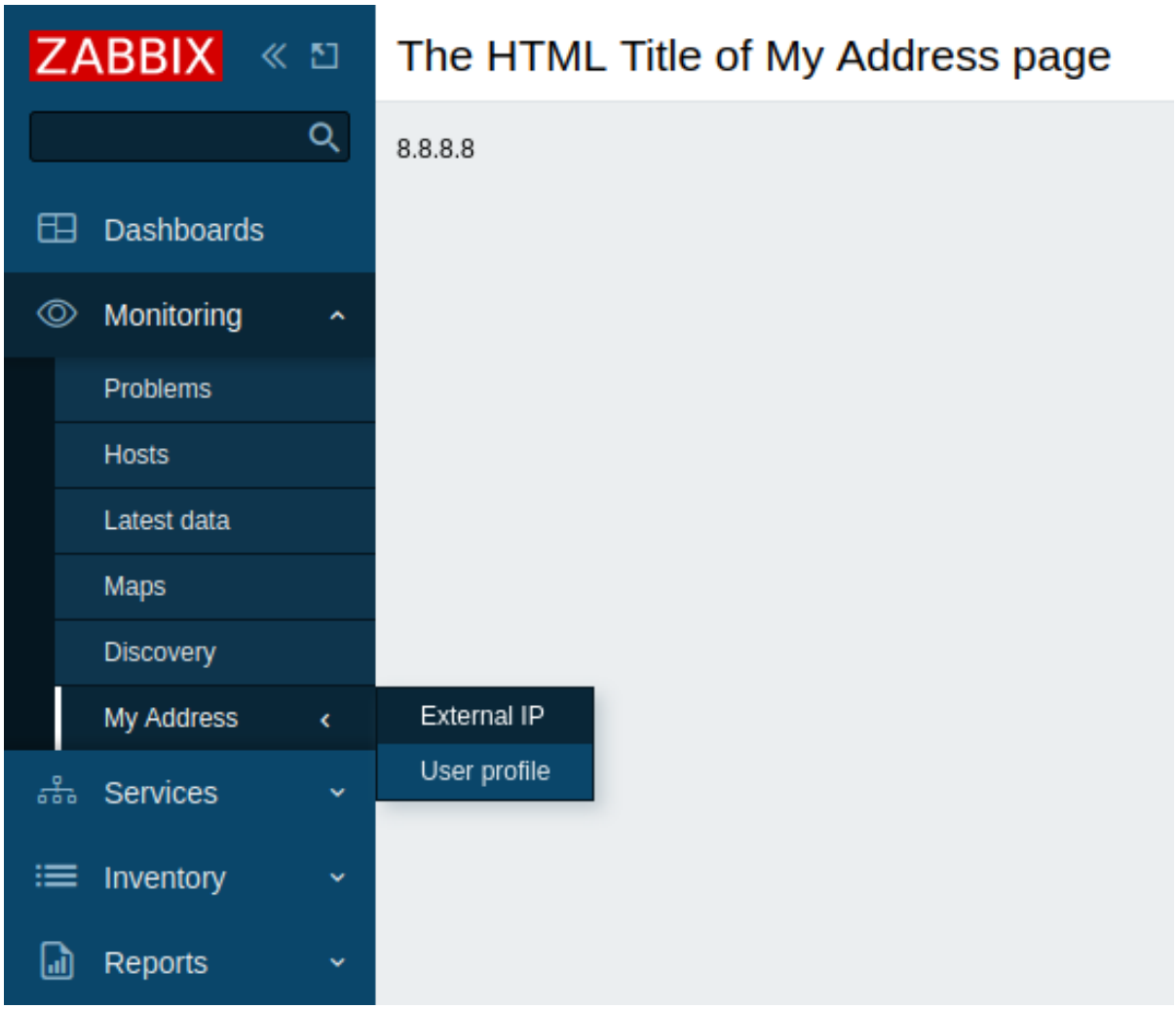
- клавиша действия с названием действия, написанным строчными буквами (a-z) и словами, разделенными точками (например, `my.address`);
- имя класса действия («MyAddress») как значение ключа «class» объекта «my.address»;
- имя представления действия («my.address») как значение ключа «view» объекта «my.address».

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"
        }
```

```
    }
}
```

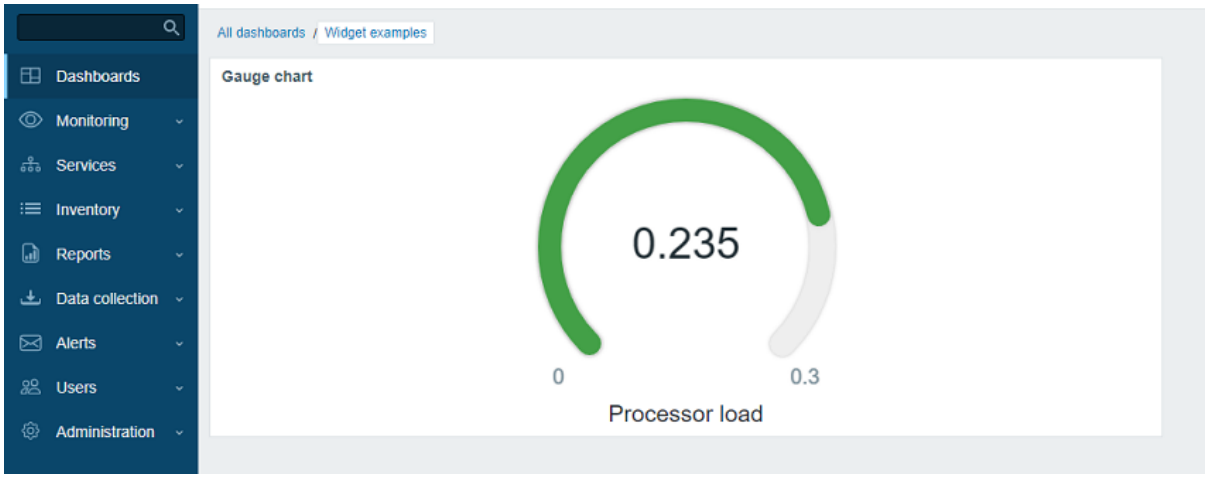
6. Обновите интерфейс Zabbix. Нажмите *Мой адрес → Внешний IP*, чтобы увидеть IP-адрес вашего компьютера.



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 будете создавать В ходе этого урока вы сначала создадите basic "Hello, world!" виджет, а затем преобразуйте его в виджет more advanced, который отображает значение элемента в виде диаграммы. Вот как будет выглядеть готовый виджет:



Часть I — «Привет, мир!» В этом разделе вы узнаете, как создать минимально необходимые элементы виджета и добавить новый виджет в веб-интерфейс Zabbix.

Добавление пустого виджета в 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 добавляться 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 пользовательским интерфейсом Zabbix.

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"
}
```

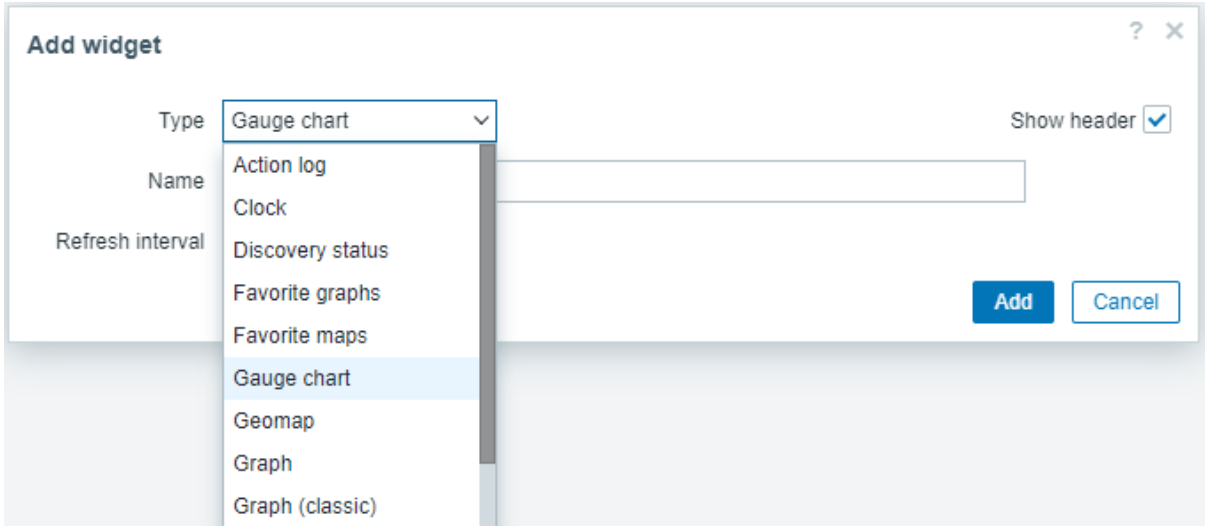
3. В веб-интерфейсе Zabbix перейдите в раздел *Администрирование* → *Общие* → *Модули* и нажмите кнопку *Сканировать каталог*.



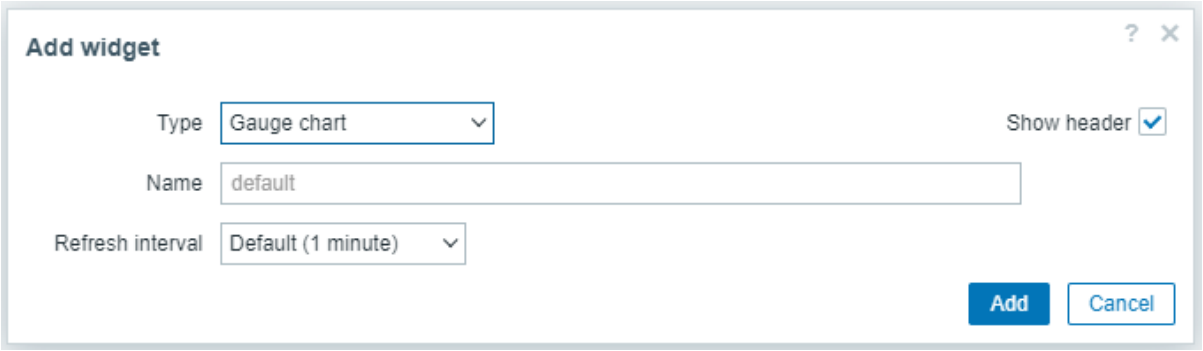
4. Найдите новый модуль *Gauge chart* в списке и нажмите на гиперссылку "Disabled", чтобы изменить статус модуля с "Disabled" на "Enabled" (если модуль отсутствует в списке, см. раздел устранения неполадок).



