

Zabbix documentation in Russian

Эти страницы содержат официальную документацию по Zabbix.

Используйте боковое меню для просмотра страниц документации.

Чтобы была возможность наблюдения за страницами, войдите с именем пользователя и паролем вашей учетной записи с [форума Zabbix](#).

Декларация переводчиков о качестве перевода

- Перевод русской документации не является официальным (впрочем как и веб-интерфейса) и осуществляется переводчиками сообщества, представленными на [этой](#) странице. Переводчики являются опытными пользователями системы мониторинга Zabbix.
- Эта декларация обновлена 01.07.2013

1. Введение

Пожалуйста, воспользуйтесь боковым меню для доступа к содержимому раздела Введение.

1 Структура руководства

Структура

Содержимое данного Руководства о Zabbix 4.4 разделено на разделы и подразделы для обеспечения удобного доступа к представляющим интерес темам.

При переходе к соответствующим разделам убедитесь, что вы раскрыли все папки раздела, чтобы увидеть все содержимое, находящееся в подразделах и на отдельных страницах.

Перекрестные ссылки между страницами соответствующего содержания используются как можно чаще, чтобы убедиться, что актуальная информация будет не упущена пользователями.

Разделы

[Введение](#) дает общую информацию об этой версии программного обеспечения Zabbix. Чтение этого раздела должно вооружить вас некоторыми весомыми причинах в пользу выбора Zabbix.

[Понятия в Zabbix](#) объясняют, используемую в Zabbix, терминологию и рассказывают о компонентах Zabbix в деталях.

Разделы [Установки](#) и [Быстрого старта](#) должны помочь вам в начале работы с Zabbix. [Готовое решение Zabbix](#) - альтернатива для быстрой возможности попробовать Zabbix, чтобы понять, что это такое и как использовать Zabbix.

Раздел [Настройка](#) - один из самых больших и очень важных разделов в этом руководстве. Этот раздел наполнен важными советами о том как настроить Zabbix для мониторинга вашей среды, от настройки узлов сети для получения необходимых данных, до просмотра данных, настройки оповещений и удаленных команд, которые будут выполняться в случае возникновения проблем.

Раздел [Услуги IT](#) объясняет как использовать Zabbix для высокоуровневого обзора вашей среды мониторинга.

[Веб-мониторинг](#) должен помочь вам в понимании мониторинга доступности веб сайтов.

[Мониторинг виртуальных машин](#) содержит рекомендации по настройке мониторинга среды VMware.

Разделы [Обслуживание](#), [Регулярные выражения](#), [Подтверждение событий](#) и [Экспорт/импорт XML](#) рассказывают каким еще образом можно использовать различные аспекты программного обеспечения Zabbix.

[Обнаружение](#) содержит инструкции по настройке автоматического обнаружения устройств сети, активных агентов, файловых систем, сетевых интерфейсов и т.д.

[Распределенный мониторинг](#) описывает возможности использования Zabbix в больших и более сложных средах.

[Шифрование](#) помогает понять возможности шифрования соединений между Zabbix компонентами.

[Веб-интерфейс](#) содержит информацию, относящуюся к использованию веб-интерфейса Zabbix.

Раздел [API](#) содержит подробности о работе с Zabbix API.

Подробный список технической информации включен в [Приложения](#). Здесь вы также можете найти раздел Часто задаваемых вопросов.

2 Что такое Zabbix

Обзор

Zabbix создан Алексеем Владышевым, и в настоящее время активно разрабатывается и поддерживается компанией Zabbix SIA.

Zabbix - это решение распределенного мониторинга корпоративного класса с открытыми исходными кодами.

Zabbix - это программное обеспечение для мониторинга многочисленных параметров сети, жизнеспособности и целостности серверов. Zabbix использует гибкий механизм оповещений, что позволяет пользователям конфигурировать уведомления основанные на e-mail практически для любого события. Это позволяет быстро реагировать на проблемы с серверами. Zabbix предлагает отличные функции отчетности и визуализации данных основанные на данных истории. Это делает Zabbix идеальным для планирования мощности.

Zabbix поддерживает и поллеры, и трапперы. Все отчеты и статистика Zabbix, так же как и параметры настройки, доступны через Веб-интерфейс. Веб-интерфейс обеспечивает доступ к информации о состоянии вашей сети и жизнеспособности ваших серверов из любого места. Правильно настроенный, Zabbix может сыграть важную роль в мониторинге ИТ инфраструктуры. Это верно и для маленьких организаций с несколькими серверами и для больших организаций со множеством серверов.

Zabbix бесплатен. Zabbix написан и распространяется под лицензией GPL General Public License версии 2. Это означает, что его исходный код свободно распространяется и доступен для неограниченного круга лиц.

[Коммерческая поддержка](#) доступна и осуществляется самой компанией Zabbix.

Узнайте больше о [возможностях Zabbix](#).

Пользователи Zabbix

Множество организаций различного размера по всему миру полагаются на Zabbix, как на основную платформу мониторинга.

3 Возможности Zabbix

Обзор

Zabbix - высоко интегрированное решение мониторинга сети, которое предлагает множество функций в одном пакете.

Сбор данных

- проверки доступности и производительности
- поддержка мониторинга с использованием SNMP (и трапперы, и поллеры), IPMI, JMX, VMware
- пользовательские проверки
- сбор желаемых данных с использованием пользовательских интервалов
- выполняются сервером/прокси и агентами

Гибкие определения порогов

- вы можете задавать очень гибкие пороги проблем, называемые триггерами, ссылаясь на значения из базы данных

Множество настроек оповещений

- отправку оповещений можно настроить, используя расписания эскалаций, получателей, типов оповещений
- оповещения можно сделать информативными и полезными при использовании переменных макросов
- автоматические действия, включающие в себя удаленные команды

Построение графиков в режиме реального времени

- при помощи встроенного функционала построения графиков сразу же доступны графики по наблюдаемым элементам данных

Возможности Веб-мониторинга

- Zabbix может имитировать нажатия мышкой на веб-сайте, проверить функционал и время ответа

Широкие возможности визуализации

- возможность создавать пользовательские графики, что позволяет комбинировать множество элементов данных в одном месте
- карты сети
- пользовательские комплексные экраны и слайд-шоу наподобие внешнего вида ПАНЕЛИ
- отчеты
- высокоуровневое (бизнес) представление наблюдаемых ресурсов

Хранение данных истории

- данные записываются в базу данных
- настраиваемая история
- встроенная процедура очистки истории

Простая настройка

- добавление наблюдаемых устройств узлами сети
- как только узлы сети имеются в базе данных, они готовы к мониторингу
- применение шаблонов к наблюдаемым устройствам

Использование шаблонов

- группировка проверок в шаблоны
- шаблоны могут наследоваться от других шаблонов

Сетевое обнаружение

- автоматическое обнаружение сетевых устройств
- автоматическая регистрация агентов
- обнаружение файловых систем, сетевых устройств и SNMP OID'ов

Быстрый Веб-интерфейс

- Веб-интерфейс, основанный на языке PHP
- доступен из любого места
- удобная навигация
- журнал аудита

Zabbix API

- Zabbix API обеспечивает программируемый интерфейс к Zabbix для массовых манипуляций, для интеграции стороннего программного обеспечения и других целей.

Система прав доступа

- безопасная аутентификация пользователей
- возможность ограничения доступа отдельным пользователям к конкретным страницам

Полнофункциональный и легко расширяемый агент

- разворачивается на наблюдаемых машинах
- можно развернуть как на Linux, так и на Windows

Бинарные демоны

- написаны на C, имеют высокую производительность и используют небольшой объем памяти
- легко переносимы

Готовность с сложным средам

- простой удаленный мониторинг с использованием Zabbix прокси

4 Обзор Zabbix

Архитектура

Zabbix состоит из нескольких основных программных компонентов, функции которых изложены ниже.

Сервер

Zabbix сервер является основным компонентом, которому агенты сообщают информацию и статистику о доступности и целостности. Сервер является главным хранилищем, в котором хранятся все данные конфигурации, статистики, а также оперативные данные.

База данных

Как таковая вся информация о конфигурации, а так же данные собранные Zabbix, хранятся в базе данных.

Веб-интерфейс

Для легкого доступа к Zabbix из любого места и с любой платформы, поставляется интерфейс на основе Веб. Интерфейс является частью Zabbix сервера и обычно (но не обязательно) работает на том же самой физической машине, что и сервер.

Прокси

[Zabbix прокси](#) может собирать данные о производительности и доступности от имени Zabbix сервера. Прокси является optionalной частью Zabbix; однако он может быть полезен чтобы распределить нагрузку одного Zabbix сервера.

Агент

[Zabbix агенты](#) разворачиваются на наблюдаемых системах для активного мониторинга за локальными ресурсами и приложениями, и для отправки собранных данных Zabbix серверу или прокси.

Поток данных

Кроме того, важно сделать шаг назад и взглянуть на весь поток данных в Zabbix. Для того чтобы создать элемент данных, который будет собирать данные, вы должны сначала создать узел сети. Перемещаясь в другой конец спектра Zabbix, у вас должен быть элемент данных, чтобы создать триггер. У вас должен быть триггер, чтобы создать действие. Таким образом, если вы хотите получать оповещения о слишком высокой загрузке CPU на Сервере X, вы сначала должны создать запись о узле сети для Сервера X, затем элемент данных для наблюдения за CPU, затем триггер, который сработает, если загрузка CPU будет слишком высокой, а затем действие которое отправит вам e-mail. Хотя может показаться, что требуется слишком много шагов, использование шаблонов значительно упрощает процесс. Однако, такое построение системы позволяет создавать очень гибкие инсталляции.

2. Определения

Обзор

В этом разделе вы можете ознакомиться со значениями некоторых терминов, которые используются в Zabbix.

Определения

узел сети

- сетевое устройство, мониторинг которого вы хотите производить, имеющее IP/DNS.

группа узлов сети

- логическая группировка узлов сети; она может содержать узлы сети и шаблоны. Узлы сети и шаблоны в группе узлов сети никаким образом не связаны с друг другом. Группы узлов сети используются при назначении прав доступа к узлам сети различным группам пользователей.

элемент данных

- конкретный фрагмент данных, который вы хотите получать от узла сети, метрика

предобработка значений

- трансформация полученного метрикой значения перед сохранением этого значения в базу данных.

триггер

- логическое выражение, которое определяет порог проблемы и используется для "оценки" данных полученных элементами данных.

Если полученные данные превышают порог, триггер переходит из состояния 'Ok' в состояние 'Проблема'. Если полученные данные ниже порога, триггер остается/возвращается в состояние 'Ok'.

событие

- одиночное возникновение того, что заслуживает внимания, например изменение состояния триггера или обнаружение/авторегистрация агента

тег события

- предопределенный маркер для события. Его можно использовать в корреляции событий, в правах доступа и т.д.

корреляция событий

- гибкий и точный метод корреляции проблем с решением этих проблем.

Например, вы можете задать, что проблема, которую сообщил один триггер, может быть решена другим триггером, который может использовать даже совершенно другой метод сбора данных.

проблема

- триггер, который находится в состоянии “Проблема”.

обновление проблемы

- варианты управления проблемами, такими как добавление комментария, подтверждение, изменение важности и закрытие вручную, поставляются Zabbix.

действие

- предопределенные средства реагирования на события

Действия состоят из операций (например, отправка оповещений) и условий (когда осуществляется операция)

эскалация

- пользовательский сценарий для выполнения операций в рамках действия; последовательность отправки оповещений/выполнений удаленных команд.

способ оповещения

- способ доставки оповещений; канал доставки.

оповещение

- сообщение о некотором событии, отправленное пользователю через выбранный канал доставки.

удаленная команда

- предопределенная команда, которая будет автоматически выполнена на наблюдаемом узле сети при некоторых условиях.

шаблон

- набор сущностей (элементы данных, триггеры, графики, комплексные экраны, группы элементов данных, правила низкоуровневого обнаружения, веб-сценарии) готовые к присоединению к одному или нескольким узлам сети.

Задача шаблонов повысить скорость развертывания задач мониторинга узлов сети; кроме того упростить применение массовых изменений к задачам наблюдения. Шаблоны соединяются напрямую с отдельными узлами сети.

группа элементов данных

- группировка элементов данных в некую логическую группу.

веб-сценарий

- один или несколько HTTP запросов для проверки доступности веб-сайта.

веб-интерфейс

- веб-интерфейс поставляемый с Zabbix

ПАНЕЛЬ

- раздел веб-интерфейса, который отображает сводку и визуализацию важной информации в визуальных блоках называемых виджетами.

виджет

- визуальный блок, который отображает информацию определенного вида и источника (сводка, карта, график, часы и т.д.), используется на панели.

Zabbix API

- Zabbix API позволяет вам использовать протокол JSON RPC для создания, обновления и получения объектов Zabbix (например, узлов сети, элементов данных, графиков и прочих) или для выполнения любых других пользовательских задач.

Zabbix сервер

- главный процесс программного обеспечения Zabbix, который выполняет мониторинг, взаимодействует с прокси и агентами Zabbix, вычисляет триггеры, отправляет оповещения; центральное хранилище данных.

Zabbix агент

- процесс разворачиваемый на наблюдаемых целях для активного мониторинга локальных ресурсов и приложений.

Zabbix прокси

- процесс, который может собирать данные от имени Zabbix сервера, перенимая часть нагрузки сервера.

шифрование

- поддержка шифрованной связи между компонентами Zabbix (сервером, прокси, агентом и утилитами zabbix_sender и zabbix_get) с использованием Протокола защиты транспортного уровня (TLS).

обнаружение сети

- автоматическое обнаружение сетевых устройств.

низкоуровневое обнаружение

- автоматическое обнаружение низкоуровневых объектов на конкретном устройстве (например, файловые системы, сетевые интерфейсы и т.д.).

правило низкоуровневого обнаружения

- набор определений для автоматического обнаружения низкоуровневых объектов на устройстве.

прототип элементов данных

- метрика, у которой некоторые параметры представлены в виде переменных, готовых для низкоуровневого обнаружения. После низкоуровневого обнаружения эти переменные автоматически заменяются реальными обнаруженными параметрами и метрика автоматически начинает сбор данных.

прототип триггеров

- триггер, у которого некоторые параметры представлены в виде переменных, готовых для низкоуровневого обнаружения. После низкоуровневого обнаружения эти переменные автоматически заменяются реальными обнаруженными параметрами и триггер автоматически начинает оценку данных.

Прототипы некоторых других Zabbix объектов также используются в низкоуровневом обнаружении - прототипы графиков, прототипы узлов сети, прототипы групп узлов сети, прототипы групп элементов данных.

авторегистрация агента

- автоматизированный процесс при котором Zabbix агент регистрируется самостоятельно узлом сети и начинает наблюдаться.

3. Процессы Zabbix

3.1 Сервер

Обзор

Zabbix сервер - центральный процесс программного обеспечения Zabbix.

Сервер выполняет опрос и отлов данных, вычисляет триггеры, отправляет оповещения пользователям. Он является центральным компонентом, которому Zabbix агенты и прокси сообщают данные о доступности и целостности систем. Сервер может самостоятельно удаленно проверять сетевые службы (такие как веб-сервера и почтовые сервера), используя простые проверки сервисов.

Сервер является главным хранилищем, в котором хранятся все конфигурационные, статистические и оперативные данные, так же он рассыпает уведомления администраторам в случае возникновения проблем с любой из наблюдаемых систем.

Функционал базового Zabbix сервера разделен на три отдельных компонента; это: Zabbix сервер, веб-интерфейс и хранилище в базе данных.

Все данные о конфигурации Zabbix хранятся в базе данных, с которой взаимодействует и сервер и веб-интерфейс. Например, когда вы создаете новый элемент данных используя веб-интерфейс (или API), запись об этом добавляется в таблицу элементов данных в базе данных. Затем, раз в минуту Zabbix сервер опрашивает таблицу элементов данных для получения списка активных элементов данных, и сохраняет этот список в кэш Zabbix сервера. Именно поэтому любые изменения в веб-интерфейсе Zabbix будут отображены в разделе последних данных с задержкой до двух минут.

Процесс сервера

Если установлен из пакета

Zabbix сервер работает как демон. Для запуска сервера выполните:

```
shell> service zabbix-server start
```

Эта команда будет работать на большинстве GNU/Linux системах. На других системах вам, возможно, потребуется выполнить:

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

Аналогично, для остановки/перезапуска/просмотра состояния, используйте следующие команды:

```
shell> service zabbix-server stop  
shell> service zabbix-server restart  
shell> service zabbix-server status
```

Запуск вручную

Если выше приведенные команды не работают, вам необходимо запустить сервер вручную. Найдите путь к бинарному файлу zabbix_server и выполните:

```
shell> zabbix_server
```

Можно использовать следующие параметры командной строки с Zabbix сервером:

-c --config <файл>	путь к файлу конфигурации (по умолчанию /usr/local/etc/zabbix_server.conf)
-R --runtime-control <опция>	выполнение административных функций
-h --help	вывод этого сообщения помощи
-V --version	вывод номера версии

Выполнение административных функций не поддерживается в OpenBSD и NetBSD.

Примеры запуска Zabbix сервера с параметрами командой строки:

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

Управление работой

Опции управления работой:

Опция	Описание	Цель
config_cache_reload	Перезагрузка кэша конфигурации. Игнорируется, если кэш уже загружается в текущий момент времени.	
housekeeper_execute	Запуск процедуры очистки базы данных. Игнорируется, если процедура очистки выполняется в данный момент.	
log_level_increase[=<цель>]	Увеличение уровня журналирования, действует на все процессы, если цель не указана.	pid - Идентификатор процесса (1 до 65535) тип процесса - Все процессы указанного типа (например, poller) тип процесса,N - Тип процесса и номер (например, poller,3)
log_level_decrease[=<цель>]	Уменьшение уровня журналирования, действует на все процессы, если цель не указана.	

Допустимый диапазон PID изменения уровня журналирования одного процесса с 1 до 65535. На системах с большими значениями PID опция <тип процесса,N> может использоваться для изменения уровня журналирования отдельных процессов.

Пример использования административных функций для перезагрузки кэша конфигурации сервера:

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

Пример использования административных функций для вызова выполнения очистки базы данных:

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

Примеры использования административных функций по изменению уровня журналирования:

Увеличение уровня журналирования по всем процессам:

```
shell> zabbix_server -c /usr/local/etc/zabbix_server.conf -R log_level_increase
```

Увеличение уровня журналирования у второго процесса поллера:

```
shell> zabbix_server -c /usr/local/etc/zabbix_server.conf -R log_level_increase=poller,2
```

Увеличение уровня журналирования у процесса с PID 1234:

```
shell> zabbix_server -c /usr/local/etc/zabbix_server.conf -R log_level_increase=1234
```

Уменьшение уровня журналирования по всем http поллер процессам:

```
shell> zabbix_server -c /usr/local/etc/zabbix_server.conf -R log_level_decrease="http poller"
```

Пользователь процесса

Zabbix сервер спроектирован для запуска от непrivилегированного пользователя (non-root). Он будет работать от любого непrivилегированного пользователя от которого был запущен. Таким образом, вы можете запускать сервер от имени любого непprivилегированного пользователя, без каких либо последствий.