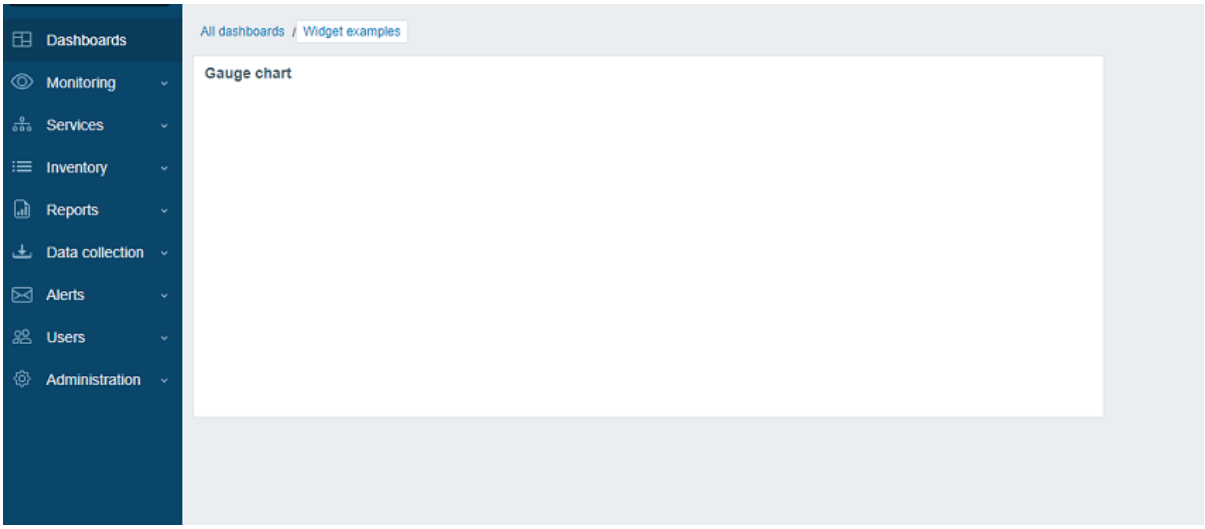
5. Откройте панель мониторинга, переключите её в режим редактирования и добавьте новый виджет. В поле *Type* выберите "Gauge chart".



6. На этом этапе конфигурация виджета *Gauge chart* содержит только общие поля виджета *Name* и *Refresh interval*. Нажмите *Add*, чтобы добавить виджет на панель мониторинга.



7. На панели мониторинга должен появиться пустой виджет. Нажмите *Save changes* в правом верхнем углу, чтобы сохранить панель мониторинга.



Добавление представления виджета

Note:
Файл **представления** виджета должен находиться в каталоге *views* (в этом руководстве — *ui/modules/lesson_gauge_chart/views/*). Если файл имеет имя по умолчанию *widget.view.php*, вам не нужно регистрировать его в файле *manifest.json*. 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*.

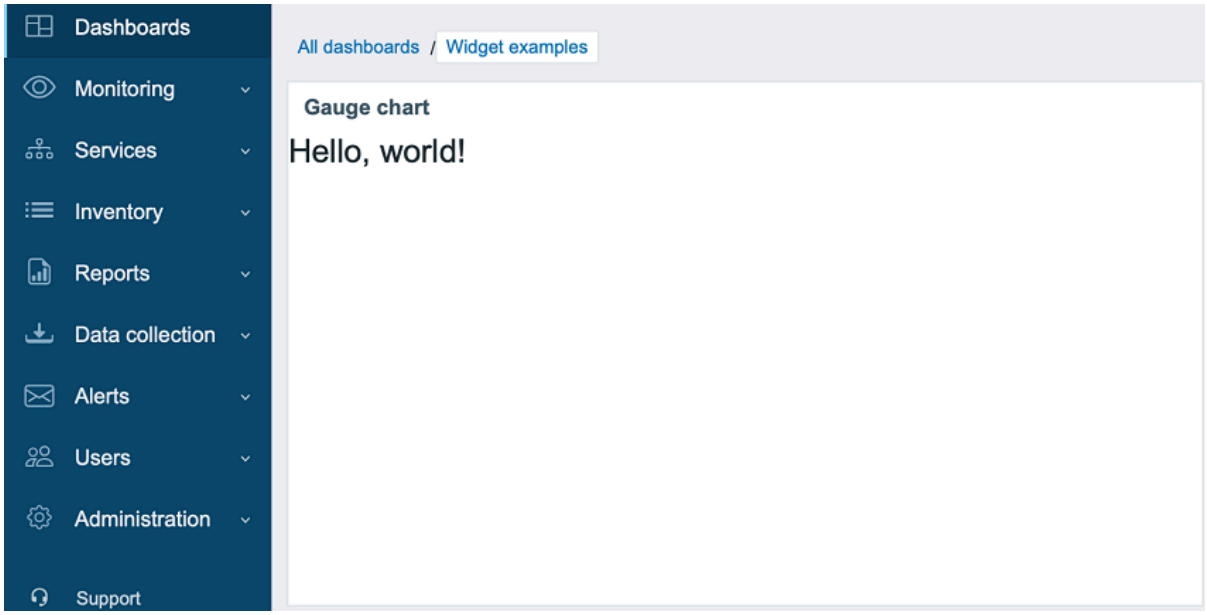
ui/modules/lesson_gauge_chart/views/widget.view.php

```
<?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 их использование в представлении виджета

В этом разделе вы узнаете, как добавить поле конфигурации виджета и отобразить введённое 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, введённых пользователем. Поле формы (*Zabbix\Widgets\CWidgetField*) для каждого типа элемента ввода преобразует значение в единый формат для хранения в базе данных.

Note:
Файл **form** виджета должен находиться в каталоге *includes* (в рамках этого руководства — *ui/modules/lesson_gauge_chart/includes/*). Если файл имеет имя по умолчанию *WidgetForm.php*, регистрировать его в файле *manifest.json* не требуется. 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 ввести любой набор символов. 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 *CWidgetFieldTextBoxView*.

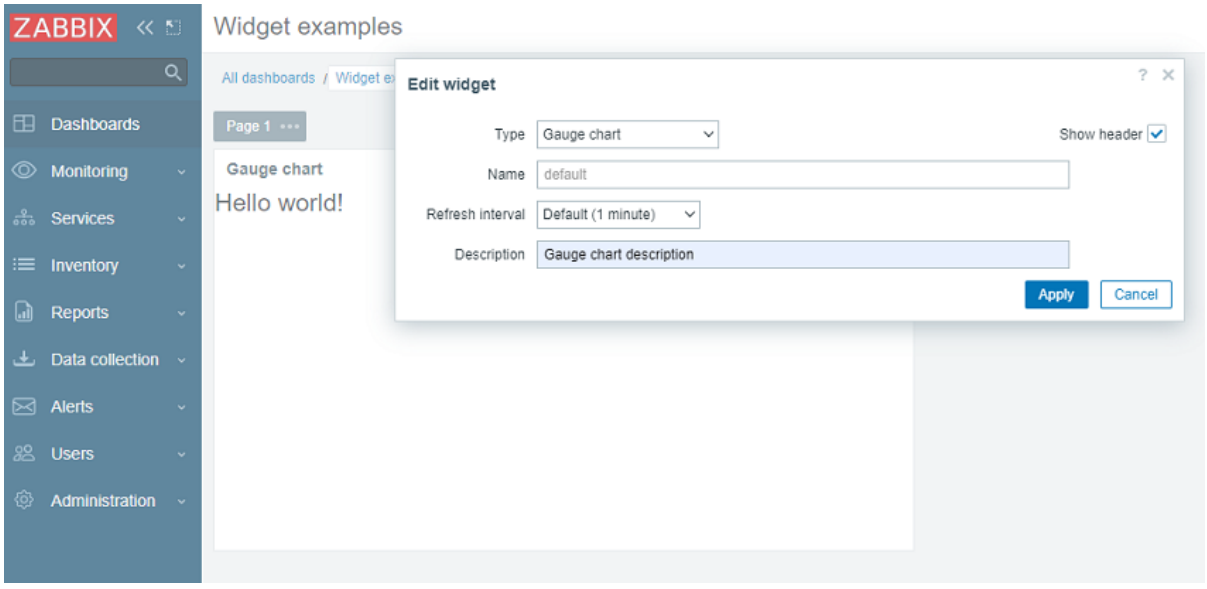
ui/modules/lesson_gauge_chart/views/widget.edit.php

```
<?php

/**
 * Gauge chart widget form view.
 *
 * @var CView $this
 * @var array $data
 */

(new CWidgetFormView($data))
    ->addField(
        new CWidgetFieldTextBoxView($data['fields']['description'])
    )
    ->show();
```

5. Перейдите на панель и нажмите на значок шестерёнки в виджете, чтобы открыть форму конфигурации виджета.
6. Форма конфигурации виджета теперь 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, а виджет по-прежнему показывает "Hello, world!".

Чтобы новое описание появилось в виджете, значение поля *Description* необходимо получить из базы данных и передать в представление виджета. Для этого нужно 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_chart.view*.

ui/modules/lesson_gauge_chart/manifest.json

```
{
    "manifest_version": 2.0,
    "id": "lesson_gauge_chart",
    "type": "widget",
    "name": "Gauge chart",
    "namespace": "LessonGaugeChart",
    "version": "1.0",
    "author": "Zabbix",
    "actions": {
        "widget.lesson_gauge_chart.view": {
            "class": "WidgetView"
        }
    }
}
```

11. 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']*.

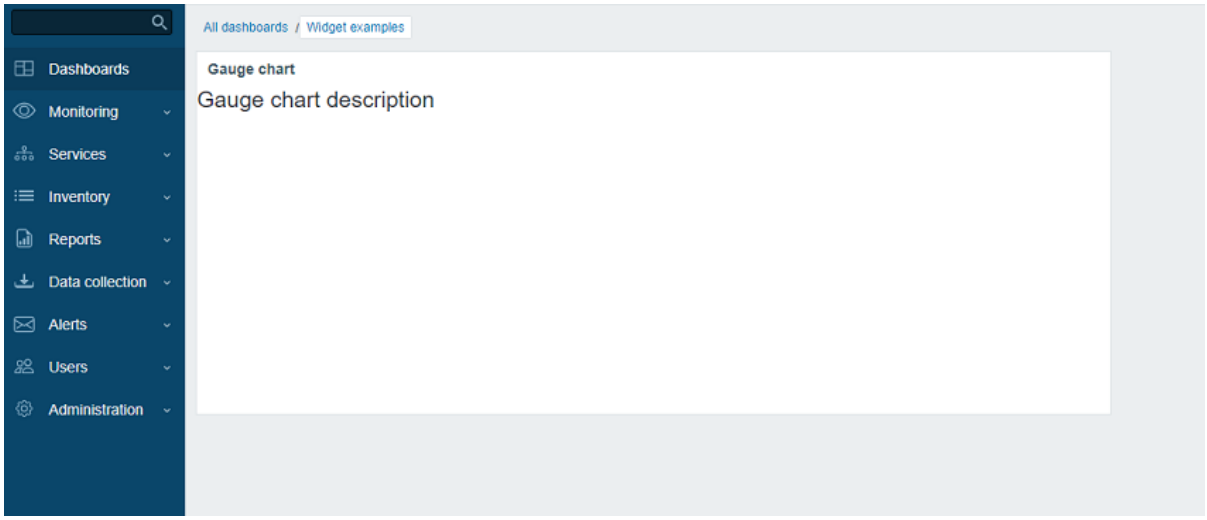
ui/modules/lesson_gauge_chart/views/widget.view.php

```
<?php

/**
 * Gauge chart widget view.
 *
 * @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!" должен отображаться текст описания виджета.