Если вы попытаетесь запустить сервер от 'root', сервер сразу переключится на пользователя 'zabbix', который должен присутствовать в вашей системе. Единственный способ запустить сервер от пользователя 'root' - соответствующим образом отредактировать параметр 'AllowRoot' в файле конфигурации сервера.

Если Zabbix сервер и [агент](#) работают на одном сервере, то рекомендуется использовать разных пользователей для запуска сервера и для запуска агента. В противном случае, если сервер и агент запущены под одним пользователем, агент будет иметь доступ к файлу конфигурации сервера и любой пользователь с правами Администратора в Zabbix может с легкостью получить, например, пароль от базы данных.

Файл конфигурации

Смотрите опции [файла конфигурации](#) для получения подробной информации по настройке Zabbix сервера.

Скрипты запуска

Скрипты используются для автоматического запуска/остановки процессов Zabbix при включении/выключении системы. Скрипты находятся в директории misc/init.d.

Поддерживаемые платформы

В связи с требованиями безопасности и критически важного характера работы сервера, UNIX является единственной операционной системой, которая может обеспечить необходимую производительность, отказоустойчивость и гибкость. Zabbix работает с ведущими на рынке версиями операционных систем.

Zabbix сервер протестирован на следующих платформах:

- Linux
- Solaris
- AIX
- HP-UX
- Mac OS X
- FreeBSD
- OpenBSD
- NetBSD
- SCO Open Server
- Tru64/OSF1

Также Zabbix может работать и на других Unix-подобных операционных системах.

Региональные настройки (локаль)

Обратите внимание, что сервер требует UTF-8 локаль, чтобы некоторые текстовые элементы данных интерпретировались корректно. Большинство современных Unix-подобных систем уже имеют локаль UTF-8 по умолчанию, тем не менее, есть некоторые системы где это необходимо указывать вручную.

3.2 Агент

Обзор

Zabbix агенты разворачиваются на наблюдаемых целях для активного мониторинга локальных ресурсов и приложений (статистика жестких дисков, памяти, процессоров и т.д.).

Агент локально собирает оперативную информацию и отправляет данные Zabbix серверу для дальнейшей обработки. В случае проблем (таких как отсутствие свободного места на жестком диске или аварийного завершения процесса сервиса), Zabbix сервер может быстро уведомить администраторов конкретного сервера, сообщившего об ошибке.

Zabbix агенты чрезвычайно эффективны, потому что используют родные системные вызовы для сбора информации статистики.

Пассивные и активные проверки

Zabbix агенты могут выполнять пассивные и активные проверки.

В случае [пассивной проверки](#) агент отвечает на запрос данных. Zabbix сервер (или прокси) запрашивает данные, например, загрузку CPU, и Zabbix агент возвращает результат.

[Активные проверки](#) требуют более сложной обработки. Агент сначала получает список элементов данных для независимой обработки от Zabbix сервера. Далее он будет периодически отправлять новые значения серверу.

Выбор между пассивной и активной проверкой осуществляется выбором соответствующего [типа элемента данных](#). Zabbix агент обрабатывает элементы данных типов 'Zabbix агент' и 'Zabbix агент (активный)'.

Поддерживаемые платформы

Zabbix агент поддерживается на:

- Linux
- IBM AIX
- FreeBSD
- NetBSD
- OpenBSD
- HP-UX
- Mac OS X
- Solaris
- Windows: все версии для настольных компьютеров и серверов начиная с XP

Агент на UNIX системах

Zabbix агент на UNIX-подобных системах запускается на хостах, которые необходимо мониторить.

Установка

Смотрите раздел [установки из пакетов](#) для получения более детальных сведений по установке Zabbix агента из пакетов.

В качестве альтернативы вы можете обратиться к инструкциям по [ручной установке](#), если вы не хотите использовать пакеты.

В целом 32-битный Zabbix агент будет работать на 64-битных системах, но в отдельных случаях возможно возникновение ошибок.

Если установлен пакетом

Zabbix агент работает как демон. Агент можно запустить, выполнив:

```
shell> service zabbix-agent start
```

Эта команда будет работать на большинстве GNU/Linux системах. На других системах вам, возможно, потребуется выполнить:

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

Аналогично, для остановки/перезапуска/просмотра состояния Zabbix агента, используйте следующие команды:

```
shell> service zabbix-agent stop  
shell> service zabbix-agent restart  
shell> service zabbix-agent status
```

Запуск вручную

Если выше приведенные команды не работают, вам необходимо запустить агент вручную. Найдите путь к бинарному файлу zabbix_agentd и выполните:

```
shell> zabbix_agentd
```

Агент на Windows системах

Zabbix агент на Windows работает как служба Windows.

Подготовка

Zabbix агент распространяется в виде zip архива. После загрузки архива вам необходимо распаковать его. Выберите любую папку для записи Zabbix агента и файла конфигурации, например:

```
C:\zabbix
```

Скопируйте bin\win64\zabbix_agentd.exe и conf\zabbix_agentd.win.conf файлы в c:\zabbix.

Измените c:\zabbix\zabbix_agentd.win.conf файл в соответствии с вашими требованиями, убедитесь что указан корректный параметр “Hostname”.

Установка

После этого используйте следующую команду для установки Zabbix агента службой Windows:

```
C:\> c:\zabbix\zabbix_agentd.exe -c c:\zabbix\zabbix_agentd.win.conf -i
```

Теперь у вас должна быть возможность настраивать службу “Zabbix agent” так же как и любую другую службу Windows.

Смотрите [более детальные сведения](#) по установке и запуску Zabbix агента на Windows.

Другие опции агента

Имеется возможность запуска нескольких экземпляров агента на одном узле сети. Одиночный экземпляр может быть настроен на использование файла конфигурации по умолчанию или файл конфигурации указанный в командной строке. В случае нескольких экземпляров каждый агент должен иметь свой собственный файл конфигурации (один из них может использовать файл конфигурации по умолчанию).

Следующие параметры командной строки могут быть использованы с Zabbix агентом:

Параметр	Описание
UNIX и Windows агент	
-с --config <файл-конфигурации>	Путь к файлу конфигурации. Вы можете использовать данную опцию, чтобы задать файл конфигурации, размещенному в папке отличной от заданной по умолчанию.

Параметр	Описание	
UNIX и Windows агент		
-p --print	<p>В UNIX, путь по умолчанию /usr/local/etc/zabbix_agentd.conf или как задано во время компиляции переменными --sysconfdir или --prefix В Windows, путь по умолчанию c:\zabbix_agentd.conf</p> <p>Вывод известных элементов данных и выход.</p> <p><i>Обратите внимание:</i> Также для получения результатов пользовательских параметров, вы можете указать файл конфигурации (если он находится вне папки заданной по умолчанию).</p>	
-t --test <ключ элемента данных>	<p>Тестирование указанного элемента данных и выход.</p> <p><i>Обратите внимание:</i> Также для получения результатов пользовательских параметров, вы можете указать файл конфигурации (если он находится вне папки заданной по умолчанию).</p>	
-h --help	Вывод справочной информации.	
-V --version	Вывод номера версии	
Только для UNIX агента		
-R --runtime-control <опция>	Выполнение административных функций. Смотрите управление работой .	
Только для Windows агента		
-m --multiple-agents	<p>Использование нескольких экземпляров агента (с -i,-d,-s,-x функциями). Для отделения имени экземпляров служб, каждое имя службы будет в значении Hostvalue из указанного файла конфигурации.</p>	
Только для Windows агента (функции)		
-i --install	Установка Zabbix агента службой	
-d --uninstall	Удаление службы Zabbix Windows агента	
-s --start	Запуск службы Zabbix Windows агента	
-x --stop	Остановка службы Zabbix Windows агента	
Специальные примеры использования параметров командной строки:		
<ul style="list-style-type: none"> отображение всех встроенных элементов данных с их значениями тестирование пользовательского параметра с ключом "mysql.ping" заданном в указанном файле конфигурации установка службы "Zabbix агента" в Windows с использованием пути по умолчанию до файла конфигурации c:\zabbix_agentd.conf установка службы "Zabbix Agent [Hostname]" в Windows с использованием файла конфигурации zabbix_agentd.conf, размещенного в той же самой папке что и бинарный файл агента и определение уникального имени службы, используя значение Hostname из файла конфигурации 		
<pre>shell> zabbix_agentd --print shell> zabbix_agentd -t "mysql.ping" -c /etc/zabbix/zabbix_agentd.conf shell> zabbix_agentd.exe -i shell> zabbix_agentd.exe -i -m -c zabbix_agentd.conf</pre>		
Управление работой		
Используя опции административных функций, вы можете изменить уровень журналирования у процессов агента.		
Опция	Описание	Цель
log_level_increase[= <цель>]	Увеличение уровня журналирования. Действует на все процессы, если цель не указана.	Цель можно указать с помощью: pid - идентификатора процесса (от 1 до 65535) тип процесса - все процессы указанного типа (например, poller)
log_level_decrease[= <цель>]	Уменьшение уровня журналирования.	

Опция	Описание	Цель
=<цель>]	журналирования. Действует на все процессы, если цель не указана.	тип процесса, N - тип процесса и номер (например, poller,3)
Обратите внимание, что допустимый диапазон PID изменения уровня журналирования одного процесса с 1 до 65535. На системах с большими значениями PID опция <тип процесса,N> может использоваться для изменения уровня журналирования отдельных процессов.		

Примеры:

- увеличение уровня журналирования по всем процессам
- увеличение уровня журналирования у второго процесса listener
- увеличение уровня журналирования у процесса с PID 1234
- уменьшение уровня журналирования по всем процессам active check

```
shell> zabbix_agentd -R log_level_increase
shell> zabbix_agentd -R log_level_increase=listener,2
shell> zabbix_agentd -R log_level_increase=1234
shell> zabbix_agentd -R log_level_decrease="active checks"
```

Выполнение административных функций не поддерживается в OpenBSD, NetBSD и Windows.

Пользователь процесса

Zabbix агент спроектирован для запуска от не привилегированного пользователя (non-root). Он будет работать от любого не привилегированного пользователя от которого был запущен. Таким образом, вы можете запускать агента от имени любого не привилегированного пользователя, без каких либо последствий.

Если вы попытаетесь запустить агента от 'root', он сразу переключится на пользователя 'zabbix', который должен присутствовать в вашей системе. Единственный способ запустить агента от пользователя 'root' - соответствующим образом отредактировать параметр 'AllowRoot' в файле конфигурации агента.

Файл конфигурации

Для получения подробной информации смотрите параметры файла конфигурации по настройке [zabbix_agentd](#) или [Windows агента](#).

Региональные настройки (локаль)

Обратите внимание что сервер требует UTF-8 локаль чтобы некоторые текстовые элементы данных могли быть интерпретированы корректно. Большинство современных Unix-подобных систем имеют локаль UTF-8 установленной по умолчанию, тем не менее, есть некоторые системы где это необходимо задать вручную.

Код завершения процесса

До версии Zabbix 2.2, Zabbix агент возвращал 0 в случае успешного завершения процесса и код 255 в случае ошибки. Начиная с версии Zabbix 2.2 и выше Zabbix агент возвращает код 0 в случае корректного завершения процесса и код 1 в случае отказа/аварийного завершения процесса.

3.3 Прокси

Обзор

Zabbix прокси - это процесс, способный собирать данные мониторинга с одного или нескольких наблюдаемых устройств и отправлять эту информацию Zabbix серверу, таким образом прокси

работает от имени сервера. Все собранные данные локально буферизуются и затем отправляются Zabbix серверу, которому принадлежит этот прокси.

Развертывание прокси не обязательно, но может быть очень полезно для распределения нагрузки одиночного Zabbix сервера. Если только прокси собирают данные, то обработка этих данных на сервере не так сильно нагружает CPU и I/O диска.

Zabbix прокси - идеальное решение для централизованного мониторинга удаленных объектов, филиалов и сетей, где отсутствуют локальные администраторы.

Для Zabbix прокси требуется отдельная база данных.

Zabbix прокси поддерживает следующие базы данных SQLite, MySQL и PostgreSQL. Вы можете использовать Oracle или IBM DB2 на свой собственный риск, при этом возможны определенные ограничения, например в [значениях](#), возвращаемых правилами низкоуровневого обнаружения.

Смотрите также: [Использование прокси в распределенной среде](#)

Процесс прокси

Если установлен из пакета

Zabbix прокси работает как демон. Для запуска прокси выполните:

```
shell> service zabbix-proxy start
```

Эта команда будет работать на большинстве GNU/Linux системах. На других системах вам, возможно, потребуется выполнить:

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

Аналогично, для остановки/перезапуска/просмотра состояния, используйте следующие команды:

```
shell> service zabbix-proxy stop
shell> service zabbix-proxy restart
shell> service zabbix-proxy status
```

Запуск вручную

Если выше приведенные команды не работают, вам необходимо запустить прокси вручную. Найдите путь к бинарному файлу zabbix_proxy и выполните:

```
shell> zabbix_proxy
```

Следующие параметры командной строки могут быть использованы с Zabbix прокси:

-c --config <файл>	путь к файлу конфигурации (по умолчанию /etc/zabbix/zabbix_proxy.conf)
-R --runtime-control <опция>	выполнение административных функций
-h --help	вывод этого сообщения помощи
-V --version	вывод номера версии

Выполнение административных функций не поддерживается в OpenBSD и NetBSD.

Примеры запуска Zabbix прокси с параметрами командой строки:

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

Управление работой

Опции управления работой:

Опция	Описание	Цель
config_cache_reloa	Перезагрузка кэша конфигурации.	

Опция	Описание	Цель
d	Игнорируется, если кэш уже загружается в текущий момент времени.	
housekeeper_execute	Активный Zabbix прокси подключится к Zabbix серверу и запросит данные конфигурации.	
log_level_increase[=<цель>]	Запуск процедуры очистки базы данных. Игнорируется, если процедура очистки выполняется в данный момент.	Увеличение уровня журналирования, действует на все процессы, если цель не указана.
log_level_decrease[=<цель>]	Уменьшение уровня журналирования, действует на все процессы, если цель не указана.	

Допустимый диапазон PID изменения уровня журналирования одного процесса с 1 до 65535. На системах с большими значениями PID опция <тип процесса,N> может использоваться для изменения уровня журналирования отдельных процессов.

Пример использования административных функций для перезагрузки кэша конфигурации прокси:

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

Пример использования административных функций для вызова выполнения очистки базы данных:

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

Примеры использования административных функций по изменению уровня журналирования:

Увеличение уровня журналирования по всем процессам:

```
shell> zabbix_proxy -c /usr/local/etc/zabbix_proxy.conf -R log_level_increase
```

Увеличение уровня журналирования у второго процесса поллера:

```
shell> zabbix_proxy -c /usr/local/etc/zabbix_proxy.conf -R log_level_increase=poller,2
```

Увеличение уровня журналирования у процесса с PID 1234:

```
shell> zabbix_proxy -c /usr/local/etc/zabbix_proxy.conf -R log_level_increase=1234
```

Уменьшение уровня журналирования по всем http поллер процессам:

```
shell> zabbix_proxy -c /usr/local/etc/zabbix_proxy.conf -R log_level_decrease="http poller"
```

Пользователь процесса

Zabbix прокси спроектирован для запуска от не привилегированного пользователя (non-root). Он будет работать от любого не привилегированного пользователя от которого был запущен. Таким образом, вы можете запускать прокси от имени любого не привилегированного пользователя, без каких-либо последствий.

Если вы попытаетесь запустить прокси от 'root', он сразу переключится на пользователя 'zabbix', который должен присутствовать в вашей системе. Единственный способ запустить прокси от пользователя 'root' - соответствующим образом отредактировать параметр 'AllowRoot' в конфигурационном файле прокси.

Файл конфигурации

Для получения подробной информации смотрите опции [файла конфигурации](#) по настройке zabbix_proxy.

Поддерживаемые платформы

Zabbix прокси запускается на тех же [поддерживаемых платформах](#), что и Zabbix сервер.

Региональные настройки (локаль)

Обратите внимание что сервер требует UTF-8 локаль чтобы некоторые текстовые элементы данных могли быть интерпретированы корректно. Большинство современных Unix-подобных систем имеют локаль UTF-8 установленной по умолчанию, тем не менее, есть некоторые системы где это необходимо задать вручную.

3.4 Java gateway

Обзор

Начиная с Zabbix 2.0 появился новый демон Zabbix, называемый “Zabbix Java gateway”, обеспечивающий нативную поддержку мониторинга JMX приложений. Zabbix Java gateway - это демон написанный на языке Java. Когда Zabbix сервер хочет знать значение конкретного JMX счетчика узла сети, он опрашивает Zabbix Java gateway, который используя [API управления JMX](#) опрашивает интересующее удаленное приложение. Приложению не требуется никакого дополнительного программного обеспечения, оно просто должно быть запущено с опцией командной строки `-Dcom.sun.management.jmxremote`.

Java gateway принимает входящие подключения от Zabbix сервера или прокси и может быть использован только как “пассивный прокси”. Но в отличии от Zabbix прокси, Java gateway может использоваться с Zabbix прокси (тогда как один Zabbix прокси не может работать через другой Zabbix прокси). Доступ к каждому Java gateway настраивается непосредственно в файле конфигурации Zabbix сервера или прокси, таким образом только один Java gateway может быть настроен на Zabbix сервере или Zabbix прокси. Если у узла сети есть элементы данных типа **JMX агент** и элементы данных других типов, то только элементы данных **JMX агент** будут запрошены через Java gateway.

Когда элемент данных должен быть обновлен через Java gateway, Zabbix сервер или прокси подключается к Java gateway и запрашивает значение, Java gateway в свою очередь запрашивает это значение 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 несколькими потоками, настраиваемыми опцией **START_POLLERS**. На стороне сервера, если соединение занимает более чем **Timeout** секунд, оно будет завершено, но Java gateway может оставаться занят получением значения JMX счетчика. Чтобы решить эту проблему, Java gateway начиная с Zabbix 2.0.15, Zabbix 2.2.10 и Zabbix 2.4.5 поддерживают опцию **TIMEOUT**, позволяющую указать время ожидания сетевых операций JMX.

Zabbix сервер и прокси будут пытаться максимально объединить запросы к одной цели JMX (зависит от интервалов обновления элементов данных) и отправлять их в Java gateway за одно подключение для лучшей производительности.

Рекомендуется выставить значение **StartJavaPollers** меньшим или равным чем **START_POLLERS**, в противном случае могут возникнуть ситуации, когда потоков Java gateway может не хватить для обслуживания входящих запросов.