Получение значения элемента данных через API

Виджет должен показывать последнее значение выбранного пользователем элемента данных. Для этого необходимо 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. После этого контроллер виджета сможет получать данные элемента данных и его значение через API-запрос. После получения значение можно будет 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
};

/**
 *          gauge chart.
 */
class WidgetForm extends CWidgetForm {

    public function addFields(): self {
        return $this
            ->addField(
                (new CWidgetFieldMultiSelectItem('itemid', _('        ')))
                    ->setFlags(CWidgetField::FLAG_NOT_EMPTY | CWidgetField::FLAG_LABEL_ASTERISK)
                    ->setMultiple(false)
            )
            ->addField(
                new CWidgetFieldTextBox('description', _('     '))
            );
    }
}
```

2. Откройте *views/widget.edit.php* и добавьте визуальный компонент поля в представление конфигурации.

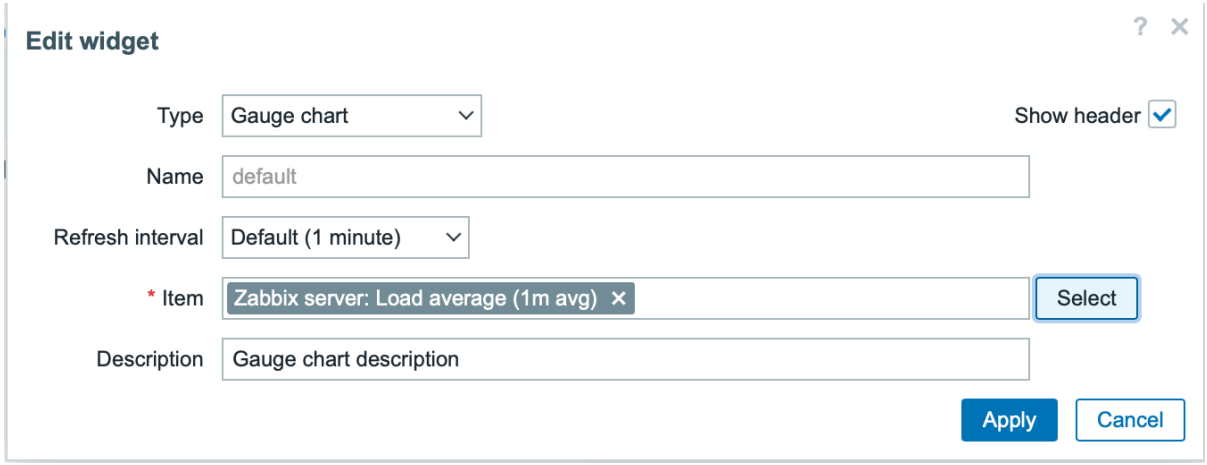
ui/modules/lesson_gauge_chart/views/widget.edit.php

```php
<?php

/**
 *                  gauge chart.
 *
 * @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 на значок шестерёнки в виджете, чтобы открыть форму конфигурации виджета.
4. Форма конфигурации виджета теперь содержит новое поле ввода *Элемент данных*. Выберите узел сети "Zabbix server" и элемент данных "Load average (1m avg)".



5. Нажмите *Применить* в форме конфигурации виджета. Затем нажмите *Сохранить изменения* в правом верхнем углу, чтобы сохранить панель.
6. Откройте и измените *actions/WidgetView.php*.

Теперь ID элемента данных будет доступен в контроллере виджета в *$this->fields_values['itemid']*. Метод контроллера *doAction()* собирает данные элемента данных (имя, тип значения, единицы измерения) с помощью метода API *item.get* и последнее значение элемента данных 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.

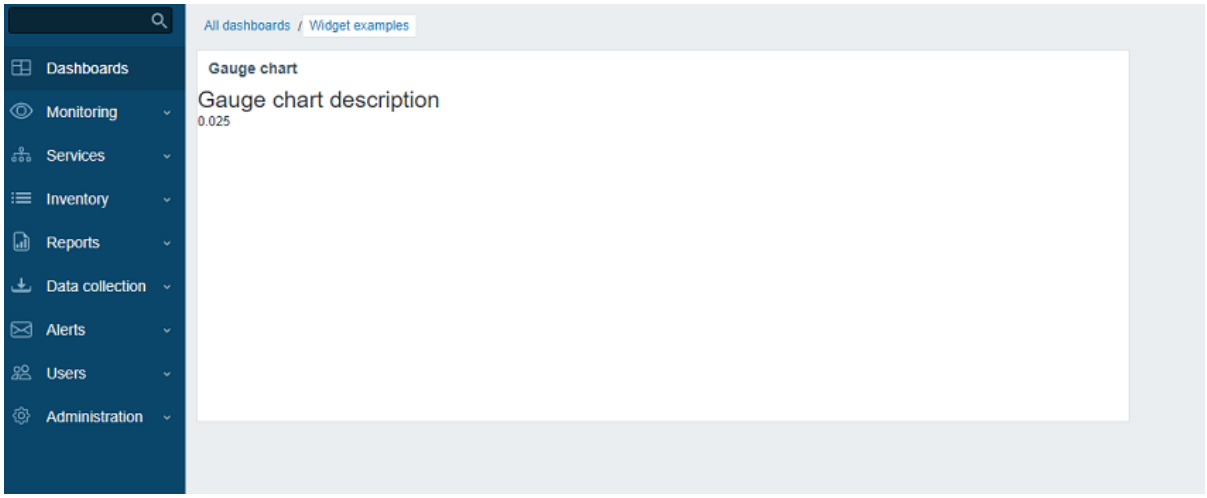
ui/modules/lesson_gauge_chart/views/widget.view.php

```
<?php

/**
 *              gauge chart.
 *
 * @var CView $this
 * @var array $data
 */

(new CWidgetView($data))
    ->addItem([
        new CTag('h1', true, $data['description']),
        new CDiv($data['value'] !== null ? $data['value']['value'] : _('      '))
    ])
    ->show();
```

8. Обновите страницу панели. Виджет отобразит последнее значение элемента данных.



Добавление расширенных настроек конфигурации в представление конфигурации

В этом разделе вы узнаете, как добавить разворачиваемый/сворачиваемый 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 добавить/переопределить настройки виджета по умолчанию (в данном случае — переводы). Приведённый ниже JavaScript отображает строку ”No data” в случае отсутствия данных. Строка ”No data” присутствует в файлах перевода интерфейса Zabbix.

Если имеются какие-либо константы виджета, 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 определите палитру цветов по умолчанию для выбора цвета, чтобы её 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.
 *
```

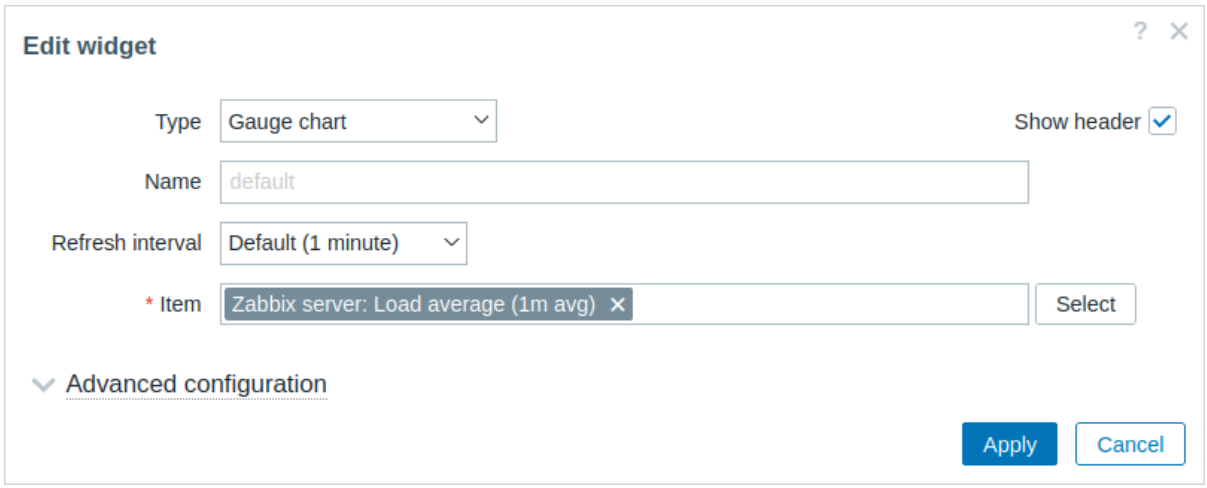
```
 * @var CView $this
 * @var array $data
 */