Разделы ниже рассказывают о том как получить и запустить Zabbix Java gateway, как настроить Zabbix сервер (или Zabbix прокси) для использования Zabbix Java gateway в мониторинге JMX, и как настроить элементы данных Zabbix в Zabbix веб-интерфейсе, которые соответствуют конкретным JMX счетчикам.

3.4.2.1 Получение Java gateway

Вы можете установить Java gateway как из исходных кодов, так и из пакетов, которые можно загрузить с [Zabbix веб-сайта](#).

Смотрите инструкции по установке Java gateway:

- [из исходных кодов](#)
- из пакетов для [RHEL/CentOS](#)
- из пакетов для [Debian/Ubuntu](#)

3.4.2.2 Обзор файлов из поставки Java gateway

Независимо от того как вы получили Java gateway, в итоге у вас должен быть набор скриптов, JAR файлом и файлами конфигурации в папке \$PREFIX/sbin/zabbix_java. Назначение этих файлов изложено ниже.

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

Собственно JAR файл Java gateway.

`lib/logback-core-0.9.27.jar`
`lib/logback-classic-0.9.27.jar`
`lib/slf4j-api-1.6.1.jar`
`lib/android-json-4.3_r3.1.jar`

Зависимости Java gateway: [Logback](#), [SLF4J](#), и библиотека [Android JSON](#).

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

Файлы конфигурации для Logback.

`shutdown.sh`
`startup.sh`

Скрипты для удобства запуска и остановки Java gateway.

`settings.sh`

Файл конфигурации, который используется вышеупомянутыми скриптами запуска и остановки.

3.4.2.3 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.

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 отличии от сервера и прокси, Java gateway легок и не требует наличия базы данных.

3.4.2.4 Настройка сервера для использования с Java gateway

Когда Java gateway запущен и работает, вы должны указать Zabbix серверу где искать 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. Вы можете это сделать таким же способом как и изменение количества поллеров и трапперов.

```
StartJavaPollers=5
```

Не забудьте перезапустить сервер или прокси после того как закончите изменение настроек.

3.4.2.5 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 "debug":

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

Обратите внимание, что в отличии от Zabbix сервера или Zabbix прокси, вам не нужно перезапускать Zabbix Java gateway после изменения файла logback.xml - изменения в logback.xml будут применены автоматически. Когда вы завершите отладку, вы можете вернуть уровень журналирования обратно в "info".

Если вы хотите записывать журнал в другой файл или в совершенно другую среду такую как база данных, настройте файл logback.xml в соответствии с вашими потребностями. Обратитесь к [Руководству по Logback](#) для получения более подробных сведений.

Иногда для 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 SLF4J для журналирования, вы можете заменить Logback выбранным вами фреймворком, поместим соответствующий JAR файл в каталог lib. Обратитесь к [Руководство по SLF4J](#) для получения более подробных сведений.

3.5 Sender

Обзор

Zabbix sender - это утилита командной строки, которая может быть использована для отправки данных о производительности Zabbix серверу для последующей их обработки.

Обычно эта утилита используется в долго выполняемых пользовательских скриптах для периодической отправки данных о доступности и производительности.

Чтобы отправлять результаты напрямую на Zabbix сервер или прокси, необходимо задать тип [элемента данных траппер](#).

Отправка одного значения

Пример отправки одного значения на Zabbix сервер используя Zabbix sender:

```
shell> cd bin  
shell> ./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 (указанное в веб-интерфейсе Zabbix)
- k - ключ элемента данных
- о - отправляемое значение

Если объекты содержат пробелы, то эти объекты должны быть заключены в двойные кавычки.

Обратитесь к [странице помощи Zabbix sender](#) для получения более подробной информации.

Zabbix sender в Windows можно выполнить тем же самым способом:

```
zabbix_sender.exe [опции]
```

Начиная с версии Zabbix 1.8.4 утилита Zabbix_sender была улучшена в плане отправки данных в реальном времени. Имеется ввиду, что большое количество значений, полученное за короткий промежуток времени, будут накапливаться во временном стеке и затем будет отправлено серверу за одно соединение. Данные, пришедшие через менее, чем 0,2 сек после предыдущего значения накапливаются в одном стеке, но максимальное время их накопления до отправки все же 1 секунда.

Zabbix sender завершится, если в файле конфигурации имеется ошибочный (не соответствующий определению *параметр=значение*) параметр.

3.6 Get

Обзор

Zabbix get - это утилита командной строки, которая подключается к Zabbix агенту и получает от него запрашиваемую информацию.

Утилита обычно используется для диагностики Zabbix агентов.

Выполнение Zabbix get

Пример выполнения Zabbix get в UNIX для получения значения загрузки процессора от агента:

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

Еще один пример выполнения утилиты Zabbix get для получения строки с веб сайта:

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

Обратите внимание, что элемент данных содержит пробелы поэтому используются двойные кавычки для выделения ключа элемента данных. Двойные кавычки не являются частью ключа элемента данных; они будут обрезаны командной строкой и не будут переданы Zabbix агенту.

Zabbix get принимает следующие параметры командной строки:

<code>-s --host <имя хоста или IP></code>	Требуется указать имя хоста или IP адрес хоста
<code>-p --port <номер порта></code>	Требуется указать номер порта агента, запущенного
на узле сети. По умолчанию 10050.	
<code>-I --source-address <IP адрес></code> подключения	Требуется указать IP адрес для исходящего

-k -key <ключ элемента данных>
которого мы хотим получить.
-h --help
-V --version

Требуется указать ключ элемента данных, значение
Вывод этого сообщения помощи
Вывод номера версии

Смотрите также [страницу помощи по Zabbix get](#) для получения более подробной информации.

Zabbix get в Windows можно выполнить тем же самым способом:

`zabbix_get.exe [опции]`

4. Установка

4.1 Получение Zabbix

Обзор

Имеется четыре способа получения Zabbix:

- Установка из [пакетов](#)
- Загрузка самого нового архива с исходными кодами и [самостоятельная их сборка](#)
- Установка из [контейнеров](#)
- Загрузка [решения виртуализации](#)

Для загрузки самых новых исходных кодов или решения виртуализации, перейдите на [страницу загрузки Zabbix](#), где указаны прямые ссылки на последние версии. Ссылки на более старые версии находятся ниже ссылок на стабильные версии.

4.2 Требования

Аппаратное обеспечение

Память

Zabbix требуется и оперативная память, и физическая память на жестком диске. Отправной точкой могут быть 128 МБ оперативной памяти и 256 МБ свободного места на жестком диске. Впрочем, очевидно, что объем необходимой дисковой памяти зависит от количества наблюдаемых узлов сети и наблюдаемых параметров. Если вы планируете достаточно долго хранить историю наблюдаемых параметров, то потребуется по крайней мере несколько гигабайт для хранения данных истории в базе данных. Каждый процесс демона Zabbix требует несколько подключений к базе данных. Объем памяти требуемый каждым подключением к базе данных зависит от настроек базы данных.

Чем больше оперативной памяти у вас имеется, тем быстрее работает база данных (а следовательно, и Zabbix)!

CPU

Zabbix и особенно база данных может потребовать значительных процессорных ресурсов в зависимости от количества наблюдаемых параметров и выбранной базы данных.

Другое оборудование

Для использования SMS уведомлений встроенных в Zabbix потребуется последовательный порт передачи данных и GSM модем. Конвертер USB-to-serial также будет работать.

Примеры конфигураций оборудования

В таблице приводятся несколько вариантов аппаратных конфигураций:

Название	Платформа	CPU/Память	База данных	Наблюдаемые узлы сети
Маленькая	CentOS	Виртуальная машина	MySQL InnoDB	20
Средняя	CentOS	2 ядра CPU / 2ГБ	MySQL InnoDB	500
Большая	RedHat Enterprise Linux	4 ядра CPU / 8ГБ	RAID10 MySQL InnoDB или PostgreSQL	>1000
Очень большая	RedHat Enterprise Linux	8 ядер CPU / 16ГБ	Быстрый RAID10 MySQL InnoDB или PostgreSQL	>10000
Фактически параметры конфигурации зависят от количества активных элементов данных и частоты обновления этих элементов. Настоятельно рекомендуем производить запуск базы данных на отдельном сервере для крупных установок.				

Поддерживаемые платформы

В связи с требованиями безопасности и критически важным характером работы системы мониторинга, единственной операционной системой, которая может обеспечить необходимую производительность, отказоустойчивость и гибкость является операционная система UNIX. Zabbix работает на всех ведущих версиях ОС.

Zabbix протестирован на следующих платформах:

- Linux
- IBM AIX
- FreeBSD
- NetBSD
- OpenBSD
- HP-UX
- Mac OS X
- Solaris
- Windows: все версии для настольных компьютеров и серверов начиная с XP (только Zabbix агент)

Также Zabbix может работать и на других Unix-подобных операционных системах.

Zabbix отключает дампы памяти, если скомпилирован с шифрованием и не запустится, если система не позволяет отключение дампов памяти.

Программное обеспечение

Zabbix построен на современном веб-сервере Apache, ведущих СУБД, и языке сценариев PHP.

Системы управления базами данных

Программа	Версия	Комментарии
MySQL	5.0.3 - 8.0.x	Требуется, если MySQL используется как основная база данных Zabbix'a. Требуется InnoDB engine. MariaDB также работает с Zabbix.
Oracle	10g или более новая	Требуется, если Oracle используется как основная база данных Zabbix'a.
PostgreSQL	8.1 или более новая	Требуется, если PostgreSQL используется как основная база данных Zabbix'a. Предлагаем использовать PostgreSQL по крайней мере версии 8.3, который показывает очень хорошую производительность VACUUM .
IBM DB2	9.7 или более новая	Требуется, если IBM DB2 используется как основная база данных Zabbix'a.
SQLite	3.3.5 или более новая	Поддерживается только на стороне Zabbix прокси. Требуется, если SQLite используется базой данных Zabbix прокси.

Поддержка IBM DB2 является экспериментальной!

Веб-интерфейс

Следующее программное обеспечение потребуется для работы веб-интерфейса Zabbix:

Приложение	Версия	Комментарии
<i>Apache</i>	1.3.12 или более поздняя	
<i>PHP</i>	5.4.0 или более поздняя	
Расширения PHP:		
<i>gd</i>	2.0 или более поздняя	Расширение PHP GD должен поддерживать формат PNG (<i>--with-png-dir</i>), JPEG (<i>--with-jpeg-dir</i>) изображения и FreeType 2 (<i>--with-freetype-dir</i>).
<i>bcmath</i>		<i>php-bcmath</i> (<i>--enable-bcmath</i>)
<i>ctype</i>		<i>php-ctype</i> (<i>--enable-ctype</i>)
<i>libXML</i>	2.6.15 или более поздняя	<i>php-xml</i> или <i>php5-dom</i> , если поставляется как отдельный пакет от поставщика.
<i>xmlreader</i>		<i>php-xmlreader</i> , если поставляется как отдельный пакет от поставщика.
<i>xmlwriter</i>		<i>php-xmlwriter</i> , если поставляется как отдельный пакет от поставщика.
<i>session</i>		<i>php-session</i> , если поставляется как отдельный пакет от поставщика.
<i>sockets</i>		<i>php-net-socket</i> (<i>--enable-sockets</i>). Требуется для поддержки пользовательских скриптов.
<i>mbstring</i>		<i>php-mbstring</i> (<i>--enable-mbstring</i>)
<i>gettext</i>		<i>php-gettext</i> (<i>--with-gettext</i>). Требуется для работы переводов.
<i>ldap</i>		<i>php-ldap</i> . Требуется только, если в веб-интерфейсе используется LDAP аутентификация.
<i>ibm_db2</i>		Требуется, если используется IBM DB2 в качестве базы для Zabbix.
<i>mysqli</i>		Требуется, если используется MySQL в качестве базы для Zabbix.
<i>oci8</i>		Требуется, если используется Oracle в качестве базы для Zabbix.
<i>pgsql</i>		Требуется, если используется PostgreSQL в качестве базы для Zabbix.
Также Zabbix может работать и с предыдущими версиями Apache, MySQL, Oracle, и PostgreSQL. Для шрифтов, кроме DejaVu, который установлен по умолчанию, нужна функция PHP imagerotate . Если функция не установлена, то эти шрифты могут неправильно отображаться при отображении графика. Эта функция доступна только если PHP скомпилирован вместе с GD, которого нет для Debian и для некоторых других дистрибутивов.		

Веб-браузер на стороне клиента

Cookies и Java Script должны быть включены.

Поддерживаются последние версии Google Chrome, Mozilla Firefox, Microsoft Internet Explorer и Opera. Также и другие браузеры (Apple Safari, Konqueror) могут работать с Zabbix.

Реализована одна и та же политика источника для IFRAMES, что означает, что Zabbix веб-интерфейс нельзя поместить во фреймы на другом домене.

Всё еще, страницы помещенные в Zabbix фреймы будут иметь доступ к веб-интерфейсу Zabbix (посредством JavaScript), если страница, которая помещена во фрейм и веб-интерфейс Zabbix располагаются на одном домене. Такая страница как <http://secure-zabbix.com/cms/page.html>, если помещена в комплексные экраны на <http://secure-zabbix.com/zabbix/>, будет иметь полный доступ к Zabbix посредством JS.

Сервер

Обязательные требования нужны всегда. Тогда как опциональные требования требуется только для поддержки определенных функций.

Требование	Статус	Описание
<i>libpcre</i>		PCRE библиотека требуется для поддержки PCRE совместимых регулярных выражений .
<i>libevent</i>	Обязательно	Наименование может отличаться в зависимости от GNU/Linux дистрибутива, например 'libpcre3' или 'libpcre1'. Обратите внимание, что необходима именно PCRE(v8.X), тогда как PCRE2(v10.X) библиотека не используется.
<i>curl</i>	Опционально	Требуется для массового сбора метрик и IPMI мониторинга. Версия 1.4 или более новая.
<i>OpenIPMI</i>		Обратите внимание, что для Zabbix прокси это требование опциональное, и нужно только для IPMI мониторинга.
<i>libssh2</i>		Требуется для поддержки SSH. Версия 1.0 или более новая.
<i>fping</i>		Требуется для элементов данных ICMP пинг .
<i>curl</i>	Опционально	Требуется для веб-мониторинга, мониторинга VMware и для SMTP аутентификации. Для SMTP аутентификации, требуется версия 7.20.0 или выше. Также необходима для поддержки Elasticsearch.
<i>libiksemel</i>		Требуется для поддержки Jabber.
<i>libxml2</i>		Требуется для мониторинга VMware.
<i>net-snmp</i>		Требуется для поддержки SNMP.

Java gateway

Если вы получили Zabbix из репозитория исходных кодов или скачали архив, то необходимые зависимости уже включены в дерево исходного кода.

Если вы получили Zabbix как пакет вашего дистрибутива, то необходимые зависимости обеспечиваются системой управления пакетами.

В обоих вышеупомянутых случаях, программное обеспечение готово к использованию и скачивать какие-либо дополнительные файлы не нужно.

Однако, если вы хотите использовать другие версии этих зависимостей (например, если вы готовите пакет для определенного дистрибутива Linux), ниже приведен список версий библиотек, для которых подтверждена работоспособность Java gateway. Zabbix может также работать с другими версиями этих библиотек.

Следующая таблица содержит список JAR файлов, которые поставляются вместе в Java gateway в оригинальном коде:

Библиотека	Лицензия	Веб сайт	Комментарии
<i>logback-core-0.9.27.jar</i>	EPL 1.0, LGPL 2.1	http://logback.qos.ch/	Протестировано с версиями 0.9.27, 1.0.13, и 1.1.1.
<i>logback-classic-0.9.27.jar</i>	EPL 1.0, LGPL 2.1	http://logback.qos.ch/	Протестировано с версиями 0.9.27, 1.0.13, и 1.1.1.
<i>slf4j-api-1.6.1.jar</i>	MIT License	http://www.slf4j.org/	Протестировано с версиями 1.6.1, 1.6.6, и 1.7.6.
<i>android-json-4.3_r3.1.jar</i>	Apache License 2.0	https://android.googlesource.com/platform/libcore/+master/json	Протестировано с версиями 2.3.3_r1.1 и 4.3_r3.1. Обратитесь к файлу src/zabbix_java/lib/README для

получения инструкций по созданию JAR файла.

Java gateway компилируется и запускается при наличии Java версии 1.6 или выше. Если вы готовите прекомпилированную версию Zabbix gateway для использования ее другими, то рекомендуется использовать для компиляции Java 1.6, в этом случае Zabbix gateway будет работать на всех версиях Java вплоть до самой последней.

Размер базы данных

Данные конфигурации Zabbix требуют фиксированное количество дискового пространства и сильно не увеличиваются.

Размер базы данных Zabbix в основном зависит от следующих переменных, которые определяют объем хранимых данных истории:

- Количество обрабатываемых запросов в секунду

Это среднее количество новых значений, которые Zabbix сервер получает каждую секунду. Например: Если имеется 3000 элементов данных с интервалом проверки 60 секунд, то количество обрабатываемых запросов за секунду рассчитывается $3000/60 = 50$.

Это означает, что каждую секунду в базу данных Zabbix добавляется 50 новых записей.

- Настройки очистки истории в базе данных

Zabbix хранит значения определенный период времени, обычно несколько недель или месяцев. Каждое новое значение требует определенный объем дискового пространства для данных и индексов.

Таким образом, если требуется сохранение 30 дней истории и каждую секунду в базу данных добавляется 50 новых записей, общее количество значений будет равно примерно $(30*24*3600)* 50 = 129.600.000$ или около 130М значений.

В зависимости от типа базы данных, типа полученных значений (с плавающей точкой, целочисленный, строки, файлы журналов и т.д.) может потребоваться от 40 байт до сотен байт дискового пространства для хранения значения. Обычно одно значение занимает около 90 байт по числовым элементам данных². В нашем случае это означает, что 130М значений потребуют $130M * 90 байт = 10.9ГБ$ дискового пространства.

Размер значений текстовых/журнальных элементов данных невозможно предугадать, но вы можете ожидать около 500 байт на значение.

- Настройки очистки динамики изменений в базе данных

Zabbix хранит ежечасную статистику значений max/min/avg/count для каждого элемента данных в таблице **trends**. Эти данные используются для отслеживания динамики изменений и для графиков при отображении большого периода времени. Период в 1 час не является настраиваемым.

Базе данных Zabbix, в зависимости от типа базы данных, требуется около 90 байт на один элемент. Предположим, что если требуется хранить динамику изменений в течении 5 лет. Значения 3000 элементов данных потребуют $3000*24*365* 90 = 2.2ГБ$ за год, или **11ГБ** за 5 лет.

- Настройки очистки событий в базе данных

Каждое событие требует около 250 байт дискового пространства¹. Сложно точно оценить количество событий, ежедневно генерируемых Zabbix сервером. В самом худшем случае, мы можем предположить, что Zabbix генерирует одно событие в секунду.

По каждому событию восстановления создается запись в event_recovery. Обычно большая часть событий восстанавливается, поэтому мы можем предположить, что в event_recovery будет по одной записи по каждому событию. Это означает дополнительные 80 байт по каждому событию.

Опционально, у событий могут быть теги, каждая запись тега требует приблизительно 100 байт дискового пространства¹. Количество тегов по каждому событию (#теги) зависит от конфигурации. Таким образом каждая запись потребует дополнительно #теги * 100 байт дискового пространства².

Это означает, что для того, чтобы хранить события 3 года, потребуется $3*365*24*3600*(250 + 80 + \#теги * 100) = \sim 30\text{ГБ} + \#теги * 100\text{Б}$ дискового пространства.

¹ Больше, когда имеются не-ASCII имена событий, тегов и значения.

² Приблизительные размеры основаны на MySQL и могут отличаться для других баз данных.

Представленная ниже таблица содержит формулы для расчета требуемого пространства жесткого диска для системы мониторинга Zabbix:

Параметр	Формула для расчета занимаемого места(в байтах)
Конфигурация Zabbix	Фиксированный размер. Ориентировочно 10МБ или меньше. дней*(элементов/частота обновления)*24*3600*байт элементы : количество элементов данных дней : количество дней хранения истории частота обновления : среднее значение периода проверки элементов данных байт : количество байт, требуемых для одного значения, зависит от типа базы данных, около 90 байт дней*(элементов/3600)*24*3600*байт элементов : количество элементов данных дней : количество дней хранения динамики изменений байт : количество байт, требуемых для одного значения, зависит от типа базы данных, около 90 байт. дней*событий*24*3600*байт событий : количество событий в секунду. Одно (1) событие в худшем случае. дней : количество дней хранения событий байт : количество байт, требуемых для одного значения, зависит от типа базы данных, обычно примерно 330 + среднее количество тегов по каждому событию * 100 байт.
История	
Динамика изменений	
События	

Общее количество требуемого места на жестком диске рассчитывается:

Конфигурация + История + Динамика изменений + События

После установки Zabbix такое дисковое пространство более НЕ будет использовано сразу. Размер базы данных будет постепенно увеличиваться и остановится по достижении определенного момента, зависящего от настроек очистки базы данных.

Необходимое свободное место на жестком диске, при использовании распределенного мониторинга, рассчитывается аналогичным образом, но оно так же будет зависеть от количества подчиненных нод, связанных с нодой рассчитываемого узла.

Синхронизация времени

Очень важно иметь точную дату и время системы на сервере с запущенным Zabbix. [ntpd](#) один из наиболее популярных демонов синхронизации времени хоста с временем на остальных серверах. Настоятельно рекомендуется поддерживать синхронизированное время на всех системах, где работают Zabbix компоненты.

Если время не синхронизировано, Zabbix будет конвертировать штампы времени собранных данных на время Zabbix сервера путем получения штампов времени клиента/сервера после установки

соединения для передачи данных и корректировки штампов времени у полученных значений элементов данных при помощи разницы между клиент-сервер временем. Чтобы не усложнять работу и избежать возможные сложности, задержка в соединении игнорируется. По этой причине задержка в соединении добавляется к штампам времени полученных данных с активных подключений (активный агент, активный прокси, sender) и вычитается из штампов времени полученных данных с пассивных подключений (пассивный прокси). Все остальные проверки выполняются по времени сервера и их штампы времени не корректируются.

Наилучшие практики для безопасной установки Zabbix

Обзор

В этом разделе содержатся рекомендации, которые следует соблюдать для того, чтобы настроить Zabbix безопасным образом.

Практики, описанные здесь, не требуются для работы Zabbix. Они рекомендуются для повышения безопасности системы.

Защищённый пользователь для Zabbix агента

В конфигурации по умолчанию процессы Zabbix сервера и Zabbix агента делят одного 'zabbix' пользователя. Если вы убедитесь, что агент не сможет получить доступ к конфиденциальной информации из конфигурации сервера (например, данные подключения в базе данных), агента необходимо запускать из под другого пользователя:

1. Создайте защищённого пользователя
2. Укажите этого пользователя в [файле конфигурации](#) агента (параметр 'User')
3. Перезапустите агента с правами администратора. Привилегии администратора сбрасываются на указанного пользователя.

Настройка SSL для Zabbix веб-интерфейса

На RHEL/Centos, установите пакет mod_ssl:

```
yum install mod_ssl
```

Создайте папку 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 -out /etc/httpd/ssl/apache-selfsigned.crt
```

Заполните подсказки соответствующим образом. Самая важная строка здесь, которая запрашивает Common Name. Вам необходимо указать имя домена, которое вы хотите связать с вашим сервером. Вместо него вы можете указать публичный IP адрес, если у вас отсутствует имя домена. В этой статье мы будем использовать *example.com*.

```
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 []:
```

Измените конфигурацию 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 сервис Apache, чтобы применить изменения:

```
systemctl restart httpd.service
```

Включение Zabbix в корневом каталоге URL

Добавьте виртуальный хост в конфигурацию Apache и задайте постоянную переадресацию для корневого канала на Zabbix SSL URL. Не забудьте заменить *example.com* на актуальное имя сервера.

```
/etc/httpd/conf/httpd.conf
```

#Добавьте строки

```
<VirtualHost *:*>  
    ServerName example.com  
    Redirect permanent / http://example.com  
</VirtualHost>
```

Перезапустите сервис Apache, чтобы применить изменения:

```
systemctl restart httpd.service
```

Отключение отображения информации о веб-сервере

Рекомендуется отключить все подписи веб-сервера, как часть процесса по улучшению защищенности веб-сервера. По умолчанию веб-сервер раскрывает подпись программного обеспечения:

```
▼ Response Headers      view source  
Cache-Control: no-store, no-cache, must-revalidate  
Connection: Keep-Alive  
Content-Encoding: gzip  
Content-Length: 1160  
Content-Type: text/html; charset=UTF-8  
Keep-Alive: timeout=5, max=100  
Pragma: no-cache  
Server: Apache/2.4.18 (Ubuntu)
```

Эту подпись можно отключить, добавив две строки в файл конфигурации Apache (используется как пример):

```
ServerSignature Off  
ServerTokens Prod
```

Подпись PHP (Заголовок X-Powered-By HTTP) можно отключить, изменив файл конфигурации php.ini (подпись отключена по умолчанию):

```
expose_php = Off
```

Чтобы изменения файлов конфигурации вступили в силу, необходимо перезапустить веб-сервер.

Дополнительный уровень безопасности можно достичь, используя mod_security (пакет libapache2-mod-security2) с Apache. mod_security позволяет совсем удалить подпись сервера вместо удаления лишь версии из подписи сервера. Подпись можно изменить на любое значение, исправив “SecServerSignature” на любое желаемое значение, после установки mod_security.

Пожалуйста, обратитесь к документации по вашему веб-серверу для того, чтобы узнать как удалять/изменять подписи к программному обеспечению.

Отключение страниц ошибок по умолчанию веб-сервера

Рекомендуется отключить страницы ошибок по умолчанию, чтобы избежать раскрытия информации. По умолчанию веб-сервер использует встроенные страницы ошибок:

Not Found

The requested URL /custom-text was not found on this server.

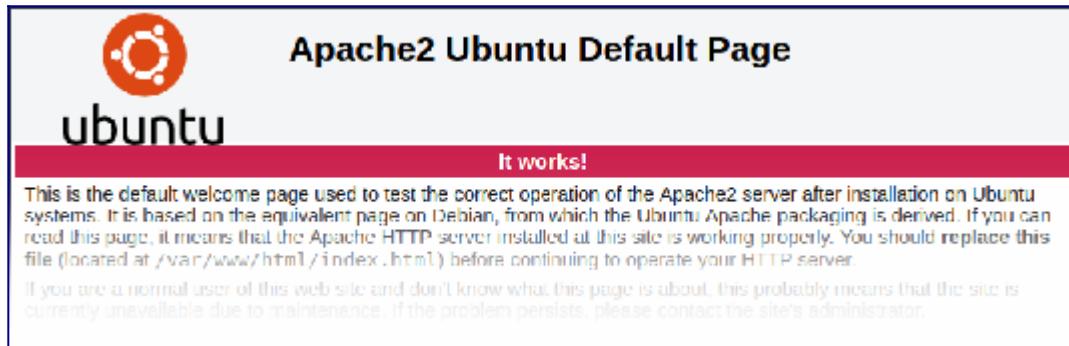
Apache/2.4.18 (Ubuntu) Server at localhost Port 80

Страницы ошибок по умолчанию необходимо заменить/удалить, как часть процесса по улучшению защищенности веб-сервера. Можно использовать директиву “ErrorDocument”, чтобы задать пользовательскую страницу/текст веб-серверу Apache (используется как пример).

Пожалуйста, обратитесь к документации по вашему веб-серверу для того, чтобы узнать как заменять/удалять страницы ошибок по умолчанию.

Удаление тестовой страницы с веб-сервера

Рекомендуется удалить тестовую страницу веб-сервера, чтобы избежать раскрытия информации. По умолчанию, корневой каталог веб-сервера содержит тестовую страницу с именем index.html (Apache2 на Ubuntu используется как пример):



Тестовую страницу необходимо удалить или сделать недоступной, как часть процесса по улучшению защищенности веб-сервера.

4.3 Установка из исходных кодов

Вы можете получить самую свежую версию Zabbix, скомпилировав её из исходных кодов.

Здесь представлено пошаговое руководство по установке Zabbix из исходных кодов.

4.3.1 Установка демонов Zabbix

4.3.1.1 Загрузите архив исходных кодов

Перейдите на [страницу загрузки Zabbix](#) и скачайте архив исходных кодов. Как только он будет загружен, распакуйте исходные коды, выполнив:

```
$ tar -zvxf zabbix-4.0.0.tar.gz
```

Укажите корректную версию Zabbix в команде. Она должна совпадать с именем загруженного архива.

4.3.1.2 Создайте аккаунт пользователя

Все процессы демонов Zabbix требуют непrivилегированного пользователя. Если демон Zabbix запущен от аккаунта пользователя без привилегий, то он будет работать под этим пользователем.

Однако, если демон запущен из под аккаунта 'root', демон переключится на аккаунт пользователя 'zabbix', который должен существовать. Для создания такого аккаунта пользователя (принадлежащего к своей группе, "zabbix") на системах на основе RedHat, выполните:

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

На системах на основе Debian, выполните:

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

Zabbix процессам не требуется домашняя директория, поэтому мы не рекомендуем создавать её. Однако, если вы используете некоторый функционал, которому требуется наличие домашней директории (например, хранение учетных данных mysql в \$HOME/.my.cnf), вы вправе создать домашнюю директорию, используя следующие команды. На системах на основе RedHat, выполните:

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

На системах на основе Debian, выполните:

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

Для установки веб-интерфейса Zabbix отдельного аккаунта пользователя не требуется.

Если Zabbix [сервер](#) и [агент](#) работают на одной машине, то рекомендуется использовать разных пользователей для запуска сервера и для запуска агента. В противном случае, если сервер и агент запущены под одним пользователем, агент будет иметь доступ к файлу конфигурации сервера и любой пользователь с правами Администратора в Zabbix может с легкостью получить, например, пароль от базы данных.

Запуск Zabbix из под root, bin или из под любых других аккаунтов со специальными правами является риском для безопасности.

4.3.1.3 Создайте базу данных Zabbix

Для демонов Zabbix [сервера](#) и [прокси](#), а так же для веб-интерфейса Zabbix, требуется база данных. Она не требуется для запуска Zabbix [агента](#).

Для создания схемы базы данных и вставки набора данных предоставляются SQL [скрипты](#). Для базы данных Zabbix proxy требуется только создание схемы данных, в то время как для Zabbix сервера помимо схемы необходимо также установить набор данных.

После создания базы данных Zabbix, перейдите к следующим шагам компиляции Zabbix.

4.3.1.4 Сконфигурируйте исходный код

При конфигурировании исходного кода Zabbix сервера или прокси, вы должны указать используемый тип базы данных. Только один тип базы данных может быть скомпилирован для процессов сервера или прокси единовременно.

Для просмотра всех доступных опций конфигурирования, выполните в папке с извлеченным исходным кодом Zabbix:

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

Для конфигурирования исходных кодов для Zabbix сервера и агента, вы можете выполнить нечто вроде:

```
./configure --enable-server --enable-agent --with-mysql --enable-ipv6 --with-net-snmp  
--with-libcurl --with-libxml2
```

Опция конфигурации --with-libcurl с cURL 7.20.0 или выше требуется для SMTP аутентификации, поддерживается начиная с Zabbix 3.0.0.

Опции конфигурации --with-libxml2 и --with-libxml2 требуются для мониторинга виртуальных машин, поддерживается начиная с Zabbix 2.2.0.

Начиная с версии Zabbix 3.4, он всегда должен быть скопирован с PCRE библиотекой. Её установка не опциональна. --with-libpcre=[DIR] позволяет указать расположение базовой директории установки, вместо поиска в нескольких стандартных местах расположения libpcre файлов.

Для конфигурирования исходных кодов для Zabbix сервера (с PostgreSQL и т.д.), вы можете выполнить:

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

Для конфигурирования исходных кодов для Zabbix прокси (с SQLite и т.д.), вы можете выполнить:

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

Для конфигурирования исходных кодов для Zabbix агента, вы можете выполнить:

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

Имеется возможность использования флага --enable-static для статической линковки библиотек. Если вы планируете распространять скомпилированные исполняемые файлы на другие сервера, вы должны использовать этот флаг, чтобы приложения могли работать без требуемых библиотек. Флаг --enable-static [не работает под OC Solaris](#).

Использование опции --enable-static не рекомендуется при сборке сервера.

В случае сборки сервера со статической линковкой у вас должны иметься статические версии всех необходимых внешних библиотек.

Утилиты командной строки zabbix_get и zabbix_sender компилируются, если использована опция --enable-agent.

Добавьте опциональный путь к файлу конфигурации

MySQL--with-mysql=/<путь_к_файлу>/mysql_config, чтобы выбрать желаемую библиотеку клиента MySQL, когда имеется необходимость использовать библиотеку, которая не расположена в месте установки по умолчанию.

Бывает полезно там, где установлено несколько версий MySQL или MariaDB установлена вместе с MySQL на одной системе.

Используйте флаг --with-ibm-db2 чтобы указать расположение CLI API.

Используйте флаг --with-oracle чтобы указать расположение OCI API.

Для поддержки шифрования смотрите [Компиляция Zabbix с поддержкой шифрования](#).

4.3.1.5 Соберите и установите всё

Если устанавливаете код взятый из SVN, вы должны сначала выполнить:

```
$ make dbschema
```

```
make install
```

Этот шаг должен быть выполнен пользователем с достаточными правами (как правило 'root', или с помощью sudo).

Выполнение `make install` установит исполняемые файлы демонов (`zabbix_server`, `zabbix_agentd`, `zabbix_proxy`) в `/usr/local/sbin` и исполняемые файлы клиентов (`zabbix_get`, `zabbix_sender`) в `/usr/local/bin`.

Для установки в другое место, отличное от `/usr/local`, используйте ключ `--prefix` в предыдущем шаге конфигурирования исходных кодов, например `--prefix=/home/zabbix`. В этом случае исполняемые файлы демонов будут установлены в папку `<предфикс>/sbin`, а утилиты в папку `<предфикс>/bin`. Страницы помощи будут установлены в папку `<предфикс>/share`.

4.3.1.6 Просмотрите и отредактируйте файлы конфигурации

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

Вам нужно сконфигурировать этот файл для каждого хоста на котором установлен `zabbix_agentd`.

В файле вы должны указать **IP адрес** Zabbix сервера. Подключения с остальных хостов будут отклонены.

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

Вы должны указать имя базы данных, пользователя и пароль (если он используется).

Остальные параметры подойдут со значениями по умолчанию, если у вас небольшая установка (до десятка наблюдаемых узлов сети). Вы должны изменить параметры по умолчанию, если вы хотите увеличить производительность Zabbix сервера (или прокси). Обратитесь к разделу [Оптимизация производительности](#) для получения подробных сведений.

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

Вы должны указать IP адрес сервера и имя прокси (должно быть известно серверу), а также имя базы данных, пользователя и пароль (если он используется).

В случае в SQLite должен быть указан путь к файлу базы данных; пользователь БД и пароль не требуются.

4.3.1.7 Запустите демоны

Запустите `zabbix_server` на стороне сервера.

```
shell> zabbix_server
```

Удостоверьтесь, что в вашей системе разрешено выделение 36МБ (или немногим 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 случится в FreeBSD, Solaris 8.

Обратитесь к разделу "[Смотрите также](#)" в нижней части этой страницы, чтобы узнать как настроить разделяемую память.

Запустите `zabbix_agentd` на всех наблюдаемых машинах.

```
shell> zabbix_agentd
```

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

Если вы установили Zabbix прокси, запустите `zabbix_proxy`.

```
shell> zabbix_proxy
```

Установка веб-интерфейса Zabbix

Копирование файлов PHP

Веб-интерфейс Zabbix написан на языке PHP, поэтому чтобы его запустить вам потребуется веб-сервер с поддержкой PHP. Установка производится путем простого копирования PHP файлов в папку HTML веб-сервера.

Общепринятые места размещения папки HTML документов для веб-сервера Apache включают:

- /usr/local/apache2/htdocs (каталог по умолчанию при установке Apache из исходных кодов)
- /srv/www/htdocs (OpenSUSE, SLES)
- /var/www/html (Debian, Ubuntu, Fedora, RHEL, CentOS)

Рекомендуется использовать подпапку вместо корневой папки HTML. Чтобы создать подпапку и скопировать файлы веб-интерфейса Zabbix, выполните следующие команды, заменив <htdocs> на корректный путь для вашего случая:

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

Если устанавливаете из SVN и планируете использовать любой другой язык, кроме английского, то вы должны сгенерировать файлы переводов. Для этого выполните:

```
locale/make_mo.sh
```

Требуется утилита `msgfmt` из пакета `gettext`.

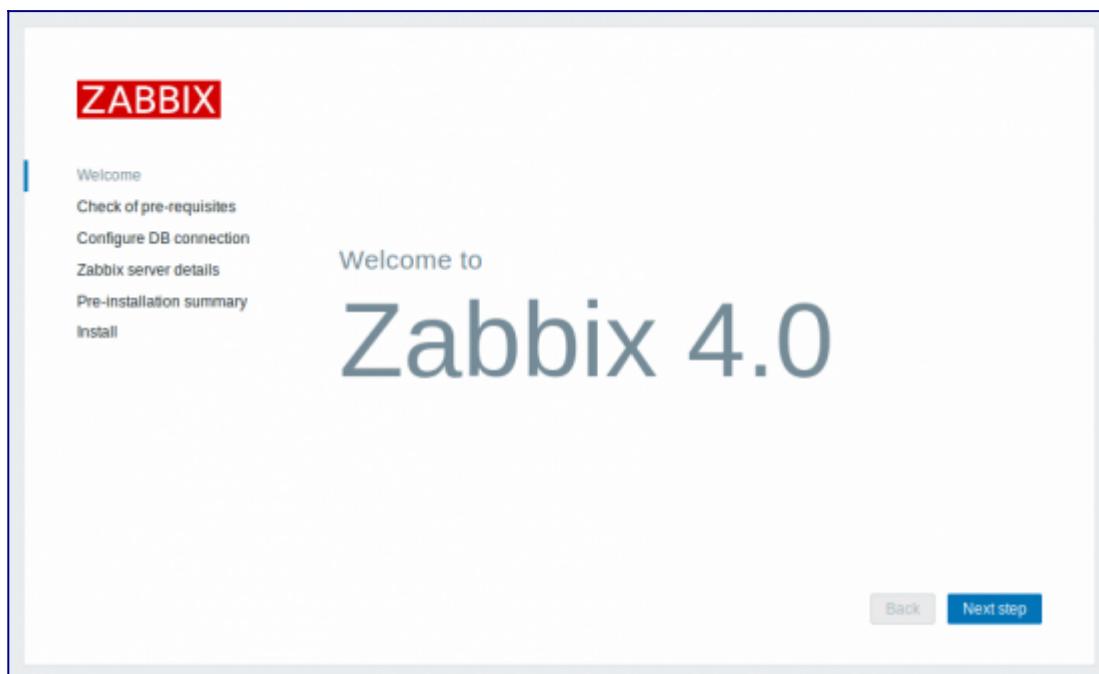
Кроме того, для использования любого другого языка отличного от английского, этот язык должен быть установлен на веб-сервере. Обратитесь к разделу "[Смотрите также](#)" страницы "Профиль пользователя", чтобы узнать как установить дополнительные языки, если они потребуются.

Установка веб-интерфейса

Шаг 1

Откройте URL Zabbix: `http://<ip_или_имя_сервера>/zabbix`, в вашем браузере.

Вы должны увидеть первую страницу помощника установки веб-интерфейса.



Шаг 2

Убедитесь, что все требования к программному обеспечению выполнены.

ZABBIX

Check of pre-requisites

	CURRENT VALUE	REQUIRED
PHP version	5.4.20	5.4.0 OK
PHP option memory_limit	128M	128M OK
PHP option post_max_size	32M	16M OK
PHP option upload_max_filesize	16M	2M OK
PHP option max_execution_time	600	300 OK
PHP option max_input_time	600	300 OK
PHP time zone	Europe/Riga	OK
PHP databases support	MySQL	OK
PHP bcmath	on	OK

[Back](#) [Next step](#)

Требование	Минимальное значение	Описание
<i>Версия PHP</i>	5.4.0	
<i>PHP опция memory_limit</i>	128МБ	В php.ini: memory_limit = 128M
<i>PHP опция post_max_size</i>	16МБ	В php.ini: post_max_size = 16M
<i>PHP опция upload_max_filesize</i>	2МБ	В php.ini: upload_max_filesize = 2M
<i>PHP опция max_execution_time</i>	300 секунд (значения 0 и -1 разрешены)	В php.ini: max_execution_time = 300
<i>PHP опция max_input_time</i>	300 секунд (значения 0 и -1 разрешены)	В php.ini: max_input_time = 300
<i>PHP опция session.auto_start</i>	должна быть отключена	В php.ini: session.auto_start = 0.
<i>Поддержка баз данных</i>	Одна из: MySQL, Oracle, PostgreSQL, IBM DB2	Один из следующих модулей должен быть установлен: mysql, oci8, pgsql, ibm_db2
<i>bcmath</i>		php-bcmath
<i>mbstring</i>		php-mbstring
<i>PHP опция mbstring.func_overload</i>	должна быть отключена	В php.ini: mbstring.func_overload = 0.
<i>PHP опция always_populate_raw_post_data</i>	должна быть равной -1	Требуется только для версий PHP 5.6.0 и новее. В php.ini: always_populate_raw_post_data = -1 php-net-socket. Требуется для поддержки пользовательских скриптов.
<i>sockets</i>		php-gd. Расширение PHP GD должно поддерживать PNG изображения (--with-png-dir), JPEG (--with-jpeg-dir) изображения и FreeType 2 (--with-freetype-dir).
<i>gd</i>	2.0 или выше	php-gd или php5-dom
<i>libxml</i>	2.6.15	php-xmlwriter
<i>xmlwriter</i>		php-xmlreader
<i>xmlreader</i>		php-ctype
<i>ctype</i>		

Требование	Минимальное значение	Описание
session	php-session	
gettext	php-gettext Начиная с Zabbix 2.2.1, расширение PHP gettext более не является обязательным для установки Zabbix. Если gettext не установлен, веб-интерфейс будет работать как обычно, однако, переводы будут недоступны.	

В списке могут также присутствовать необязательные требования. Если необязательное требование не удовлетворено, то оно отображается оранжевым цветом и имеет состояние Предупреждение (Warning). Установка может продолжаться, если необязательные требования не удовлетворены.

Если имеется необходимость изменить пользователя или группу пользователей Apache, необходимо проверить права доступа к папке сессий. В противном случае установка Zabbix не сможет продолжиться.

Шаг 3

Укажите информацию для подключения к базе данных. База данных Zabbix должна быть уже создана.

ZABBIX

Configure DB connection

Please create database manually, and set the configuration parameters for connection to this database. Press "Next step" button when done.

Welcome		
Check of pre-requisites		
Configure DB connection		
Zabbix server details		
Pre-Installation summary		
Install		
Database type	MySQL	
Database host	localhost	
Database port	0	0 - use default port
Database name	zabbix	
User	zabbix	
Password	*****	

Back Next step

Шаг 4

Введите подробные сведения о сервере Zabbix.

ZABBIX

Zabbix server details

Welcome

Check of pre-requisites

Configure DB connection

Zabbix server details

Pre-installation summary

Install

Please enter the host name or host IP address and port number of the Zabbix server, as well as the name of the installation (optional).

Host

Port

Name

[Back](#)

[Next step](#)

Шаг 5

Просмотрите результат настроек.

ZABBIX

Pre-installation summary

Welcome

Check of pre-requisites

Configure DB connection

Zabbix server details

Pre-installation summary

Install

Please check configuration parameters. If all is correct, press "Next step" button, or "Back" button to change configuration parameters.

Database type **MySQL**

Database server **localhost**

Database port **default**

Database name **zabbix**

Database user **zabbix**

Database password *********

Zabbix server **localhost**

Zabbix server port **10051**

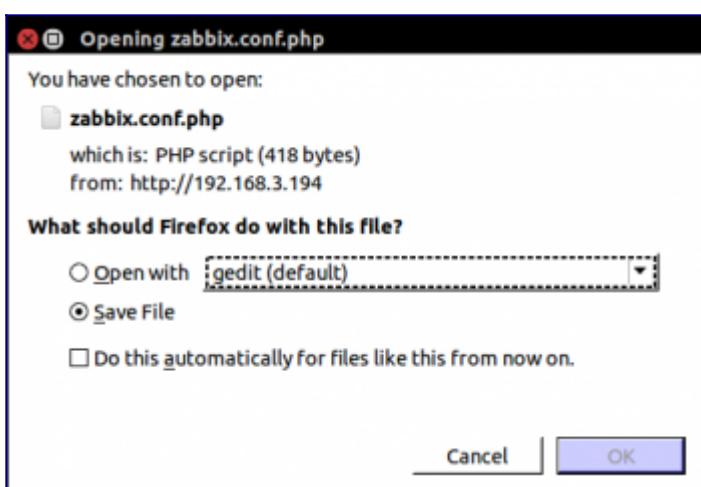
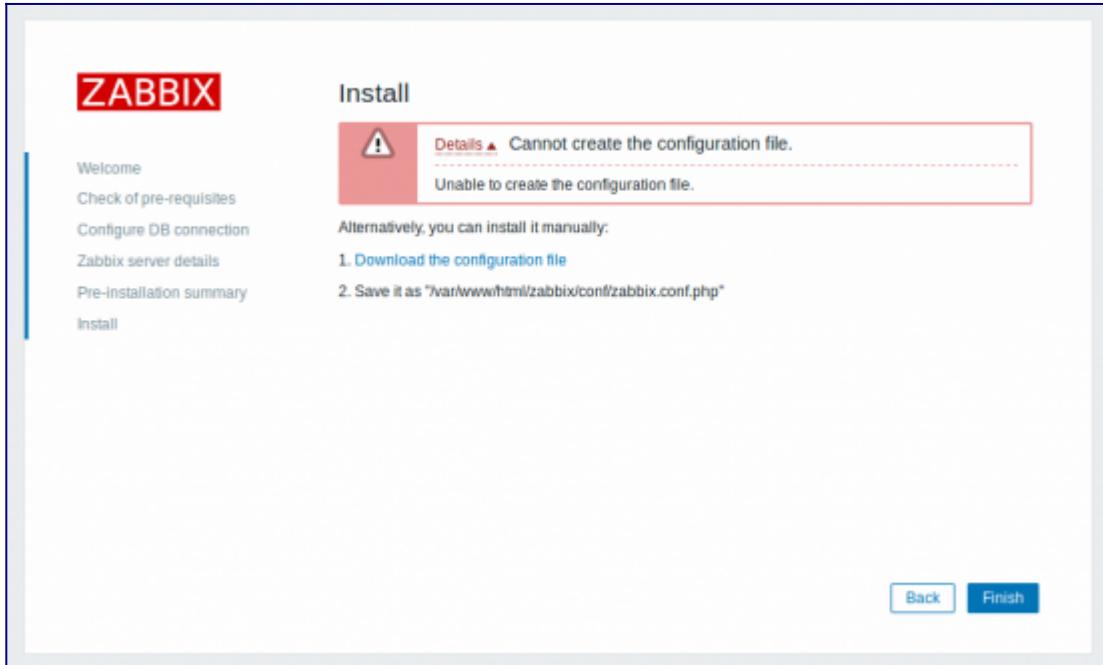
Zabbix server name

[Back](#)

[Next step](#)

Шаг 6

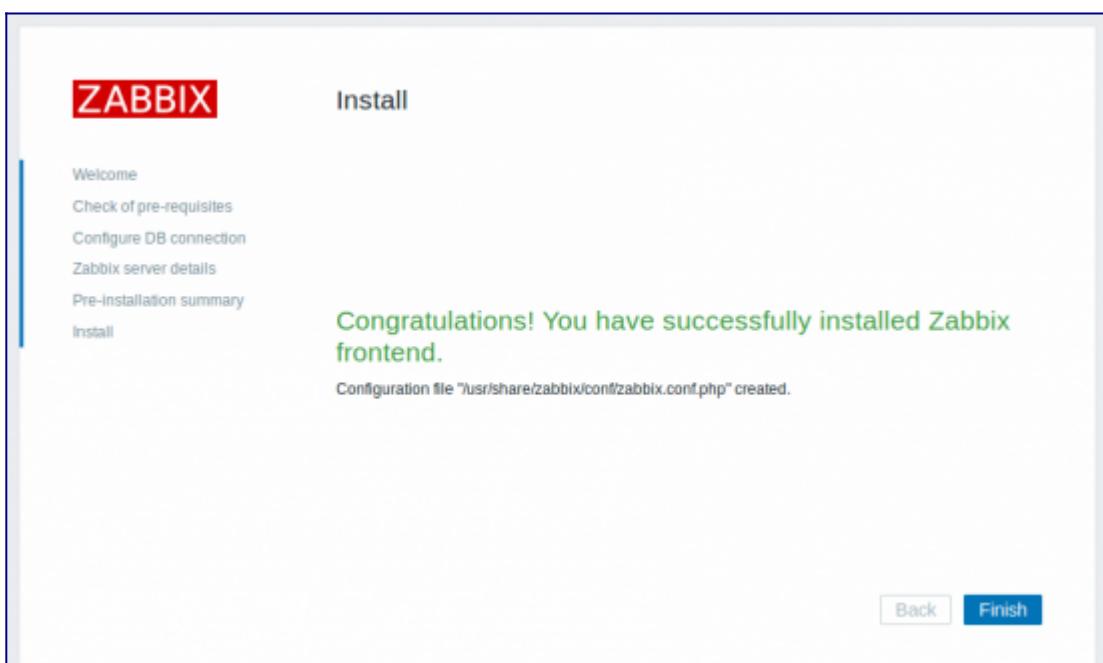
Скачайте файл конфигурации и поместите его в каталог conf/.



В случае, если веб-сервер имеет право на запись в каталог conf/, файл конфигурации будет сохранен автоматически и можно будет сразу же перейти к следующему шагу.

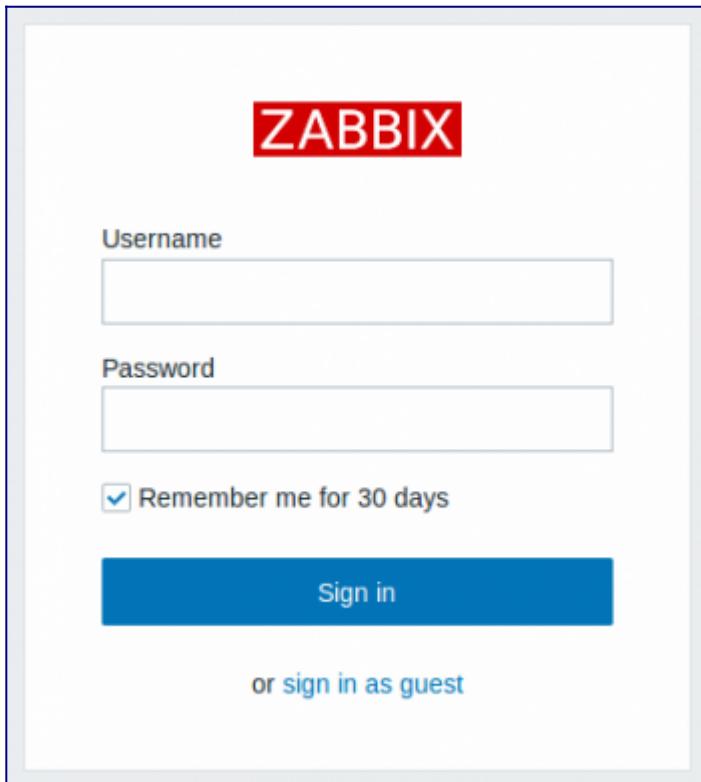
Шаг 7

Завершение установки.



Шаг 8

Веб-интерфейс Zabbix готов! По умолчанию имя пользователя **Admin**, пароль **zabbix**.



Перейти к [началу работы с Zabbix](#).

3 Установка Java gateway

Необходимо установить Java gateway, если вы хотите мониторить JMX приложения. Java gateway легковесный и не требует наличия базы данных.

Чтобы установить его из исходных кодов, сначала [загрузите](#) и извлеките исходные коды из архива.

Чтобы скомпилировать Java gateway, выполните `./configure` скрипт с `--enable-java` опцией. Целесообразно указать опцию `--prefix` для изменения пути установки с `/usr/local` по умолчанию, так как при установке Java gateway будет создано целое дерево директорий, не просто один исполняемый файл.

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

Для компиляции и сборки пакета Java gateway в JAR файл выполните `make`. Обратите внимание, для выполнения этого шага вам потребуется наличие выполняемых файлов `javac` и `jar` в известных системе местах.

```
$ make
```

Теперь у вас имеется `zabbix-java-gateway-$VERSION.jar` файл в `src/zabbix_java/bin`. Если вам удобно запускать Java gateway из `src/zabbix_java` в директории с исходными кодами, тогда вам нужно перейти к инструкциям по настройке и запуску [Java gateway](#). В противном случае убедитесь, что у вас имеются достаточные привилегии и выполните `make install`.

```
$ make install
```

Перейдите на [страницу Java gateway](#) для получения более детальных сведений о том как запустить Java gateway и как настроить Zabbix сервер на работу с ним.

Смотрите также

1. [Как настроить разделенную память для демонов Zabbix \[en\]](#)

4.4 Установка из пакетов

Из пакетов дистрибутивов

Некоторые популярные дистрибутивы ОС предоставляют пакеты Zabbix. Вы можете использовать эти пакеты для установки Zabbix.

В репозиториях ОС может не быть последней версии Zabbix.

Из официального репозитория Zabbix

Zabbix SIA поставляет официальные RPM и DEB пакеты для:

- [Red Hat Enterprise Linux/CentOS](#)
- [Debian/Ubuntu](#)

Файлы пакетов доступны по адресу repo.zabbix.com. Репозитории yum и apt также доступны на сервере. Пошаговое руководство по установке Zabbix из пакетов представлено в подстраницах этого раздела.

4.4.1 Red Hat Enterprise Linux/CentOS

Обзор

Официальные пакеты Zabbix доступны для RHEL 7, Oracle Linux 7 и CentOS 7. В этой документации мы будем ссылаться на все 3, используя термин RHEL.

Пакеты агента и прокси доступны также и для [RHEL 6](#) и [RHEL 5](#).

Добавление Zabbix репозитория

Установите пакет конфигурации репозитория. Этот пакет содержит файлы конфигурации yum (менеджера пакетов приложений).

RHEL 8:

```
# rpm -Uvh https://repo.zabbix.com/zabbix/4.4/rhel/8/x86_64/zabbix-release-4.4-1.el8.noarch.rpm
```

RHEL 7:

```
# rpm -Uvh https://repo.zabbix.com/zabbix/4.4/rhel/7/x86_64/zabbix-release-4.4-1.el7.noarch.rpm
```

RHEL 6:

```
# rpm -Uvh https://repo.zabbix.com/zabbix/4.4/rhel/6/x86_64/zabbix-release-4.4-1.el6.noarch.rpm
```

RHEL 5:

```
# rpm -Uvh https://repo.zabbix.com/zabbix/4.4/rhel/5/x86_64/zabbix-release-4.4-1.el5.noarch.rpm
```

Предварительные условия установки веб-интерфейса

Веб-интерфейс Zabbix требует дополнительные пакеты, которые отсутствуют в базовой установке. Вам необходимо активировать репозиторий опциональных rpm пакетов в системе, где вы запускаете веб-интерфейс Zabbix:

RHEL 7:

```
# yum-config-manager --enable rhel-7-server-optinal-rpms
```

Установка сервера/прокси/веб-интерфейса

Для установки сервера (пакеты доступны для RHEL 7, [устарело для RHEL 6](#)) с MySQL:

```
# yum install zabbix-server-mysql zabbix-web-mysql
```

Для установки Zabbix прокси с MySQL:

```
# yum install zabbix-proxy-mysql
```

Для установки веб-интерфейса Zabbix (пакеты доступны для RHEL 7, [устарело для RHEL 6](#)) с MySQL:

```
# yum install zabbix-web-mysql
```

Замените 'mysql' на 'pgsql' в командах для использования PostgreSQL, или на 'sqlite' для использования SQLite3 (только прокси).