$lefty_units = new CWidgetFieldSelectView($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();
```

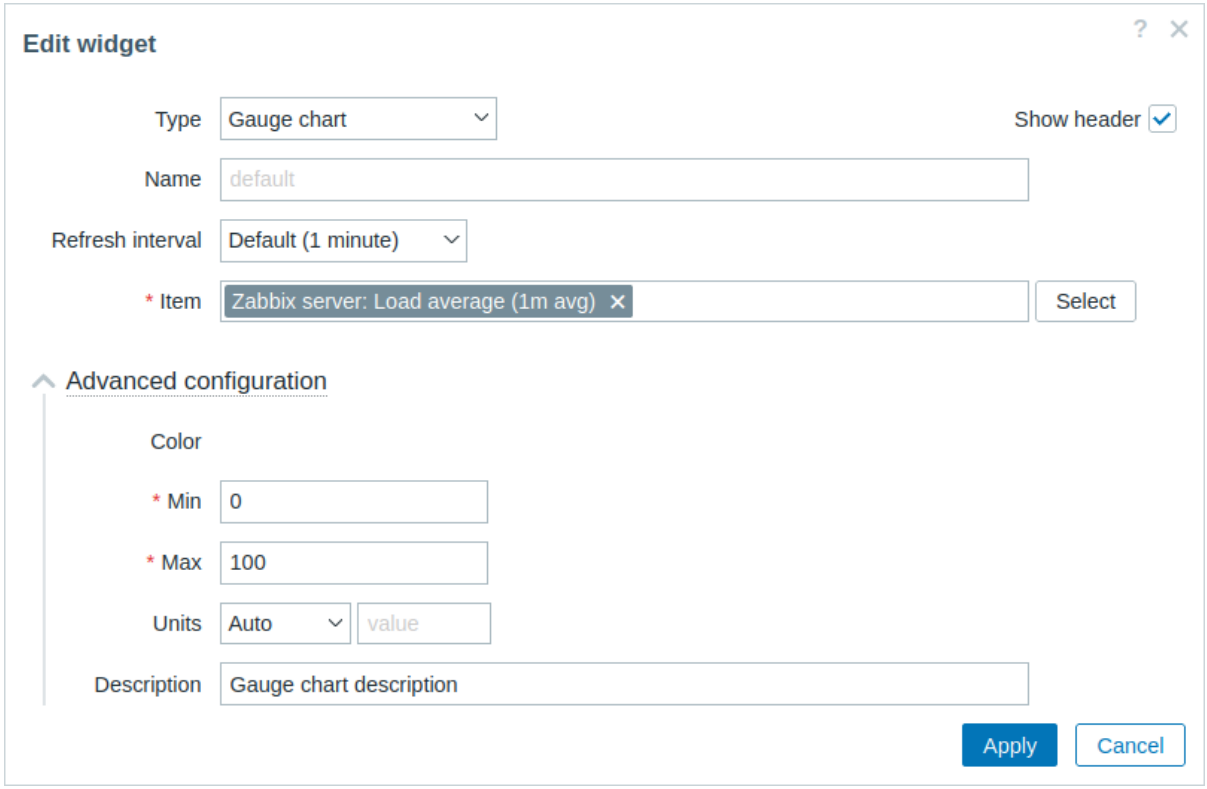
Note:
Метод *addField()* класса *CWidgetFormView* принимает строку CSS-класса в качестве второго параметра.

4. Вернитесь к панели мониторинга, переключитесь в режим редактирования и нажмите на значок шестерёнки в виджете, чтобы открыть форму конфигурации виджета. Теперь форма конфигурации виджета содержит новый разворачиваемый/сворачиваемый 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 быть инициализирован

с помощью JavaScript, который будет добавлен в следующем разделе — *Add JavaScript to the widget*.



Добавление JavaScript в виджет

В этом разделе вы узнаете, как добавить диаграмму gauge, созданную с помощью JavaScript, которая показывает, является ли последнее значение нормальным, слишком высоким или слишком низким.

1. Создайте файл *widget.edit.js.php* в каталоге *views*.

JavaScript будет отвечать за инициализацию выбора цвета в представлении конфигурации.

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 представление конфигурации. Для этого используйте метод *includeJsFile()*. Чтобы добавить встроенный JavaScript, используйте метод *addJavaScript()*.

ui/modules/lesson_gauge_chart/views/widget.edit.php

```
<?php

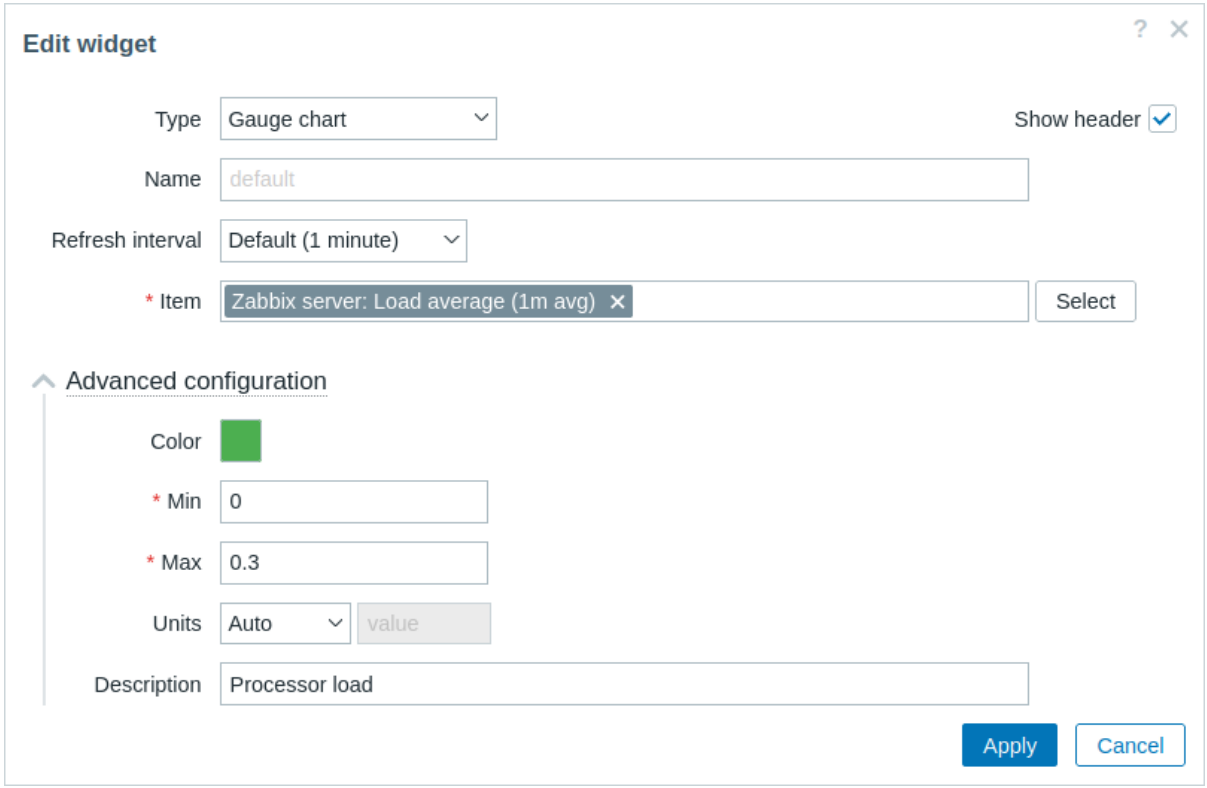
/**
 *                        gauge.
 *
 * @var CView $this
 * @var array $data
 */

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(_('              ')))
            ->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 и щелкните по значку шестеренки в виджете, чтобы открыть форму конфигурации виджета. Теперь разверните раздел *Расширенная конфигурация*, чтобы увидеть инициализированный выбор цвета. Заполните поля значениями и выберите цвет для диаграммы gauge.



4. Нажмите *Применить* в форме конфигурации виджета. Затем нажмите *Сохранить изменения* в правом верхнем углу, чтобы сохранить панель.

5. Откройте *actions/WidgetView.php* и обновите контроллер.

Свойство **$this->fields_values** теперь содержит значения всех полей *Расширенной конфигурации*. Завершите настройку контроллера, чтобы обеспечить передачу конфигурации и значения выбранного 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*.

Необходимо создать контейнер для диаграммы gauge, которую вы будете рисовать на следующих шагах, и контейнер для описания.

Чтобы передать значения в JavaScript как объект JSON, используйте метод *setVar()*.

ui/modules/lesson_gauge_chart/views/widget.view.php

```
<?php

/**
 *                    gauge.
 *
 * @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, любых 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 виджету всю необходимую информацию, вызывая соответствующие JavaScript-методы. Панель также ожидает, что виджет будет генерировать события при возникновении какого-либо взаимодействия. Таким образом, класс *CWidget* содержит набор методов с реализацией поведения виджета по умолчанию, которое можно настроить путем расширения класса.

В данном случае требуется некоторая настройка, поэтому будет реализована пользовательская логика для следующего поведения виджета:

- инициализация виджета, которая отвечает за определение начального состояния виджета (см. метод *onInitialize()*);
- отображение содержимого виджета (то есть отрисовка диаграммы gauge), 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 gauge будет использоваться реализация поведения виджета по умолчанию. Чтобы узнать больше о JavaScript-методах класса *CWidget*, см.: 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 стилей CSS в виджет

В этом разделе вы узнаете, как добавлять собственные стили CSS, чтобы виджет выглядел более привлекательно.

1. Для стилей виджетов создайте новый каталог *css* в каталоге *assets*.
2. Создайте файл *widget.css* в каталоге *assets/css*. Чтобы стилизовать элементы виджета, используйте селектор *div.dashboard-widget-{widget id}*. Чтобы настроить CSS для всего виджета, используйте селектор *form.dashboard-widget-{idget 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;
}
```

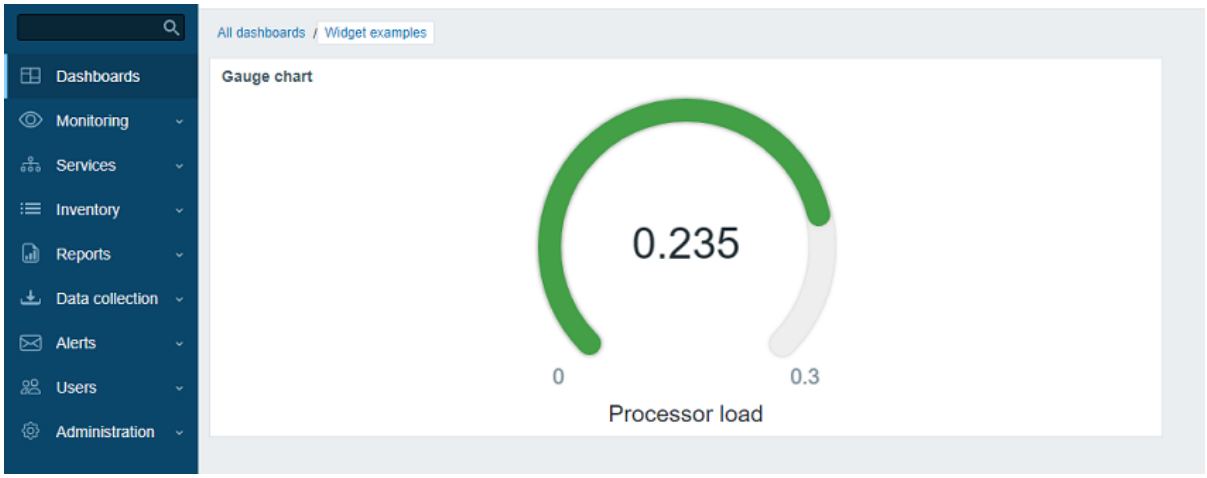
3. Откройте *manifest.json* и добавьте имя файла CSS (*widget.css*) в массив в разделе *assets/css*. Это позволит стилям CSS, определенным в *widget.css*, загружаться вместе со страницей панели.

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, 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

На этой странице перечислены все изменения (если таковые имеются), касающиеся разработки пользовательских расширений Zabbix.

Справочные страницы Zabbix

Справочная документация по инструментам и службам Zabbix командной строки со списком доступных команд, параметров и примеров использования.

zabbix_agent2

Раздел: Команды обслуживания (8)
Обновлено: 2019-01-29
Оглавление Вернуться на главную

НАИМЕНОВАНИЕ

zabbix_agent2 — Zabbix агент 2

СИНТАКСИС

zabbix_agent2 [**-c** *config-file*]
zabbix_agent2 [**-c** *config-file*] [**-v**] **-p**
zabbix_agent2 [**-c** *config-file*] [**-v**] **-t** *item-key*
zabbix_agent2 [**-c** *config-file*] **-R** *runtime-option*
zabbix_agent2 [**-c** *config-file*] **-T**
zabbix_agent2 -h
zabbix_agent2 -V

ОПИСАНИЕ

zabbix_agent2 — это приложение для мониторинга параметров различных сервисов.

ПАРАМЕТРЫ

-c, **--config** *config-file* Используйте альтернативный *config-file* вместо файла по умолчанию.

-R, **--runtime-control** *runtime-option* Выполняйте административные функции в соответствии с *runtime-option*.

Параметры управления во время выполнения: userparameter_reload
Перезагрузить пользовательские параметры из файла конфигурации

log_level_increase
Увеличить уровень логирования

log_level_decrease
Уменьшить уровень логирования

help
Вывести список доступных параметров управления во время выполнения

metrics
Вывести список доступных метрик

version
Показать версию

-p, **--print**
Вывести известные элементы данных и завершить работу. Для каждого элемента данных используются либо общие значения по умолчанию, либо специальные значения по умолчанию для тестирования. Эти значения по умолчанию указаны в квадратных скобках как параметры ключа элемента данных. Возвращаемые значения заключаются в квадратные скобки и предваряются типом возвращаемого значения, отделённым символом вертикальной черты. Для пользовательских параметров тип всегда **t**, так как агент не может определить все возможные возвращаемые значения. Для элементов данных, отображаемых как рабочие, не гарантируется работа с сервера Zabbix или zabbix_get при запросе к запущенному демону агента, так как права доступа или окружение могут отличаться. Типы возвращаемых значений:

d
Число с десятичной частью.

m
Не поддерживается. Это может быть вызвано запросом элемента данных, который работает только в активном режиме, например элемента данных мониторинга журнала, или элемента данных, которому требуется несколько собранных значений. Проблемы с правами доступа или некорректные пользовательские параметры также могут привести к состоянию «не поддерживается».

s
Текст. Максимальная длина не ограничена.

t
Текст. То же, что и **s**.

u
Беззнаковое целое число.

-t, **--test** *item-key*
Проверить один элемент данных и завершить работу. Описание вывода см. в **--print**.

-v, **--verbose**
Включить подробный вывод при проверке элемента данных или выводе известных элементов данных.

-T, **--test-config**
Проверить файл конфигурации и завершить работу.

-h, **--help**
Показать эту справку и завершить работу.

-V, **--version**
Вывести информацию о версии и завершить работу.

ФАЙЛЫ

/usr/local/etc/zabbix_agent2.conf
Расположение по умолчанию файла конфигурации Zabbix агент 2 (если не было изменено во время компиляции).

СМОТРИТЕ ТАКЖЕ

Документация https://www.zabbix.com/manuals

zabbix_agentd(8), **zabbix_get**(8), **zabbix_js**(8), **zabbix_proxy**(8), **zabbix_sender**(8), **zabbix_server**(8)

Указатель

ИМЯ

СИНТАКСИС

ОПИСАНИЕ

ПАРАМЕТРЫ

ФАЙЛЫ

СМ. ТАКЖЕ

АВТОР

Этот документ был создан: 14:07:57 GMT, 22 ноября 2021 г.

zabbix_agentd

Раздел: Команды обслуживания (8)
Обновлено: 2019-01-29
Оглавление Вернуться на главную

НАИМЕНОВАНИЕ

zabbix_agentd — демон Zabbix агент

СИНТАКСИС

zabbix_agentd [**-c** *config-file*]
zabbix_agentd [**-c** *config-file*] **-p**
zabbix_agentd [**-c** *config-file*] **-t** *item-key*
zabbix_agentd [**-c** *config-file*] **-R** *runtime-option*
zabbix_agentd -h
zabbix_agentd -V

ОПИСАНИЕ

zabbix_agentd — демон (служба) для мониторинга различных параметров сервера.

ПАРАМЕТРЫ

-c, **--config** *config-file*
Использовать альтернативный файл конфигурации *config-file* вместо файла по умолчанию.

-f, **--foreground**
Запустить агент Zabbix на переднем плане (отключить переход в фоновый режим).

-R, **--runtime-control** *runtime-option*
Выполнить административные функции в соответствии с *runtime-option*.

Параметры управления во время выполнения userparameter_reload
Перезагрузить пользовательские параметры из файла конфигурации

log_level_increase[=*target*]
Увеличить уровень журналирования; влияет на все процессы, если target не указан

log_level_decrease[=*target*]
Уменьшить уровень журналирования; влияет на все процессы, если target не указан

Цели управления уровнем журналирования *process-type*
Все процессы указанного типа (active checks, collector, listener)

process-type,N
Тип процесса и номер (например, listener,3)

pid
Идентификатор процесса, до 65535. Для больших значений укажите цель как ”process-type,N”

-p, **--print**
Вывести известные элементы данных и завершить работу. Для каждого элемента данных используются либо общие значения по умолчанию, либо предоставляются специальные значения по умолчанию для тестирования. Эти значения по умолчанию перечислены в квадратных скобках как параметры ключа элемента данных. Возвращаемые значения заключаются в квадратные скобки и предваряются типом возвращаемого значения, отделённым символом вертикальной черты. Для пользовательских параметров тип всегда **t**, так как агент не может определить все возможные возвращаемые значения. Элементы данных, отображаемые как рабочие, не гарантированно будут работать с сервера Zabbix или zabbix_get при запросе к работающему демону агента, так как права доступа или окружение могут отличаться. Типы возвращаемых значений:

d
Число с десятичной частью.

m
Не поддерживается. Это может быть вызвано запросом элемента данных, который работает только в активном режиме, например элемента данных мониторинга журнала, или элемента данных, которому требуется несколько собранных значений. Проблемы с правами доступа или некорректные пользовательские параметры также могут привести к состоянию ”не поддерживается”.

s
Текст. Максимальная длина не ограничена.

t
Текст. То же, что и **s**.

u
Беззнаковое целое число.

-t, **--test** *item-key*
Проверить один элемент данных и завершить работу. Описание вывода см. в **--print**.

-T, **--test-config**
Проверить файл конфигурации и завершить работу.

-h, **--help**
Показать эту справку и завершить работу.

-V, **--version**
Вывести информацию о версии и завершить работу.

ФАЙЛЫ

/usr/local/etc/zabbix_agentd.conf
Расположение по умолчанию для файла конфигурации агента Zabbix (если не было изменено во время компиляции).

СМОТРИТЕ ТАКЖЕ

Документация https://www.zabbix.com/manuals

zabbix_agent2(8), **zabbix_get**(1), **zabbix_js**(1), **zabbix_proxy**(8), **zabbix_sender**(1), **zabbix_server**(8)

Содержание

НАЗВАНИЕ

СИНТАКСИС

ОПИСАНИЕ

ПАРАМЕТРЫ

ФАЙЛЫ

СМ. ТАКЖЕ

АВТОР

Этот документ был создан: 20:50:13 GMT, 22 ноября 2021 г.

zabbix_get

Раздел: Пользовательские команды (1)
Обновлено: 2022-01-06
Index Вернуться к основному содержанию

НАИМЕНОВАНИЕ

zabbix_get — утилита Zabbix для получения метрик

СИНТАКСИС

zabbix_get -s *имя-узла-сети-или-IP* [**-p** *номер-порта*] [**-I** *IP-адрес*] [**-t** *тайм-аут*] **-k** *ключ-элемента-данных*
zabbix_get -s *имя-узла-сети-или-IP* [**-p** *номер-порта*] [**-I** *IP-адрес*] [**-t** *тайм-аут*] **--tls-connect cert --tls-ca-file** *CA-файл* [**--tls-crl-file** *CRL-file*] [**--tls-agent-cert-issuer** *эмитент*] [**--tls-agent-cert-subject** *субъект*] **--tls-cert-file** *файл-сертификата* **--tls-key-file** *файл-ключа* [**--tls-cipher13** *строка-шифра*] [**--tls-cipher** *строка-шифра*] **-k** *ключ-элемента-данных*
zabbix_get -s *имя-узла-сети-или-IP* [**-p** *номер-порта*] [**-I** *IP-адрес*] [**-t** *тайм-аут*] **--tls-connect psk --tls-psk-identity** *идентификатор-PSK* **--tls-psk-file** *файл-PSK* [**--tls-cipher13** *строка-шифра*] [**--tls-cipher** *строка-шифра*] **-k** *ключ-элемента-данных*
zabbix_get -h
zabbix_get -V

ОПИСАНИЕ

zabbix_get — это утилита командной строки для получения данных от агента Zabbix.

ПАРАМЕТРЫ

-s, **--host** *host-name-or-IP*
Укажите имя узла сети или IP-адрес узла сети.

-p, **--port** *port-number*
Укажите номер порта агента, запущенного на узле сети. По умолчанию используется 10050.

-I, **--source-address** *IP-address*
Укажите исходный IP-адрес.