Создание базы данных

Для Zabbix [сервера](#) и [прокси](#) демонов требуется база данных. Она не требуется для работы Zabbix [агента](#).

Если Zabbix сервер и прокси установлены на один хост, их базы данных необходимо создавать с разными именами!

Создайте базу данных, используя имеющиеся [скрипты по созданию базы данных](#) для [MySQL](#) или [PostgreSQL](#).

Импорт данных

Теперь импортируйте изначальную схему и данные сервера на MySQL:

```
# zcat /usr/share/doc/zabbix-server-mysql*/create.sql.gz | mysql -uzabbix -p zabbix
```

Вам будет предложено ввести пароль от недавно созданной базы данных.

На PostgreSQL:

```
# zcat /usr/share/doc/zabbix-server-pgsql*/create.sql.gz | sudo -u <имя_пользователя> psql zabbix
```

Для прокси импортируйте изначальную схему:

```
# zcat /usr/share/doc/zabbix-proxy-mysql*/schema.sql.gz | mysql -uzabbix -p zabbix
```

Для прокси на PostgreSQL (или SQLite):

```
# zcat /usr/share/doc/zabbix-proxy-pgsql*/schema.sql.gz | sudo -u <имя_пользователя> psql zabbix
# zcat /usr/share/doc/zabbix-proxy-sqlite3*/schema.sql.gz | sqlite3 zabbix.db
```

Настройка базы данных для Zabbix сервера/прокси

Измените zabbix_server.conf или zabbix_proxy.conf на использование созданной базы данных.

Например:

```
# vi /etc/zabbix/zabbix_server.conf
DBHost=localhost
DBName=zabbix
DBUser=zabbix
DBPassword=<пароль>
```

В параметре DBPassword используйте пароль от MySQL базы данных Zabbix; пароль пользователя PostgreSQL для PostgreSQL.

Используйте DBHost= для PostgreSQL. Вы возможно, захотите оставить настройку по умолчанию DBHost=localhost (или IP адрес), но в этом случае PostgreSQL будет использовать сетевой сокет при подключении к Zabbix. Смотрите раздел **Настройка SELinux** ниже для получения более подробных сведений.

Запуск процесса Zabbix сервера

Самое время запустить процесс Zabbix сервера:

```
# systemctl start zabbix-server
```

и добавить его в автозагрузку:

RHEL 7 и более новые версии:

```
# systemctl enable zabbix-server
```

RHEL до версии 7:

```
# chkconfig --level 12345 zabbix-server on
```

Замените 'zabbix-server' на 'zabbix-proxy', если вы установили Zabbix прокси.

Настройка Zabbix веб-интерфейса

Для RHEL 7 и поздних версий файл конфигурации Apache для Zabbix веб-интерфейса располагается в /etc/httpd/conf.d/zabbix.conf.

Если вы используете RHEL 6, пожалуйста, ознакомьтесь с разделом об [использовании Zabbix веб-интерфейса на RHEL 6](#), как настроить веб-интерфейс.

Некоторые настройки PHP уже выполнены. Однако, необходимо раскомментировать "date.timezone" настройку и [указать корректный для вас часовой пояс](#).

```
php_value max_execution_time 300
php_value memory_limit 128M
php_value post_max_size 16M
php_value upload_max_filesize 2M
php_value max_input_time 300
php_value max_input_vars 10000
php_value always_populate_raw_post_data -1
# php_value date.timezone Europe/Riga
```

Теперь вы готовы приступить к выполнению [шагов по установке веб-интерфейса](#), которые позволят вам получить доступ к недавно установленному Zabbix.

Официальный репозиторий Zabbix также поставляет пакеты fping, iksemel, libssh2. Эти пакеты располагаются в [non-supported](#) папке.

Настройка SELinux

Если состояние SELinux в принудительном режиме, вам необходимо выполнить следующие команды, чтобы разрешить соединения между Zabbix веб-интерфейсом и сервером:

RHEL 7 и более новые версии:

```
# setsebool -P httpd_can_connect_zabbix on
```

Если база данных доступна через сеть (включая 'localhost' в случае PostgreSQL) вам также потребуется разрешить соединение между Zabbix веб-интерфейсом и базой данных:

```
# setsebool -P httpd_can_network_connect_db on
```

RHEL до версии 7:

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

После завершения настройки веб-интерфейса и SELinux вам необходимо перезапустить веб-сервер Apache:

```
# service httpd restart
```

Веб-интерфейс и сервер Zabbix на RHEL 6

Веб-интерфейс Zabbix на RHEL 6 не поддерживается из-за версии PHP. Начиная с версии Zabbix 3.0, требуется PHP версии 5.4.0 и выше, тогда как в RHEL 6 самая последняя версия 5.3.3.

В большинстве случаев Zabbix сервер и веб-интерфейс устанавливается на одной машине. При обновлении с 2.2 на 3.0 Zabbix сервер выполняет обновление базы данных и веб-интерфейс перестает работать. В этом случае нет возможности откатить изменения в базе данных поэтому пользователи должны обновить PHP используя сторонние пакеты. Это причина по которой Zabbix сервер также считается устаревшим на RHEL 6.

Если вам всё же необходимо использовать Zabbix веб-интерфейс на RHEL 6 и вы уже обновили версию PHP используя сторонние пакеты, тогда вам необходимо сначала активировать репозиторий `zabbix-deprecated`:

- откройте файл `/etc/yum.repos.d/zabbix.repo`
- найдите раздел `[zabbix-deprecated]`
- измените `enabled=1`
- сохраните файл

Вам придётся сделать еще немного ручной настройки. Потому что мы не можем определить версию Apache необходимую для вашего PHP, что делает невозможным для нас предоставить необходимую конфигурацию Apache для веб-интерфейса Zabbix. Мы поставляем 2 конфигурационных файла Apache в пакете Zabbix веб-интерфейса один для Apache 2.2, другой для 2.4, какой вам необходимо использовать с Apache выбирайте сами:

- `httpd22-example.conf`
- `httpd24-example.conf`

Для получения полного пути к файлам выполните:

```
$ rpm -ql zabbix-web | grep example.conf
```

Установка агента

Чтобы установить агента, выполните

```
# yum install zabbix-agent
```

Чтобы запустить агента, выполните:

```
# service zabbix-agent start
```

Установка Java gateway

Необходимо установить [Java gateway](#), если вы хотите мониторить JMX приложения. Java gateway легковесный и не требует наличия базы данных.

Как только требуемый [репозиторий](#) будет добавлен, вы сможете установить Zabbix Java gateway выполнив команду:

```
# yum install zabbix-java-gateway
```

Параметры конфигурации Zabbix Java gateway можно настроить в следующем файле:

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

Для получения более подробных сведенийсмотрите [параметры](#) конфигурации Zabbix Java gateway.

Чтобы запустить Zabbix Java gateway:

```
# service zabbix-java-gateway restart
```

Обратите внимание, что Zabbix сервер [нужно настроить](#) для работы с Java gateway.

Чтобы Zabbix Java gateway автоматически запускался при загрузке системы:

RHEL 7 и более новые версии:

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

RHEL до версии 7:

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

Файл журнала Zabbix Java gateway:

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

Если вы хотите увеличить уровень журналирования, измените 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>
```

4.4.2 Debian/Ubuntu

Обзор

Официальные пакеты Zabbix доступны для:

- Debian 9 (Stretch)
- Debian 8 (Jessie)
- Ubuntu 18.04 (Bionic Beaver) LTS
- Ubuntu 16.04 (Xenial Xerus) LTS
- Ubuntu 14.04 (Trusty Tahr) LTS

Добавление Zabbix репозитория

Установите пакет конфигурации репозитория. Этот пакет содержит файлы конфигурации apt (менеджера пакетов приложений).

Для Debian 9, выполните следующие команды:

Обратите внимание! Для Debian 8, в командах замените 'stretch' на 'jessie'.

```
# wget https://repo.zabbix.com/zabbix/4.2/debian/pool/main/z/zabbix-release/zabbix-
release_4.2-1+stretch_all.deb
# dpkg -i zabbix-release_4.2-1+stretch_all.deb
# apt update
```

Для Ubuntu **Ubuntu 18.04 (bionic)**, выполните следующие команды:

```
# wget https://repo.zabbix.com/zabbix/4.2/ubuntu/pool/main/z/zabbix-release/zabbix-release_4.2-1+bionic_all.deb  
# dpkg -i zabbix-release_4.2-1+bionic_all.deb  
# apt update
```

- Для Ubuntu 16.04 LTS измените в командах 'bionic' на 'xenial'.
- Для Ubuntu 14.04 LTS измените в командах 'bionic' на 'trusty'.

Установка сервера/прокси/веб-интерфейса

Для установки Zabbix сервера с поддержкой MySQL:

```
# apt install zabbix-server-mysql
```

Для установки Zabbix прокси с поддержкой MySQL:

```
# apt install zabbix-proxy-mysql
```

Для установки Zabbix веб-интерфейса:

```
# apt install zabbix-frontend-php
```

Замените 'mysql' в командах на 'pgsql' при использовании PostgreSQL, или на 'sqlite' при использовании SQLite3 (только прокси).

Создание базы данных

Для Zabbix [сервера](#) и [прокси](#) демонов требуется база данных. Она не требуется для работы Zabbix [агента](#).

Если Zabbix сервер и прокси установлены на один хост, их базы данных необходимо создавать с разными именами!

Создайте базу данных, используя инструкции для [MySQL](#) или [PostgreSQL](#).

Импорт данных

Теперь импортируйте изначальную схему и данные сервера на MySQL:

```
# zcat /usr/share/doc/zabbix-server-mysql/create.sql.gz | mysql -uzabbix -p zabbix
```

Вам будет предложено ввести пароль от недавно созданной базы данных.

На PostgreSQL:

```
# zcat /usr/share/doc/zabbix-server-pgsql/create.sql.gz | psql -U <username> zabbix
```

Для прокси импортируйте изначальную схему:

```
# zcat /usr/share/doc/zabbix-proxy-mysql/create.sql.gz | mysql -uzabbix -p zabbix
```

Для прокси на PostgreSQL (или SQLite):

```
# zcat /usr/share/doc/zabbix-proxy-pgsql/create.sql.gz | psql -U <username> zabbix  
# zcat /usr/share/doc/zabbix-proxy-sqlite/create.sql.gz | sqlite3 zabbix.db
```

Настройка базы данных для Zabbix сервера/прокси

Измените zabbix_server.conf или zabbix_proxy.conf для использования созданной базы данных.

Например:

```
# vi /etc/zabbix/zabbix_server.conf  
DBHost=localhost  
DBName=zabbix  
DBUser=zabbix  
DBPassword=<пароль>
```

В параметре DBPassword используйте пароль от MySQL базы данных Zabbix; пароль пользователя PostgreSQL для PostgreSQL.

Используйте DBHost= для PostgreSQL. Вы возможно, захотите оставить настройку по умолчанию DBHost=localhost (или IP адрес), но в этом случае PostgreSQL будет использовать сетевой сокет при подключении к Zabbix. Обратитесь к [соответствующему разделу](#) по RHEL/CentOS для получения детальных инструкций.

Запуск процесса Zabbix сервера

Самое время запустить процесс Zabbix сервера и добавить его в автозагрузку:

```
# service zabbix-server start  
# update-rc.d zabbix-server enable
```

Замените 'zabbix-server' на 'zabbix-proxy' для запуска процесса Zabbix прокси.

Настройка SELinux

Обратитесь к [соответствующему разделу](#) по RHEL/CentOS.

После завершения настройки веб-интерфейса и SELinux вам необходимо перезапустить веб-сервер Apache:

```
# service apache2 restart
```

Настройка веб-интерфейса

Файл конфигурации Apache для Zabbix веб-интерфейса располагается в /etc/apache2/conf-enabled/zabbix.conf. Некоторые настройки PHP уже выполнены. Однако, необходимо раскомментировать "date.timezone" настройку и [указать корректный для вас часовой пояс](#).

```
php_value max_execution_time 300  
php_value memory_limit 128M  
php_value post_max_size 16M  
php_value upload_max_filesize 2M  
php_value max_input_time 300  
php_value max_input_vars 10000  
php_value always_populate_raw_post_data -1  
# php_value date.timezone Europe/Riga
```

Теперь вы готовы приступить к выполнению [шагов по установке веб-интерфейса](#), которые позволят вам получить доступ к недавно установленному Zabbix.

Установка агента

Для установки агента выполните

```
# apt install zabbix-agent
```

Для запуска агента выполните:

```
# service zabbix-agent start
```

Установка Java gateway

Необходимо установить [Java gateway](#), если вы хотите мониторить JMX приложения. Java gateway легковесный и не требует наличия базы данных.

Как только требуемый [репозиторий](#) будет добавлен, вы сможете установить Zabbix Java gateway выполнив команду:

```
# apt install zabbix-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](#) конфигурации Zabbix Java gateway.

Чтобы запустить Zabbix Java gateway:

```
# service zabbix-java-gateway restart
```

Обратите внимание, что Zabbix сервер [нужно настроить](#) для работы с Java gateway.

Чтобы Zabbix Java gateway автоматически запускался при загрузке системы:

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

Файл журнала Zabbix Java gateway:

/var/log/zabbix/zabbix_java_gateway.log

Если вы хотите увеличить уровень журналирования, измените 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>
```

4.5 Установка из контейнеров

Docker

Zabbix предоставляет [Docker](#) образы для каждого компонента Zabbix как переносимые и самодостаточные контейнеры для ускорения развертывания и процедуры обновления.

Zabbix компоненты поставляются с поддержкой MySQL и PostgreSQL баз данных, поддержкой Apache2 и Nginx веб-серверов. Эти образы разделены на несколько различных образом.

Базовые образы Docker

Zabbix компоненты поставляются на основе Ubuntu, Alpine Linux и CentOS образов:

Образ Версия

[alpine](#) 3.4

[ubuntu](#) trusty

[centos](#) latest

Все образы сконфигурированы таким образом, что будут пересозданы в случае, если базовые образы обновятся.

Образы Zabbix агента, Zabbix прокси, Zabbix сервера и Zabbix готового решения основаны на Alpine Linux версии 3.4, все остальные образы основаны на Alpine Linux 3.7.

Исходные файлы Docker

Каждый может следить за изменениями Docker файлов в [официальном репозитории](#) Zabbix на [github.com](#). Вы можете сделать ответвления от проекта или создавать свои собственные образы на основе официальных Docker файлов.

Структура

Все компоненты Zabbix доступны в следующих Docker репозиториях:

- Готовое решение Zabbix с поддержкой MySQL базы данных и Nginx веб-сервером - [zabbix/zabbix-appliance](#)
- Zabbix агент - [zabbix/zabbix-agent](#)
- Zabbix сервер
 - Zabbix сервер с поддержкой MySQL базы данных - [zabbix/zabbix-server-mysql](#)
 - Zabbix сервер с поддержкой PostgreSQL базы данных - [zabbix/zabbix-server-pgsql](#)
- Zabbix веб-интерфейс
 - Zabbix веб-интерфейс на основе Apache2 веб-сервера с поддержкой MySQL базы данных - [zabbix/zabbix-web-apache-mysql](#)
 - Zabbix веб-интерфейс на основе Apache2 веб-сервера с поддержкой PostgreSQL базы данных - [zabbix/zabbix-web-apache-pgsql](#)
 - Zabbix веб-интерфейс на основе Nginx веб-сервера с поддержкой MySQL базы данных - [zabbix/zabbix-web-nginx-mysql](#)
 - Zabbix веб-интерфейс на основе Nginx веб-сервера с поддержкой PostgreSQL базы данных - [zabbix/zabbix-web-nginx-pgsql](#)
- Zabbix прокси
 - Zabbix прокси с поддержкой SQLite3 базы данных - [zabbix/zabbix-proxy-sqlite3](#)
 - Zabbix прокси с поддержкой MySQL базы данных - [zabbix/zabbix-proxy-mysql](#)
- Zabbix Java Gateway - [zabbix/zabbix-java-gateway](#)

Дополнительно имеется поддержка SNMP трапов. Она поставляется как отдельный репозиторий ([zabbix/zabbix-snmptraps](#)) основанный только на Ubuntu Trusty. Его можно связать с Zabbix сервером или Zabbix прокси.

Версии

Каждый репозиторий Zabbix компонент содержит следующие тэги:

- **latest** - последняя стабильная версия Zabbix компоненты на основе образа Alpine Linux
- **alpine-latest** - последняя стабильная версия Zabbix компоненты на основе образа Alpine Linux
- **ubuntu-latest** - последняя стабильная версия Zabbix компоненты на основе образа Ubuntu
- **alpine-4.0-latest** - последняя минорная версия Zabbix 4.0 компоненты на основе образа Alpine Linux
- **ubuntu-4.0-latest** - последняя минорная версия Zabbix 4.0 компоненты на основе образа Ubuntu
- **alpine-4.0.*** - различные минорные версии Zabbix 4.0 компоненты на основе образа Alpine Linux, где * - минорная версия Zabbix компоненты
- **ubuntu-4.0.*** - различные минорные версии Zabbix 4.0 компоненты на основе образа Ubuntu, где * - минорная версия Zabbix компоненты

Использование

Переменные окружения

Все образы Zabbix компонент предусматривают наличие переменных окружения для управления конфигурацией. Эти переменные окружения перечислены в репозиториях каждого компонента. Переменные окружения являются опциями из файлов конфигурации Zabbix, но имеют немного другое наименование. Например, `ZBX_LOGSLOWQUERIES` идентичен `LogSlowQueries` из Zabbix сервера и Zabbix прокси файлов конфигурации.

Некоторые опции конфигурации не разрешены для изменения. Например, `PIDFile` и `LogType`.

Некоторые компоненты имеют специфичные переменные окружения, которые отсутствуют в официальных файлах конфигурации Zabbix:

Переменная	Компоненты	Описание
DB_SERVER_HOST	Сервер Прокси Веб-интерфейс	Переменная является IP адресом или DNS именем MySQL или PostgreSQL сервера. По умолчанию, значение mysql-server или postgres-server для MySQL или PostgreSQL соответственно
DB_SERVER_PORT	Сервер Прокси Веб-интерфейс	Переменная является портом MySQL или PostgreSQL сервера. По умолчанию, значение '3306' или '5432' соответственно.
MYSQL_USER	Сервер Proxy Веб-интерфейс	Имя пользователя к базе данных MySQL. По умолчанию, значение 'zabbix'.
MYSQL_PASSWORD	Сервер Прокси Веб-интерфейс	Пароль к базе данных MySQL. По умолчанию, значение 'zabbix'.
MYSQL_DATABASE	Сервер Прокси Веб-интерфейс	Имя базы данных Zabbix. По умолчанию, значение 'zabbix' для Zabbix сервера и 'zabbix_proxy' для Zabbix прокси.
POSTGRES_USER	Сервер Веб-интерфейс	Имя пользователя к базе данных PostgreSQL. По умолчанию, значение 'zabbix'.
POSTGRES_PASSWORD	Сервер Веб-интерфейс	Пароль к базе данных PostgreSQL. По умолчанию, значение 'zabbix'.
RD	Сервер Веб-интерфейс	Имя базы данных Zabbix. По умолчанию, значение 'zabbix'.
POSTGRES_DB	Сервер Веб-интерфейс	Имя базы данных Zabbix. По умолчанию, значение 'zabbix' для Zabbix сервера и 'zabbix_proxy' для Zabbix прокси.
TZ	Веб-интерфейс	Часовой пояс в PHP формате. Полный список поддерживаемых часовых поясов доступен на php.net . По умолчанию, значение 'Europe/Riga'.
ZBX_SERVER_NAME	Веб-интерфейс	Видимое имя Zabbix инсталляции в верхнем правом углу веб-интерфейса. По умолчанию, значение 'Zabbix Docker'
ZBX_JAVAGATEWAY_ENABLE	Сервер Прокси	Включение взаимодействия с Zabbix Java gateway для сбора проверок связанных с Java. По умолчанию, значение "false"
ZBX_ENABLE_SNMP_TRAPS	Сервер Прокси	Включение функции SNMP трапов. Эта переменная требует zabbix-snmptraps экземпляр и разделяемый том /var/lib/zabbix/snmptraps с Zabbix сервером или Zabbix прокси.

Тома

Образы позволяют использовать некоторые точки мониторинга. Такие точки мониторинга различны и зависят от типа Zabbix компонента:

Том	Описание
Zabbix агент	
/etc/zabbix/zabbix_agentd.d	Том позволяет включать *.conf файлы и расширять Zabbix агент используя функционал UserParameter
/var/lib/zabbix/modules	Том позволяет загружать дополнительные модули и расширять Zabbix агент используя функционал LoadModule
/var/lib/zabbix/enc	Том используется для хранения файлов связанных с TLS функционалом. Эти имена файлов указываются при помощи переменных окружения ZBX_TLSCAFILE, ZBX_TLSCRLFILE, ZBX_TLSKEY_FILE и ZBX_TLSPSKFILE
Zabbix сервер	
/usr/lib/zabbix/	Том используется для пользовательских скриптов оповещения. Том является

<code>alerts</code>	параметром <code>AlertScriptsPath</code> в zabbix_server.conf
<code>/usr/lib/zabbix/externalscripts</code>	Том используется для внешних проверок . Том является параметром <code>ExternalScripts</code> в zabbix_server.conf
<code>/var/lib/zabbix/modules</code>	Том позволяет загружать дополнительные модули и расширять Zabbix сервер используя функционал LoadModule
<code>/var/lib/zabbix/enc</code>	Том используется для хранения файлов связанных с TLS функционалом. Эти имена файлов указываются при помощи переменных окружения <code>ZBX_TLSCAFILE</code> , <code>ZBX_TLSCRLFILE</code> , <code>ZBX_TLSKEY_FILE</code> и <code>ZBX_TLSPSKFILE</code>
<code>/var/lib/zabbix/ssl/certs</code>	Том используется для размещения файлов клиентских SSL сертификатов для аутентификации клиентов. Этот том является аналогом параметра <code>SSLCertLocation</code> в zabbix_server.conf
<code>/var/lib/zabbix/ssl/keys</code>	Том используется для размещения файлов приватных SSL ключей для аутентификации клиентов. Этот том является аналогом параметра <code>SSLKeyLocation</code> в zabbix_server.conf
<code>/var/lib/zabbix/ssl/ssl_ca</code>	Том используется для размещения файлов центра сертификации (CA) для верификации SSL сертификатов сервера. Этот том является аналогом параметра <code>SSLCALocation</code> в zabbix_server.conf
<code>/var/lib/zabbix/snmptraps</code>	Том используется как размещение <code>snmptraps.log</code> файла. Этот том может быть использован совместо с <code>zabbix-snmptraps</code> контейнером и унаследован из <code>volumes_from</code> опции Docker при создании нового экземпляра Zabbix сервера. Функцию обработки SNMP трапов можно включить использовав совместный том и переключив <code>ZBX_ENABLE_SNMP_TRAPS</code> переменную окружения в 'true'
<code>/var/lib/zabbix/mibs</code>	Том позволяет добавлять новые MIB файлы. Функция не поддерживает подпапки, все MIB файлы должны быть помещены в <code>/var/lib/zabbix/mibs</code>
Zabbix прокси	
<code>/usr/lib/zabbix/externalscripts</code>	Том используется для внешних проверок . Том является параметром <code>ExternalScripts</code> в zabbix_proxy.conf
<code>/var/lib/zabbix/modules</code>	Том позволяет загружать дополнительные модули и расширять Zabbix прокси используя функционал LoadModule
<code>/var/lib/zabbix/enc</code>	Том используется для хранения файлов связанных с TLS функционалом. Эти имена файлов указываются при помощи переменных окружения <code>ZBX_TLSCAFILE</code> , <code>ZBX_TLSCRLFILE</code> , <code>ZBX_TLSKEY_FILE</code> и <code>ZBX_TLSPSKFILE</code>
<code>/var/lib/zabbix/ssl/certs</code>	Том используется для размещения файлов клиентских SSL сертификатов для аутентификации клиентов. Этот том является аналогом параметра <code>SSLCertLocation</code> в zabbix_proxy.conf
<code>/var/lib/zabbix/ssl/keys</code>	Том используется для размещения файлов клиентских SSL сертификатов для аутентификации клиентов. Этот том является аналогом параметра <code>SSLCertLocation</code> в zabbix_proxy.conf
<code>/var/lib/zabbix/ssl/ssl_ca</code>	Том используется для размещения файлов центра сертификации (CA) для верификации SSL сертификатов сервера. Этот том является аналогом параметра <code>SSLCALocation</code> в zabbix_proxy.conf
<code>/var/lib/zabbix/snmptraps</code>	Том используется как размещение <code>snmptraps.log</code> файла. Этот том может быть использован совместо с <code>zabbix-snmptraps</code> контейнером и унаследован из <code>volumes_from</code> опции Docker при создании нового экземпляра Zabbix прокси. Функцию обработки SNMP трапов можно включить использовав совместный том и переключив <code>ZBX_ENABLE_SNMP_TRAPS</code> переменную окружения в 'true'
<code>/var/lib/zabbix/mibs</code>	Том позволяет добавлять новые MIB файлы. Функция не поддерживает подпапки, все MIB файлы должны быть помещены в <code>/var/lib/zabbix/mibs</code>
Zabbix веб-интерфейс на основе Apache2 веб-сервера	
<code>/etc/ssl/apache2</code>	Том позволяет активировать HTTPS для Zabbix веб-интерфейса. Этот том должен содержать два файла <code>ssl.crt</code> и <code>ssl.key</code> , подготовленные для Apache2 SSL соединений
Zabbix веб-интерфейс на основе Nginx веб-сервера	

Том позволяет активировать HTTPS для Zabbix веб-интерфейса. Этот том должен содержать три файла `ssl.crt`, `ssl.key` и `dhpam.pem` подготовленные для Nginx SSL соединений

Zabbix snmptraps

`/var/lib/zabbix/snmptraps` Том содержит файл журнала с именем `snmptraps.log` с полученными SNMP трапами
`/var/lib/zabbix/mibs` Том позволяет добавлять новые MIB файлы. Функция не поддерживает подпапки, все MIB файлы должны быть помещены в `/var/lib/zabbix/mibs`

Для получения дополнительной информации обратитесь к официальными репозиториям Zabbix в Docker Hub.

Примеры использования

Пример 1

Этот пример демонстрирует как запустить готовое решение Zabbix со встроенными MySQL базой данных, Zabbix сервером, Zabbix веб-интерфейсов на основе Nginx веб-сервера и Zabbix Java gateway.

```
# docker run --name zabbix-appliance -t \
    -p 10051:10051 \
    -p 80:80 \
    -d zabbix/zabbix-appliance:latest
```

Экземпляр готового решения Zabbix раскрывает 10051/TCP порт (Zabbix траппера) и 80/TCP порт (HTTP) на хост машину.

Пример 2

Этот пример демонстрирует как запустить Zabbix сервер с поддержкой MySQL базы данных, Zabbix веб-интерфейсом на основе Nginx веб-сервера и Zabbix Java gateway.

1. Запустите пустой экземпляр MySQL сервера

```
# 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" \
    -d mysql:5.7 \
    --character-set-server=utf8 --collation-server=utf8_bin
```

2. Запустите экземпляр Zabbix Java gateway

```
# docker run --name zabbix-java-gateway -t \
    -d zabbix/zabbix-java-gateway:latest
```

3. Запустите экземпляр Zabbix сервера и соедините этот экземпляр с недавно созданным экземпляром MySQL сервера

```
# 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" \
    --link mysql-server:mysql \
    --link zabbix-java-gateway:zabbix-java-gateway \
    -p 10051:10051 \
    -d zabbix/zabbix-server-mysql:latest
```

Экземпляр Zabbix сервера раскрывает 10051/TCP порт (Zabbix траппера) на хост машину.

4. Запустите Zabbix веб-интерфейс и соедините этот экземпляр с недавно созданными экземплярами MySQL сервера и Zabbix сервера

```
# docker run --name zabbix-web-nginx-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" \
--link mysql-server:mysql \
--link zabbix-server-mysql:zabbix-server \
-p 80:80 \
-d zabbix/zabbix-web-nginx-mysql:latest
```

Экземпляр Zabbix веб-интерфейс раскрывает 80/TCP порт (HTTP) на хост машину.

Пример 3

Этот пример демонстрирует как запустить Zabbix сервер с поддержкой PostgreSQL базы данных, Zabbix веб-интерфейсом на основе Nginx веб-сервера и с функцией приёма SNMP трапов.

1. Запустите пустой экземпляр PostgreSQL сервера

```
# docker run --name postgres-server -t \
-e POSTGRES_USER="zabbix" \
-e POSTGRES_PASSWORD="zabbix" \
-e POSTGRES_DB="zabbix_pwd" \
-d postgres:latest
```

2. 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 \
-p 162:162/udp \
-d zabbix/zabbix-snmptraps:latest
```

Экземпляр Zabbix snmptraps раскрывает 162/UDP порт (SNMP трапы) на хост машину.

3. Запустите экземпляр Zabbix сервера и соедините этот экземпляр с недавно созданным экземпляром PostgreSQL сервера

```
# docker run --name zabbix-server-pgsql -t \
-e DB_SERVER_HOST="postgres-server" \
-e POSTGRES_USER="zabbix" \
-e POSTGRES_PASSWORD="zabbix" \
-e POSTGRES_DB="zabbix_pwd" \
-e ZBX_ENABLE_SNMP_TRAPS="true" \
--link postgres-server:postgres \
-p 10051:10051 \
--volumes-from zabbix-snmptraps \
-d zabbix/zabbix-server-pgsql:latest
```

Экземпляр Zabbix сервера раскрывает 10051/TCP порт (Zabbix трассера) на хост машину.

4. Запустите Zabbix веб-интерфейс и соедините этот экземпляр с недавно созданными экземплярами PostgreSQL сервера и Zabbix сервера

```
# docker run --name zabbix-web-nginx-pgsql -t \
-e DB_SERVER_HOST="postgres-server" \
-e POSTGRES_USER="zabbix" \
-e POSTGRES_PASSWORD="zabbix" \
-e POSTGRES_DB="zabbix_pwd" \
--link postgres-server:postgres \
--link zabbix-server-pgsql:zabbix-server \
-p 443:443 \
-v /etc/ssl/nginx:/etc/ssl/nginx:ro \
-d zabbix/zabbix-web-nginx-pgsql:latest
```

Экземпляр Zabbix веб-интерфейса раскрывает 443/TCP порт (HTTPS) на хост машину.

Папка `/etc/ssl/nginx` должна содержать сертификат с требуемым именем.

Docker Compose

Zabbix также поставляет файлы наборов для определения и запуска нескольких контейнеров Zabbix компонент в Docker. Такие файлы наборов доступны в официальном репозитории Zabbix docker на [github.com: https://github.com/zabbix/zabbix-docker](https://github.com/zabbix/zabbix-docker). Они добавлены как примеры, эти файлы перегружены. Например, они содержат прокси с поддержкой MySQL и SQLite3.

Имеется несколько различных версий файлов наборов:

Имя файла	Описание
docker-compose_v3_alpine_mysql_localtest.yaml	Файл набора запускает последнюю версию компонент Zabbix 4.0 на Alpine Linux с поддержкой MySQL базы данных.
docker-compose_v3_alpine_mysql_local.yaml	Файл набора собирает локально последнюю версию Zabbix 4.0 и запускает компоненты Zabbix на Alpine Linux с поддержкой MySQL базы данных.
docker-compose_v3_alpine_pgsql_localtest.yaml	Файл набора запускает последнюю версию компонент Zabbix 4.0 на Alpine Linux с поддержкой PostgreSQL базы данных.
docker-compose_v3_alpine_pgsql_local.yaml	Файл набора собирает локально последнюю версию Zabbix 4.0 и запускает компоненты Zabbix на Alpine Linux с поддержкой PostgreSQL базы данных.
docker-compose_v3_centos_mysql_localtest.yaml	Файл набора запускает последнюю версию компонент Zabbix 4.0 на CentOS 7 с поддержкой MySQL базы данных.
docker-compose_v3_centos_mysql_local.yaml	Файл набора собирает локально последнюю версию Zabbix 4.0 и запускает компоненты Zabbix на CentOS 7 с поддержкой MySQL базы данных.
docker-compose_v3_centos_pgsql_localtest.yaml	Файл набора запускает последнюю версию компонент Zabbix 4.0 на CentOS 7 с поддержкой PostgreSQL базы данных.
docker-compose_v3_centos_pgsql_local.yaml	Файл набора собирает локально последнюю версию Zabbix 4.0 и запускает компоненты Zabbix на CentOS 7 с поддержкой PostgreSQL базы данных.
docker-compose_v3_ubuntu_mysql_localtest.yaml	Файл набора запускает последнюю версию компонент Zabbix 4.0 на Ubuntu 14.04 с поддержкой MySQL базы данных.
docker-compose_v3_ubuntu_mysql_local.yaml	Файл набора собирает локально последнюю версию Zabbix 4.0 и запускает компоненты Zabbix на Ubuntu 14.04 с поддержкой MySQL базы данных.
docker-compose_v3_ubuntu_pgsql_localtest.yaml	Файл набора запускает последнюю версию компонент Zabbix 4.0 на Ubuntu 14.04 с поддержкой PostgreSQL базы данных.
docker-compose_v3_ubuntu_pgsql_local.yaml	Файл набора собирает локально последнюю версию Zabbix 4.0 и запускает компоненты Zabbix на Ubuntu 14.04 с поддержкой PostgreSQL базы данных.

Доступные файлы наборов Docker поддерживают как версию 2, так и версию 3 Docker Compose.

Хранилище

Файлы наборов сконфигурированы для поддержки локального хранилища на хост машине. Docker Compose создаст директорию `zbx_env` в папке с файлом набора, когда вы запустите компоненты Zabbix с использованием файла набора. Директория будет содержать ту же структуру, что и описано выше в разделе [Тома](#) и директорию для хранения базы данных.

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

Файлы окружения

В той же директории с файлами наборов на github.com вы можете найти файлы с переменными окружения по умолчанию для каждой компоненты из файла наборов. Эти файлы окружения называются примерно так `.env_<тип компоненты>`.

Примеры

Пример 1

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

Эта команда загрузит последние образы Zabbix 4.0 для каждой Zabbix компоненты и затем запустит их в detach режиме.

Не забудьте загрузить `.env_<тип компонента>` файлы с github.com официального репозитория Zabbix вместе с файлами наборов.

Пример 2

```
# docker-compose -f ./docker-compose_v3_ubuntu_mysql_local.yaml up -d
```

Эта команда загрузит базовый образ Ubuntu 14.04, затем соберет Zabbix 4.0 компоненты локально и запустит их в detach режиме.

4.6 Процедура обновления

Обзор

Этот раздел описывает информацию об обновлении для **Zabbix 4.0**:

- с использованием пакетов:
 - для [Red Hat Enterprise Linux/CentOS](#)
 - для [Debian/Ubuntu](#)
- с использованием [исходного кода](#)

Обновление напрямую до Zabbix 4.0 возможно с Zabbix **3.4.x, 3.2.x, 3.0.x, 2.4.x, 2.2.x и 2.0.x**. При обновлении с более ранних версий обратитесь к документации Zabbix по 2.0 и более ранним версиям.

Обновление с использованием пакетов

Обзор

В этом разделе описаны шаги, которые требуется выполнить для успешного [обновления](#) с использованием официальных RPM и DEB пакетов, которые поставляются Zabbix для:

- [Red Hat Enterprise Linux/CentOS](#)
- [Debian/Ubuntu](#)

1 Red Hat Enterprise Linux/CentOS

Обзор

Этот раздел описывает требуемые шаги для успешного [обновления](#) с Zabbix **3.4.x** до Zabbix **4.0.x** с использованием официальных пакетов Zabbix для Red Hat Enterprise Linux/CentOS.

В то время как обновление Zabbix агентов не является обязательным шагом (но рекомендуемым), Zabbix сервер и прокси должны быть [одной мажорной версии](#). Поэтому, при наличии сервер-прокси инсталляции, Zabbix сервер и все прокси должны быть остановлены и обновлены.

Чтобы минимизировать время простоя и потери данных в процессе обновления, рекомендуется остановить и обновить Zabbix сервер и один за другим затем остановить, обновить и запустить Zabbix прокси. Когда все прокси будут обновлены, запустите Zabbix сервер. Пока Zabbix сервер будет остановлен, запущенные прокси будут продолжать сбор и хранение данных и отправят эти данные на Zabbix сервер, когда сервер запустится и заработает. Любые оповещения о проблемах в процессе простоя Zabbix сервера будут сгенерированы только после того как обновленный сервер запустится.

Обратите внимание, что при наличии прокси с SQLite базой данных, данные истории этих прокси будут потеряны до обновления, так как обновление файла базы данных SQLite не поддерживается и этот файл необходимо удалить вручную. Когда прокси запускается в первый раз и будет отсутствовать файл базы данных SQLite, прокси создаст его автоматически.

В зависимости от размера базы данных обновление базы данных до версии 4.0 может занять продолжительное время.