-t, **--timeout** *seconds*
Укажите тайм-аут. Допустимый диапазон: 1-600 секунд (по умолчанию: 30)

-k, **--key** *item-key*
Укажите ключ элемента данных, для которого нужно получить значение.

-P, **--protocol** *value*
Протокол, используемый для связи с агентом. Значения:

auto подключение с использованием протокола JSON, с возвратом и повторной попыткой с использованием текстового протокола (по умолчанию)

json подключение с использованием протокола JSON

plaintext подключение с использованием текстового протокола, при котором передаётся только ключ элемента данных (выпуски 6.4.x и более ранние)

--tls-connect *value*
Способ подключения к агенту. Значения:

без шифрования
подключение без шифрования (по умолчанию)

psk
подключение с использованием TLS и предварительно общего ключа

cert
подключение с использованием TLS и сертификата

--tls-ca-file *CA-file*
Полный путь к файлу, содержащему сертификаты корневого(ых) центра(ов) сертификации для проверки сертификата удалённой стороны.

--tls-crl-file *CRL-file*
Полный путь к файлу, содержащему отозванные сертификаты.

--tls-agent-cert-issuer *cert-issuer*
Разрешённый издатель сертификата агента.

--tls-agent-cert-subject *cert-subject*
Разрешённый субъект сертификата агента.

--tls-cert-file *cert-file*
Полный путь к файлу, содержащему сертификат или цепочку сертификатов.

--tls-key-file *key-file*
Полный путь к файлу, содержащему закрытый ключ.

--tls-psk-identity *PSK-identity*
Строка PSK-идентификатора.

--tls-psk-file *PSK-file*
Полный путь к файлу, содержащему предварительно общий ключ.

--tls-cipher13 *cipher-string*
Строка шифров для OpenSSL 1.1.1 или новее для TLS 1.3.
Переопределяет критерии выбора набора шифров по умолчанию.
Этот параметр недоступен, если версия OpenSSL ниже 1.1.1.

--tls-cipher *cipher-string*
Строка приоритетов GnuTLS (для TLS 1.2 и выше) или строка шифров OpenSSL
(только для TLS 1.2).
Переопределяет критерии выбора набора шифров по умолчанию.

-h, **--help**
Показать эту справку и выйти.

-V, **--version**
Вывести информацию о версии и выйти.

ПРИМЕРЫ

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 Operations,O=Example Corp,DC=example,DC=com” --tls-cert-file /home/zabbix/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

СМОТРИТЕ ТАКЖЕ

Документация https://www.zabbix.com/manuals

zabbix_agentd(8), **zabbix_proxy**(8), **zabbix_sender**(1), **zabbix_server**(8), **zabbix_js**(1), **zabbix_agent2**(8), **zabbix_web_service**(8)

Содержание

НАЗВАНИЕ

СИНТАКСИС

ОПИСАНИЕ

ПАРАМЕТРЫ

ПРИМЕРЫ

СМ. ТАКЖЕ

АВТОР

Этот документ был создан: 08:42:29 GMT, 11 июня 2021 г.

zabbix_js

Раздел: Пользовательские команды (1)
Обновлено: 2022-01-06
Index Вернуться к основному содержанию

НАИМЕНОВАНИЕ

zabbix_js — утилита Zabbix для выполнения скриптов JavaScript

СИНТАКСИС

zabbix_js -s *файл-скрипта* **-p** *входной-параметр* [**-l** *log-level*] [**-t** *timeout*]
zabbix_js -s *файл-скрипта* **-i** *входной-файл* [**-l** *log-level*] [**-t** *тайм-аут*]
zabbix_js -h
zabbix_js -V

ОПИСАНИЕ

zabbix_js — это утилита командной строки, которую можно использовать для тестирования встроенных скриптов.

ПАРАМЕТРЫ

-s, **--script** *script-file*
Укажите имя файла скрипта для выполнения. Если в качестве имени файла указано '-', скрипт будет прочитан из stdin.

-p, **--param** *input-param*
Укажите входной параметр.

-i, **--input** *input-file*
Укажите имя файла входного параметра. Если в качестве имени файла указано '-', входные данные будут прочитаны из stdin.

-w, **--webdriver** *url*
Укажите URL webdriver.

-l, **--loglevel** *log-level*
Укажите уровень логирования.

-t, **--timeout** *timeout*
Укажите тайм-аут в секундах. Допустимый диапазон: 1-600 секунд (по умолчанию: 10)

-h, **--help**
Показать эту справку и выйти.

-V, **--version**
Вывести информацию о версии и выйти.

ПРИМЕРЫ

zabbix_js -s script-file.js -p example

СМОТРИТЕ ТАКЖЕ

Документация https://www.zabbix.com/manuals

zabbix_agent2(8), **zabbix_agentd**(8), **zabbix_get**(1), **zabbix_proxy**(8), **zabbix_sender**(1), **zabbix_сервер**(8)

Оглавление

НАИМЕНОВАНИЕ

СИНТАКСИС

ОПИСАНИЕ

ПАРАМЕТРЫ

ПРИМЕРЫ

СМОТРИТЕ ТАКЖЕ

Этот документ был создан: 18 Марта 2020 года, 21:23:35 GMT

zabbix_proxy

Раздел: Команды обслуживания (8)
Обновлено: 2022-09-04
Оглавление Вернуться на главную

НАИМЕНОВАНИЕ

zabbix_proxy — демон сервиса zabbix прокси

СИНТАКСИС

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

ОПИСАНИЕ

zabbix_proxy — это демон, который собирает данные мониторинга с устройств и отправляет их на сервер Zabbix.

ПАРАМЕТРЫ

-c, **--config** *config-file*
Использовать альтернативный файл конфигурации *config-file* вместо файла по умолчанию.

-f, **--foreground**
Запустить zabbix прокси без перехода в фоновый режим.

-R, **--runtime-control** *runtime-option*
Выполнить административные функции в соответствии с *runtime-option*.

Параметры управления во время выполнения config_cache_reload
Перезагрузить кэш конфигурации.
Игнорируется, если кэш в данный момент загружается.
Активный прокси Zabbix подключится к серверу Zabbix и запросит данные конфигурации.
Для поиска PID-файла и отправки сигнала процессу, указанному в PID-файле, будет использован файл конфигурации по умолчанию (если не указана опция **-c**).

snmp_cache_reload
Перезагрузить кэш SNMP.
Обратите внимание, что Zabbix кэширует сопоставления SNMPv3 EngineID→IP и может автоматически повторно использовать EngineID для уменьшения накладных расходов на опрос.

housekeeper_execute
Выполнить housekeeper.
Игнорируется, если housekeeper в данный момент уже выполняется.

diaginfo[=*section*]
Записать во внутренний журнал диагностическую информацию указанного раздела.
Разделом может быть *historycache*, *preprocessing*, *locks*.
По умолчанию записывается диагностическая информация всех разделов.

log_level_increase[=*target*]
Увеличить уровень журналирования; если target не указан, влияет на все процессы.

log_level_decrease[=*target*]
Уменьшить уровень журналирования; если target не указан, влияет на все процессы.

Цели управления уровнем журналирования *process-type*
Все процессы указанного типа (availability manager, browser poller, configuration syncer, data sender, discovery manager, history syncer, housekeeper, http poller, icmp pinger, 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
Тип процесса и номер (например, poller,3)

pid
Идентификатор процесса, до 65535. Для больших значений укажите цель как ”process-type,N”

Цели управления профилированием

process-type
Все процессы указанного типа (configuration syncer, data sender, discovery manager, history syncer, housekeeper, http poller, preprocessing manager, icmp pinger, ipmi manager, ipmi poller, java poller, poller, agent poller, http agent poller, snmp poller, self-monitoring, snmp trapper, task manager, trapper, unreachable poller, vmware collector, history poller, availability manager, odbc poller)

process-type,N
Тип процесса и номер (например, history syncer,1)

pid
Идентификатор процесса, до 65535. Для больших значений укажите цель как ”process-type,N”

scope
Область профилирования (rwlock, mutex, processing) может использоваться с типом процесса (например, history syncer,1,processing)

-T, **--test-config**
Проверить файл конфигурации и выйти.

-h, **--help**
Показать эту справку и выйти.

-V, **--version**
Вывести информацию о версии и выйти.

ФАЙЛЫ

/usr/local/etc/zabbix_proxy.conf
Расположение по умолчанию файла конфигурации прокси Zabbix (если не было изменено во время компиляции).

СМОТРИТЕ ТАКЖЕ

Документация https://www.zabbix.com/manuals

zabbix_agentd(8), **zabbix_get**(1), **zabbix_sender**(1), **zabbix_server**(8), **zabbix_js**(1), **zabbix_agent2**(8)

Содержание

ИМЯ

СИНТАКСИС

ОПИСАНИЕ

ПАРАМЕТРЫ

ФАЙЛЫ

СМ. ТАКЖЕ

АВТОР

Этот документ был создан: 16:12:22 GMT, 04 сентября 2020 г.

zabbix_sender

Раздел: Пользовательские команды (1)
Обновлено: 2021-06-01
Оглавление Вернуться на главную

НАИМЕНОВАНИЕ

zabbix_sender — утилита Zabbix для отправки метрик

СИНТАКСИС

zabbix_sender [**-v**] **-z** *сервер* [**-p** *порт*] [**-I** *IP-адрес*] [**-t** *тайм-аут*] **-s** *узел сети* **-k** *ключ* **-o** *значение*
zabbix_sender [**-v**] **-z** *сервер* [**-p** *порт*] [**-I** *IP-адрес*] [**-t** *тайм-аут*] [**-s** *узел сети*] [**-T**] [**-N**] [**-r**] [**-g**] **-i** *входной-файл*
zabbix_sender [**-v**] **-c** *config-file* [**-z** *сервер*] [**-p** *порт*] [**-I** *IP-адрес*] [**-t** *тайм-аут*] [**-s** *узел сети*] **-k** *ключ* **-o** *значение*
zabbix_sender [**-v**] **-c** *config-file* [**-z** *сервер*] [**-p** *порт*] [**-I** *IP-адрес*] [**-t** *тайм-аут*] [**-s** *узел сети*] [**-T**] [**-N**] [**-r**] [**-g**] **-i** *входной-файл*
zabbix_sender [**-v**] **-z** *сервер* [**-p** *порт*] [**-I** *IP-адрес*] [**-t** *тайм-аут*] **-s** *узел сети* **--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** *ключ* **-o** *значение*
zabbix_sender [**-v**] **-z** *сервер* [**-p** *порт*] [**-I** *IP-адрес*] [**-t** *тайм-аут*] [**-s** *узел сети*] **--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**] [**-g**] **-i** *входной-файл*
zabbix_sender [**-v**] **-c** *config-file* [**-z** *сервер*] [**-p** *порт*] [**-I** *IP-адрес*] [**-t** *тайм-аут*] [**-s** *узел сети*] **--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** *ключ* **-o** *значение*
zabbix_sender [**-v**] **-c** *config-file* [**-z** *сервер*] [**-p** *порт*] [**-I** *IP-адрес*] [**-t** *тайм-аут*] [**-s** *узел сети*] **--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**] [**-g**] **-i** *входной-файл*
zabbix_sender [**-v**] **-z** *сервер* [**-p** *порт*] [**-I** *IP-адрес*] [**-t** *тайм-аут*] **-s** *узел сети* **--tls-connect psk** **--tls-psk-identity** *PSK-identity* **--tls-psk-file** *PSK-file* [**--tls-cipher13** *cipher-string*] [**--tls-cipher** *cipher-string*] **-k** *ключ* **-o** *значение*
zabbix_sender [**-v**] **-z** *сервер* [**-p** *порт*] [**-I** *IP-адрес*] [**-t** *тайм-аут*] [**-s** *узел сети*] **--tls-connect psk** **--tls-psk-identity** *PSK-identity* **--tls-psk-file** *PSK-file* [**--tls-cipher13** *cipher-string*] [**--tls-cipher** *cipher-string*] [**-T**] [**-N**] [**-r**] [**-g**] **-i** *входной-файл*
zabbix_sender [**-v**] **-c** *config-file* [**-z** *сервер*] [**-p** *порт*] [**-I** *IP-адрес*] [**-t** *тайм-аут*] [**-s** *узел сети*] **--tls-connect psk** **--tls-psk-identity** *PSK-identity* **--tls-psk-file** *PSK-file* [**--tls-cipher13** *cipher-string*] [**--tls-cipher** *cipher-string*] **-k** *ключ* **-o** *значение*
zabbix_sender [**-v**] **-c** *config-file* [**-z** *сервер*] [**-p** *порт*] [**-I** *IP-адрес*] [**-t** *тайм-аут*] [**-s** *узел сети*] **--tls-connect psk** **--tls-psk-identity** *PSK-identity* **--tls-psk-file** *PSK-file* [**--tls-cipher13** *cipher-string*] [**--tls-cipher** *cipher-string*] [**-T**] [**-N**] [**-r**] [**-g**] **-i** *входной-файл*

zabbix_sender -h
zabbix_sender -V

ОПИСАНИЕ

zabbix_sender — это утилита командной строки для отправки значений мониторинга на Zabbix сервер или Zabbix прокси. На Zabbix сервере должен быть создан элемент данных типа **Zabbix траппер** с соответствующим ключом. Обратите внимание, что входящие данные будут приняты только от узлов сети, указанных в поле **Разрешённые хосты** для этого элемента данных.

ПАРАМЕТРЫ

-c, **--config** *config-file*
Использовать *config-file*. **Zabbix sender** считывает сведения о сервере из файла конфигурации agentd. По умолчанию **Zabbix sender** не считывает никакой файл конфигурации. Поддерживаются только параметры **Hostname**, **ServerActive**, **SourceIP**, **TLSConnect**, **TLSCAFile**, **TLSCRLFile**, **TLSServerCertIssuer**, **TLSServerCertSubject**, **TLSCertFile**, **TLSKeyFile**, **TLSPSKIdentity** и **TLSPSKFile**. Имя узла сети, заданное через параметр **HostnameItem**, не будет использовано; в этом случае имя узла сети следует указать через командную строку (см. параметр -s). Для отправки данных используются все адреса, определённые в параметре конфигурации **ServerActive** агента. Если отправка пакета данных на один адрес завершается неудачей, следующие пакеты на этот адрес не отправляются.

-z, **--zabbix-server** *server*
Имя узла или IP-адрес сервера Zabbix. Если узел сети мониторится через прокси, вместо этого следует использовать имя узла или IP-адрес прокси. При использовании вместе с **--config** переопределяет записи параметра **ServerActive**, указанные в файле конфигурации agentd.

-p, **--port** *port*
Указать номер порта trapper-процесса сервера Zabbix, работающего на сервере. По умолчанию используется 10051. При использовании вместе с **--config** переопределяет записи порта параметра **ServerActive**, указанные в файле конфигурации agentd.

-I, **--source-address** *IP-address*
Указать исходный IP-адрес. При использовании вместе с **--config** переопределяет параметр **SourceIP**, указанный в файле конфигурации agentd.

-t, **--timeout** *seconds*
Указать тайм-аут. Допустимый диапазон: 1-300 секунд (по умолчанию: 60)

-s, **--host** *host*
Указать имя узла сети, которому принадлежит элемент данных (как зарегистрировано в веб-интерфейсе Zabbix). IP-адрес узла сети и DNS-имя не подойдут. При использовании вместе с **--config** переопределяет параметр **Hostname**, указанный в файле конфигурации agentd.

-k, **--key** *key*
Указать ключ элемента данных, в который следует отправить значение.

-o, **--value** *value*
Указать значение элемента данных.

-i, **--input-file** *input-file*
Загрузить значения из входного файла. Укажите **-** в качестве **<input-file>**, чтобы считывать значения из стандартного ввода. Каждая строка файла содержит разделённые пробелами поля: **<hostname> <key> <value>**. Каждое значение должно быть указано в отдельной строке. Каждая строка должна содержать 3 поля, разделённые пробелами: **<hostname> <key> <value>**, где ”hostname” — это имя мониторируемого узла сети, как зарегистрировано в веб-интерфейсе Zabbix, ”key” — это ключ целевого элемента данных, а ”value” — отправляемое значение. Укажите **-** в качестве **<hostname>**, чтобы использовать имя узла сети из файла конфигурации агента или из аргумента **--host**.

Пример строки входного файла:

”Linux DB3” db.connections 43

Тип значения должен быть правильно задан в конфигурации элемента данных в веб-интерфейсе Zabbix. Zabbix sender отправляет до 250 значений в одном соединении. Ограничение размера для отправки значений из входного файла зависит от размера, описанного в протоколе обмена данными Zabbix. Содержимое входного файла должно быть в кодировке UTF-8. Все значения из входного файла отправляются последовательно, сверху вниз. Записи должны быть отформатированы по следующим правилам:

- Поддерживаются записи в кавычках и без кавычек.
- Символ двойной кавычки используется как символ кавычек.
- Записи, содержащие пробелы, должны быть заключены в кавычки.

- Символы двойной кавычки и обратной косой черты внутри записи в кавычках должны экранироваться обратной косой чертой.
- Экранирование не поддерживается в записях без кавычек.
- В строках в кавычках поддерживаются escape-последовательности перевода строки (\n).
- Escape-последовательности перевода строки удаляются из конца записи.

-T, **--with-timestamps**
Этот параметр можно использовать только вместе с параметром **--input-file**.

Каждая строка входного файла должна содержать 4 поля, разделённые пробелами: **<hostname> <key> <timestamp> <value>**. Метка времени должна быть указана в формате Unix timestamp. Если целевой элемент данных имеет триггеры, ссылающиеся на него, все метки времени должны идти в возрастающем порядке, иначе вычисление событий будет некорректным.

Пример строки входного файла:

”Linux DB3” db.connections 1429533600 43

Подробнее см. параметр **--input-file**.

Если значение с меткой времени отправляется для узла сети, находящегося в обслуживании типа “без данных”, это значение будет отброшено; однако можно отправить значение с меткой времени для уже завершившегося периода обслуживания, и оно будет принято.

-N, **--with-ns**
Этот параметр можно использовать только вместе с параметром **--with-timestamps**.

Каждая строка входного файла должна содержать 5 полей, разделённых пробелами: **<hostname> <key> <timestamp> <ns> <value>**.

Пример строки входного файла:

”Linux DB3” db.connections 1429533600 7402561 43

Подробнее см. параметр **--input-file**.

-r, **--real-time**
Отправлять значения по одному сразу после их получения. Это можно использовать при чтении из стандартного ввода.

-g, **--group**
Группировать значения по узлам сети и отправлять для каждого узла сети отдельным пакетом.

--tls-connect *value*
Как подключаться к серверу или прокси. Значения:

unencrypted
подключаться без шифрования (по умолчанию)

psk
подключение с использованием TLS и общего ключа (pre-shared key)

cert
подключение с использованием TLS и сертификата

--tls-ca-file *CA-file*
Полный путь к файлу, содержащему сертификаты CA верхнего уровня для проверки сертификата партнёра.

--tls-crl-file *CRL-file*
Полный путь к файлу, содержащему отозванные сертификаты.

--tls-server-cert-issuer *эмитент*
Разрешённый эмитент сертификата сервера.

--tls-server-cert-subject *субъект*
Разрешённый субъект сертификата сервера.

--tls-cert-file *cert-file*
Полный путь к файлу, содержащему сертификат или цепочку сертификатов.

--tls-key-file *key-file*
Полный путь к файлу, содержащему закрытый ключ (private key).

--tls-psk-identity *PSK-identity*
Строка идентификации PSK.

--tls-psk-file *PSK-file*
Полный путь к файлу, содержащему общий ключ (pre-shared key).

--tls-cipher13 *cipher-string*
Строка шифра для OpenSSL 1.1.1 или новее для TLS 1.3. Переопределить критерии выбора набора шифров по умолчанию. Эта опция недоступна, если версия OpenSSL ниже 1.1.1.

--tls-cipher *cipher-string*
Строка приоритета GnuTLS (для TLS 1.2 и выше) или строка шифра OpenSSL (только для TLS 1.2). Переопределить критерии выбора набора шифров по умолчанию.

-v, **--verbose**
Подробный режим, **-vv** для ещё более подробного.

-h, **--help**
Вывести справочную информацию и выйти.

-V, **--version**
Вывести информацию о версии и выйти.

СТАТУС ЗАВЕРШЕНИЯ

Статус завершения равен 0, если значения были отправлены и все они были успешно обработаны сервером. Если данные были отправлены, но обработка хотя бы одного из значений не удалась, статус завершения равен 2. Если отправка данных не удалось, статус завершения будет 1.

ПРИМЕРЫ

zabbix_sender -c /etc/zabbix/zabbix_agentd.conf -k mysql.queries -o 342,45

Отправить **342.45** в качестве значения элемента данных **mysql.queries** наблюдаемого узла сети. Будут использованы отслеживаемый узел сети и сервер Zabbix, определённые в конфигурационном файле агента.

zabbix_sender -c /etc/zabbix/zabbix_agentd.conf -s ”Наблюдаемый узел сети” -k mysql.queries -o 342.45

Отправить **342.45** в качестве значения элемента данных **mysql.queries** узла сети **Наблюдаемый узел сети**, используя сервер Zabbix, определённый в конфигурационном файле агента.

zabbix_sender -z 192.168.1.113 -i data_values.txt

Отправить значения из файла **data_values.txt** на сервер Zabbix с IP **192.168.1.113**. Имена узлов и ключи определяются в файле.

echo ”- hw.serial.number 1287872261 SQ4321ASDF” | zabbix_sender -c /usr/local/etc/zabbix_agentd.conf -T -i -

Отправить значение с отметкой времени из командной строки на сервер Zabbix, указанный в конфигурационном файле агента. Знак минуса во входных данных указывает, что имя узла сети также должно использоваться из того же конфигурационного файла.

echo ’”Zabbix сервер” trapper.item ””’ | zabbix_sender -z 192.168.1.113 -p 10000 -i -

Отправить пустое значение элемента данных на сервер Zabbix с IP-адресом **192.168.1.113** на порт **10000** из командной строки. Пустые значения должны быть обозначены пустой строкой в двойных кавычках.

zabbix_sender -z 192.168.1.113 -s ”Наблюдаемый узел” -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

Отправить **342.45** в качестве значения для элемента данных **mysql.queries** в **Наблюдаемый узел** на Zabbix сервер с IP-адресом **192.168.1.113** с использованием TLS сертификата.

zabbix_sender -z 192.168.1.113 -s ”Наблюдаемый узел” -k mysql.queries -o 342.45 --tls-connect psk --tls-psk-identity ”PSK ID Zabbix agentd” --tls-psk-file /home/zabbix/zabbix_agentd.psk

Отправить **342.45** в качестве значения для элемента данных **mysql.queries** в **Наблюдаемый узел** на Zabbix сервер с IP-адресом **192.168.1.113** с использованием TLS и общего ключа (pre-shared key, PSK).

СМОТРИТЕ ТАКЖЕ

Документация https://www.zabbix.com/manuals

zabbix_agentd(8), **zabbix_get**(1), **zabbix_proxy**(8), **zabbix_server**(8), **zabbix_js**(1), **zabbix_agent2**(8), **zabbix_web_service**(8)

Оглавление

НАИМЕНОВАНИЕ

СИНТАКСИС

ОПИСАНИЕ

ПАРАМЕТРЫ

СТАТУС ЗАВЕРШЕНИЯ

ПРИМЕРЫ

СМОТРИТЕ ТАКЖЕ

АВТОР

Этот документ был создан: 11 июня 2021 года, 08:42:39 GMT

zabbix_server

Раздел: Команды обслуживания (8)
Обновлено: 2020-09-04
Оглавление Вернуться на главную

НАИМЕНОВАНИЕ

zabbix_server — демон службы Zabbix сервера

СИНТАКСИС

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

ОПИСАНИЕ

zabbix_server — это основной демон программного обеспечения Zabbix.

ПАРАМЕТРЫ

-c, **--config** *config-file* Использовать альтернативный файл конфигурации *config-file* вместо файла по умолчанию.

-f, **--foreground** Запустить сервер Zabbix без перехода в фоновый режим.

-R, **--runtime-control** *runtime-option* Выполнить административные функции в соответствии с *runtime-option*.

-T, **--test-config** Проверить файл конфигурации и выйти.

-h, **--help** Вывести справочную информацию и выйти.

-V, **--version** Отобразить информацию о версии и выйти.

Примеры запуска Zabbix сервера с параметрами командной строки:

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

УПРАВЛЕНИЕ ВО ВРЕМЯ ВЫПОЛНЕНИЯ

Параметры управления во время выполнения:

config_cache_reload
Перезагрузить кэш конфигурации.
Игнорируется, если кэш в данный момент загружается.
Для поиска PID-файла будет использован файл конфигурации по умолчанию (если не указана опция **-c**), и сигнал будет отправлен процессу, указанному в PID-файле.

snmp_cache_reload
Перезагрузить кэш SNMP, очистить свойства SNMP (время движка, загрузки движка, идентификатор движка, учетные данные) для всех узлов сети.
Обратите внимание, что Zabbix кэширует сопоставления EngineID→IP для SNMPv3 и может автоматически повторно использовать EngineID для снижения накладных расходов на опрос.

housekeeper_execute
Выполнить housekeeper.
Игнорируется, если housekeeper в данный момент уже выполняется.

trigger_housekeeper_execute
Выполнить housekeeper триггеров (удалить проблемы для удаленных триггеров).
Игнорируется, если housekeeper триггеров для сервисов в данный момент уже выполняется.

diaginfo[=*section*]
Записать во внутренний журнал диагностическую информацию указанного раздела.
Разделом может быть *historycache*, *preprocessing*, *alerting*, *lld*, *valuecache*, *locks*.
По умолчанию записывается диагностическая информация всех разделов.

ha_status
Записать в журнал состояние кластера высокой доступности (HA).

ha_remove_node[=*target*]
Удалить узел высокой доступности (HA), указанный по имени или ID. Обратите внимание, что активные/резервные узлы удалить нельзя.

ha_set_failover_delay[=*delay*]
Установить задержку переключения при отказе для высокой доступности (HA). Поддерживаются суффиксы времени, например 10s, 1m.

proxy_config_cache_reload[=*target*] Перезагрузить кэш конфигурации прокси.

secrets_reload
Перезагрузить секреты из Vault.

service_cache_reload
Перезагрузить кэш менеджера служб.

prof_enable[=*target*]
Включить профилирование. Если target не указан, влияет на все процессы. Включенное профилирование предоставляет сведения обо всех rwlocks/mutexes по имени функции.

prof_disable[=*target*]
Отключить профилирование. Если target не указан, влияет на все процессы.

log_level_increase[=*target*]
Увеличить уровень журналирования, если target не указан, влияет на все процессы

log_level_decrease[=*target*]
Уменьшить уровень журналирования, если target не указан, влияет на все процессы

Цели управления уровнем журналирования *process-type*
Все процессы указанного типа (alerter, alert manager, 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, icmp pinger, internal poller, ipmi manager, ipmi poller, java poller, odbc poller, poller, agent poller, http agent poller, snmp poller, preprocessing manager, proxy group manager, proxy poller, self-monitoring, service manager, snmp trapper, task manager, timer, trapper, unreachable poller, vmware collector)

process-type,N
Тип процесса и номер (например, poller,3)

pid
Идентификатор процесса, до 65535. Для больших значений укажите цель как ”process-type,N”

Цели управления профилированием

process-type
Все процессы указанного типа (alerter, alert manager, 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, icmp pinger, internal poller, ipmi manager, ipmi poller, java poller, odbc poller, poller, agent poller, http agent poller, snmp poller, preprocessing manager, proxy group manager, proxy poller, self-monitoring, service manager, snmp trapper, task manager, timer, trapper, unreachable poller, vmware collector)

process-type,N
Тип процесса и номер (например, history syncer,1)

pid
Идентификатор процесса, до 65535. Для больших значений укажите цель как ”process-type,N”

scope
Область профилирования (rwlock, mutex, processing) может использоваться с типом процесса (например, history syncer,1,processing)

ФАЙЛЫ

/usr/local/etc/zabbix_server.conf
Расположение по умолчанию файла конфигурации сервера Zabbix (если не было изменено во время компиляции).

СМОТРИТЕ ТАКЖЕ

Документация https://www.zabbix.com/manuals

zabbix_agentd(8), **zabbix_get**(1), **zabbix_proxy**(8), **zabbix_sender**(1), **zabbix_js**(1), **zabbix_agent2**(8)

Содержание

НАЗВАНИЕ

СИНТАКСИС

ОПИСАНИЕ

ПАРАМЕТРЫ

ФАЙЛЫ

СМ. ТАКЖЕ

АВТОР

Этот документ был создан: 16:12:14 GMT, 04 сентября 2020 г.

zabbix_web_service

Раздел: Команды обслуживания (8)
Обновлено: 2019-01-29
Оглавление Вернуться на главную

НАИМЕНОВАНИЕ

zabbix_web_service — веб-сервис Zabbix

СИНТАКСИС

zabbix_web_service [**-c** *config-file*]
zabbix_web_service [**-c** *config-file*] **-T**
zabbix_web_service -h
zabbix_web_service -V

ОПИСАНИЕ

zabbix_web_service — веб-сервис для предоставления доступа к компонентам Zabbix.

ПАРАМЕТРЫ

-c, **--config** *config-file* ·Использовать альтернативный файл конфигурации *config-file* вместо файла по умолчанию.

-T, **--test-config** Проверить файл конфигурации и выйти.

-h, **--help** ·Вывести справочную информацию и выйти.

-V, **--version** ·Вывести информацию о версии и выйти.

ФАЙЛЫ

/usr/local/etc/zabbix_web_service.conf
Расположение по умолчанию файла конфигурации веб-сервиса Zabbix (если не изменено во время компиляции).

СМОТРИТЕ ТАКЖЕ

Документация https://www.zabbix.com/manuals

zabbix_agentd(8), **zabbix_get**(1), **zabbix_proxy**(8), **zabbix_sender**(1), **zabbix_server**(8), **zabbix_js**(1), **zabbix_agent2**(8)

Оглавление

НАИМЕНОВАНИЕ

СИНТАКСИС

ОПИСАНИЕ

ПАРАМЕТРЫ

ФАЙЛЫ

СМОТРИТЕ ТАКЖЕ

АВТОР

Этот документ был создан: 11 июня 2021 года, 12:58:30 GMT

Уведомление об авторских правах

Документация Zabbix НЕ распространяется по лицензии 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, если фактическое содержимое никоим образом не изменяется и не редактируется. Вы не должны публиковать или распространять эту документацию в какой-либо форме или на каком-либо носителе, за исключением случаев, когда вы распространяете документацию способом, аналогичным тому, как это делает Zabbix (то есть в электронном виде для загрузки на веб-сайте Zabbix), либо на USB-носителе или аналогичном носителе, при

условии, однако, что документация распространяется вместе с программным обеспечением на том же носителе. Любое иное использование, включая распространение печатных копий или использование этой документации полностью или частично в другой публикации, требует предварительного письменного согласия уполномоченного представителя Zabbix. Zabbix сохраняет за собой все права на эту документацию, которые прямо не предоставлены выше.