Перед выполнением обновления убедитесь что прочитали все соответствующие **заметки по обновлению!**

Доступны следующие заметки по обновлению:

Обновление с	Прочтайте заметки по обновлению полностью	Важные заметки/изменения между версиями
3.4.x	Для 4.0	Библиотеки 'libpthread' и 'zlib' теперь обязательны; Поддержка протокола в виде простого текста убрана и заголовок обязателен; Zabbix агенты версий Pre-1.4 более не поддерживаются; Параметр Server в конфигурации пассивного прокси теперь обязательен
3.2.x	Также для 3.4	Поддержка SQLite в виде основной базы данных убрана для Zabbix сервера/веб-интерфейса; Поддерживается Perl совместимые регулярные выражения (PCRE) вместо POSIX расширенных; Библиотеки 'libpcre' and 'libevent' обязательны для Zabbix сервера; Добавлены проверки кода выхода для пользовательских параметров, удаленных команд и элементов данных system.run[] без 'nowait' флага, а также для выполняемых скриптов Zabbix сервером; Zabbix Java gateway необходимо обновить для поддержки новых функций
3.0.x	Также для 3.2	Обновление базы данных может быть медленным, в зависимости от размеров таблиц истории

Вы возможно захотите также проверить [требования](#) для 4.0.

Возможно удобно запустить две параллельные SSH сессии на время обновления, выполняя шаги обновления в одной сессии и наблюдая за файлами журналов сервера/прокси в другой. Например, при выполнении `tail -f zabbix_server.log` или `tail -f zabbix_proxy.log` во второй SSH сессии будут отображаться последние записи из файла журнала и возможные ошибки в режиме реального времени. Такой подход может быть критичным на продуктивных серверах.

Процесс обновления

1 Остановите Zabbix процессы

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

```
# systemctl stop zabbix-server
```

При обновлении прокси, остановите также и его.

```
# systemctl stop zabbix-proxy
```

Известно, что имеется возможность запустить обновленный сервер с ещё не обновленными прокси, которые будут отправлять данные на новый сервер (хотя прокси и не смогут обновлять свою конфигурацию). Однако, такой подход не рекомендуется и не поддерживается Zabbix, его выбор исключительно на ваш страх и риск.

2 Сделайте архивную копию существующей базы данных Zabbix

Этот шаг очень важен. Убедитесь, что у вас есть архивная копия вашей базы данных. Это поможет, если процедура обновления закончится неудачно (отсутствие свободного места на диске, выключение питания, любая неожиданная 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.

Файлы конфигурации:

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

Файлы PHP и бинарные файлы Zabbix:

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

4 Обновите пакет конфигурации репозитория

Чтобы продолжить обновление, необходимо обновить пакет текущий репозитория.

```
# rpm -Uvh https://repo.zabbix.com/zabbix/4.0/rhel/7/x86_64/zabbix-release-4.0-1.el7.centos.noarch.rpm
```

5 Обновите компоненты Zabbix

Для обновления компонентов Zabbix вы можете выполнить что-то вроде:

```
# yum upgrade zabbix-server-mysql zabbix-web-mysql zabbix-agent
```

Если используете PostgreSQL, в команде замените mysql на postgresql. Если обновляете прокси, в команде замените server на proxy.

6 Просмотрите параметры конфигурации компонент

Для получения более подробных сведений смотрите заметки по обновлению на предмет [обязательных изменений](#).

7 Запустите процессы Zabbix

Запустите обновленные компоненты Zabbix.

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

8 Очистите cookies и кэш в веб-браузере

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

Обновление между минорными версиями

Имеется возможность обновления между минорными версиями 4.0.x (например, с 4.0.1 на 4.0.3). Процедура минорного обновления Zabbix очень проста.

Для выполнения минорного обновления Zabbix необходимо выполнить:

```
$ sudo yum upgrade 'zabbix-*'
```

Для выполнения обновления минорной версии Zabbix сервера выполните:

```
$ sudo yum upgrade 'zabbix-server-*'
```

Для выполнения обновления минорной версии Zabbix агента выполните:

```
$ sudo yum upgrade 'zabbix-agent-*'
```

Обратите внимание, что в этих командах вы можете также использовать 'update' вместо 'upgrade'. В то время как 'upgrade' удалит устаревшие пакеты, 'update' сохранит их.

2 Debian/Ubuntu

Overview

Этот раздел описывает требуемые шаги для успешного [обновления](#) с Zabbix 3.4.x до Zabbix 4.0.x с использованием официальных пакетов Zabbix для Debian/Ubuntu.

В то время как обновление Zabbix агентов не является обязательным шагом (но рекомендуемым), Zabbix сервер и прокси должны быть [одной мажорной версии](#). Поэтому, при наличии сервер-прокси инсталляции, Zabbix сервер и все прокси должны быть остановлены и обновлены.

Чтобы минимизировать время простоя и потери данных в процессе обновления, рекомендуется остановить и обновить Zabbix сервер и один за другим затем остановить, обновить и запустить Zabbix прокси. Когда все прокси будут обновлены, запустите Zabbix сервер. Пока Zabbix сервер будет остановлен, запущенные прокси будут продолжать сбор и хранение данных и отправят эти данные на Zabbix сервер, когда сервер запустится и заработает. Любые оповещения о проблемах в процессе простоя Zabbix сервера будут сгенерированы только после того как обновленный сервер запустится.

Обратите внимание, что при наличии прокси с SQLite базой данных, данные истории этих прокси будут потеряны до обновления, так как обновление файла базы данных SQLite не поддерживается и этот файл необходимо удалить вручную. Когда прокси запускается в первый раз и будет отсутствовать файл базы данных SQLite, прокси создаст его автоматически.

В зависимости от размера базы данных обновление базы данных до версии 4.0 может занять продолжительное время.

Перед выполнением обновления убедитесь что прочитали все соответствующие **заметки по обновлению!**

Доступны следующие заметки по обновлению:

Обновление с	Прочтайте заметки по обновлению полностью	Важные заметки/изменения между версиями
3.4.x	Для 4.0	Библиотеки 'libpthread' и 'zlib' теперь обязательны; Поддержка протокола в виде простого текста убрана и заголовок обязателен; Zabbix агенты версий Pre-1.4 более не поддерживаются; Параметр Server в конфигурации пассивного прокси теперь обязателен
3.2.x	Также для 3.4	Поддержка SQLite в виде основной базы данных убрана для

Обновление с	Прочтайте заметки по обновлению полностью	Важные заметки/изменения между версиями
3.0.x	Также для 3.2	Zabbix сервера/веб-интерфейса; Поддерживается Perl совместимые регулярные выражения (PCRE) вместо POSIX расширенных; Библиотеки 'libpcre' and 'libevent' обязательны для Zabbix сервера; Добавлены проверки кода выхода для пользовательских параметров, удаленных команд и элементов данных system.run[] без 'nowait' флага, а также для выполняемых скриптов Zabbix сервером; Zabbix Java gateway необходимо обновить для поддержки новых функций

Обновление базы данных может быть медленным, в зависимости от размеров таблиц истории

Вы возможно захотите также проверить [требования](#) для 4.0.

Возможно удобно запустить две параллельные SSH сессии на время обновления, выполняя шаги обновления в одной сессии и наблюдая за файлами журналов сервера/прокси в другой. Например, при выполнении `tail -f zabbix_server.log` или `tail -f zabbix_proxy.log` во второй SSH сессии будут отображаться последние записи из файла журнала и возможные ошибки в режиме реального времени. Такой подход может быть критичным на продуктивных серверах.

Процесс обновления

1 Остановите Zabbix процессы

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

```
# service zabbix-server stop
```

При обновлении прокси, остановите также и его.

```
# service zabbix-proxy stop
```

Известно, что имеется возможность запустить обновленный сервер с ещё не обновленными прокси, которые будут отправлять данные на новый сервер (хотя прокси и не смогут обновлять свою конфигурацию). Однако, такой подход не рекомендуется и не поддерживается Zabbix, его выбор исключительно на ваш страх и риск.

2 Сделайте архивную копию существующей базы данных Zabbix

Этот шаг очень важен. Убедитесь, что у вас есть архивная копия вашей базы данных. Это поможет, если процедура обновления закончится неудачно (отсутствие свободного места на диске, выключение питания, любая неожиданная 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.

Файлы конфигурации:

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

Файлы PHP и бинарные файлы Zabbix:

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

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

4 Обновите пакет конфигурации репозитория

Чтобы продолжить обновление, необходимо удалить пакет текущий репозитория.

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

Затем установить пакет нового репозитория.

Для **Debian 9** выполните:

```
# wget https://repo.zabbix.com/zabbix/3.5/debian/pool/main/z/zabbix-release/zabbix-release_3.5-1+stretch_all.deb  
# dpkg -i zabbix-release_3.5-1+stretch_all.deb
```

Для **Debian 8** выполните:

```
# wget https://repo.zabbix.com/zabbix/3.5/debian/pool/main/z/zabbix-release/zabbix-release_3.5-1+jessie_all.deb  
# dpkg -i zabbix-release_3.5-1+jessie_all.deb
```

Для **Debian 7** выполните:

```
# wget https://repo.zabbix.com/zabbix/3.5/debian/pool/main/z/zabbix-release/zabbix-release_3.5-1+wheezy_all.deb  
# dpkg -i zabbix-release_3.5-1+wheezy_all.deb
```

Для **Ubuntu 18.04** выполните:

```
# wget https://repo.zabbix.com/zabbix/3.5/ubuntu/pool/main/z/zabbix-release/zabbix-release_3.5-1+bionic_all.deb  
# dpkg -i zabbix-release_3.5-1+bionic_all.deb
```

Для **Ubuntu 16.04** выполните:

```
# wget https://repo.zabbix.com/zabbix/3.5/ubuntu/pool/main/z/zabbix-release/zabbix-release_3.5-1+xenial_all.deb  
# dpkg -i zabbix-release_3.5-1+xenial_all.deb
```

Для **Ubuntu 14.04** выполните:

```
# wget https://repo.zabbix.com/zabbix/3.5/ubuntu/pool/main/z/zabbix-release/zabbix-release_3.5-1+trusty_all.deb  
# dpkg -i zabbix-release_3.5-1+trusty_all.deb
```

Обновите информацию о репозитории.

```
# apt-get update
```

5 Обновите компоненты Zabbix

Для обновления компонентов Zabbix вы можете выполнить что-то вроде:

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

Если используете PostgreSQL, в команде замените `mysql` на `pgsql`. Если обновляете прокси, в команде замените `server` на `proxy`.

6 Просмотрите параметры конфигурации компонент

Для получения более подробных сведений смотрите заметки по обновлению на предмет [обязательных изменений](#).

Для ознакомления с новыми необязательными параметрами, пожалуйста, смотрите раздел [Что нового](#).

7 Запустите процессы Zabbix

Запустите обновленные компоненты Zabbix.

```
# service zabbix-server start  
# service zabbix-proxy start  
# service zabbix-agent start
```

8 Очистите cookies и кэш в веб-браузере

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

Обновление между минорными версиями

Имеется возможность обновления минорные версии 4.0.x (например, с 4.0.1 на 4.0.3). Процедура очень проста.

Для минорного обновления Zabbix, пожалуйста, выполните:

```
$ sudo apt install --only-upgrade 'zabbix.*'
```

Для обновления минорной версии Zabbix сервера, пожалуйста, выполните:

```
$ sudo apt install --only-upgrade 'zabbix-server.*'
```

Для обновления минорной версии Zabbix агента, пожалуйста, выполните:

```
$ sudo apt install --only-upgrade 'zabbix-agent.*'
```

Обновление из исходных кодов

Обзор

Этот раздел описывает требуемые шаги для успешного [обновления](#) с Zabbix 3.4.x до Zabbix 4.0.x с использованием официальных исходных кодов Zabbix.

В то время как обновление Zabbix агентов не является обязательным шагом (но рекомендуемым), Zabbix сервер и прокси должны быть [одной мажорной версии](#). Поэтому, при наличии сервер-прокси инсталляции, Zabbix сервер и все прокси должны быть остановлены и обновлены.

Чтобы минимизировать время простоя и потери данных в процессе обновления, рекомендуется остановить и обновить Zabbix сервер и один за другим затем остановить, обновить и запустить Zabbix прокси. Когда все прокси будут обновлены, запустите Zabbix сервер. Пока Zabbix сервер будет остановлен, запущенные прокси будут продолжать сбор и хранение данных и отправят эти данные на Zabbix сервер, когда сервер запустится и заработает. Любые оповещения о проблемах в процессе простоя Zabbix сервера будут сгенерированы только после того как обновленный сервер запустится.

Известно, что имеется возможность запустить обновленный сервер с ещё не обновленными прокси, которые будут отправлять данные на новый сервер (хотя прокси и не смогут обновлять свою конфигурацию). Однако, такой подход не рекомендуется и не поддерживается Zabbix, его выбор исключительно на ваш страх и риск.

Обратите внимание, что при наличии прокси с SQLite базой данных, данные истории этих прокси будут потеряны до обновления, так как обновление файла базы данных SQLite не поддерживается и этот файл необходимо удалить вручную. Когда прокси запускается в первый раз и будет отсутствовать файл базы данных SQLite, прокси создаст его автоматически.

В зависимости от размера базы данных обновление базы данных до версии 4.0 может занять продолжительное время.

Перед выполнением обновления убедитесь что прочитали все соответствующие **заметки по обновлению!**

Доступны следующие заметки по обновлению:

Обновление с	Прочтайте заметки по обновлению полностью	Важные заметки/изменения между версиями
3.4.x	Для 4.0	Библиотеки 'libpthread' и 'zlib' теперь обязательны; Поддержка протокола в виде простого текста убрана и заголовок обязательен; Zabbix агенты версий Pre-1.4 более не поддерживаются; Параметр Server в конфигурации пассивного прокси теперь обязательен
3.2.x	Также для 3.4	Поддержка SQLite в виде основной базы данных убрана для Zabbix сервера/веб-интерфейса; Поддерживается Perl совместимые регулярные выражения (PCRE) вместо POSIX расширенных; Библиотеки 'libpcre' and 'libevent' обязательны для Zabbix сервера; Добавлены проверки кода выхода для пользовательских параметров, удаленных команд и элементов данных system.run[] без 'nowait' флага, а также для выполняемых скриптов Zabbix сервером; Zabbix Java gateway необходимо обновить для поддержки новых функций
3.0.x	Также для 3.2	Обновление базы данных может быть медленным, в зависимости от размеров таблиц истории
2.4.x	Также для 3.0	Минимально требуемая версия PHP повышенна с 5.3.0 до 5.4.0 Необходимо указывать параметрLogFile агента
2.2.x	Также для 2.4	Удалён распределенный мониторинг на основе нод
2.0.x	Также для 2.2	Минимально требуемая версия PHP повышенна с 5.1.6 до 5.3.0; Для правильной работы сервера требуется регистрозависимая база данных MySQL; Расширение 'mysqli' PHP требуется вместо 'mysql'

Вы возможно захотите также проверить [требования](#) для 4.0.

Возможно удобно запустить две параллельные SSH сессии на время обновления, выполняя шаги обновления в одной сессии и наблюдая за файлами журналов сервера/прокси в другой. Например, при выполнении `tail -f zabbix_server.log` или `tail -f zabbix_proxy.log` во второй SSH сессии будут отображаться последние записи из файла журнала и возможные ошибки в режиме реального времени. Такой подход может быть критичным на продуктивных серверах.

Процесс обновления сервера

1 Остановите Zabbix сервер

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

2 Сделайте архивную копию существующей базы данных Zabbix

Этот шаг очень важен. Убедитесь, что у вас есть архивная копия вашей базы данных. Это поможет, если процедура обновления закончится неудачно (отсутствие свободного места на диске, выключение питания, любая неожиданная проблема).

3 Сделайте копию файлов конфигурации, PHP файлов и приложений Zabbix

Сделайте резервные копии приложений Zabbix, файлов конфигурации и папки с файлами PHP.

4 Установите новые исполняемые файлы сервера

Используйте эти [инструкции](#) для компиляции Zabbix сервера из исходных кодов.

5 Проверьте параметры конфигурации сервера

Смотрите заметки по обновлению для получения детальной информации об [обязательных изменениях](#).

Для ознакомления с новыми необязательными параметрами, пожалуйста,смотрите раздел [Что нового](#).

6 Запустите новые бинарные файлы Zabbix

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

Процесс Zabbix сервера после запуска автоматически обновит схему базы данных. При запуске Zabbix сервер сообщает текущую (обязательную и опциональную) и требуемую версии базы данных. Если текущая обязательная версия старше чем требуемая версия, 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 записывается сообщение “database upgrade fully completed”. Если какой-либо из патчей обновления будет ошибочным, 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 начала запуска процесса 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 Установите новый веб-интерфейс Zabbix

Минимально требуемая версия PHP 5.4.0. Обновите, если требуется, и следуйте [инструкции по установке](#).

8 Очистите cookies и кэш в веб-браузере

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

Процесс обновления прокси

1 Остановите Zabbix прокси

Остановите Zabbix прокси.

2 Сделайте копию файлов конфигурации и бинарных файлов Zabbix прокси

Сделайте резервные копии бинарных файла Zabbix прокси и файлов конфигурации.

3 Установите новые исполняемые файлы прокси

Используйте эти [инструкции](#) для компиляции Zabbix прокси из исходных кодов.

4 Проверьте параметры конфигурации прокси

В этой версии обязательные изменения в [параметрах](#) прокси отсутствуют. Для ознакомления с новыми необязательными параметрами, пожалуйста,смотрите раздел [Что нового](#).

5 Запустите новый Zabbix прокси

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

Процесс Zabbix прокси после запуска автоматически обновит схему базы данных. Обновление базы данных происходит аналогично тому, как при запуске [Zabbix сервера](#).

Процесс обновления агента

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

1 Остановите Zabbix агент

Остановите Zabbix агент.

2 Сделайте копию файлов конфигурации и бинарных файлов Zabbix агента

Сделайте резервные копии бинарного файла Zabbix агента и файла конфигурации.

3 Установите новые исполняемые файлы агента

Используйте эти [инструкции](#) для компиляции Zabbix агента из исходных кодов.

Дополнительно, вы можете загрузить уже скомпилированные Zabbix агенты со [страницы загрузки Zabbix](#).

4 Проверьте параметры конфигурации агента

В этой версии обязательные изменения в [параметрах](#) агента отсутствуют.

5 Запустите новый Zabbix агент

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

Обновление между минорными версиями

При обновлении между минорными версиями 4.0.x (например, с 4.0.1 на 4.0.3) необходимо выполнить те же действия для сервера/прокси/агента, как и при обновлении между мажорными версиями. Единственное различие заключается в том, что при обновлении между минорными версиями никаких изменений в базе данных не производится.

4.7 Известные проблемы

Глобальная корреляция событий

События могут быть некорректно скоррелированы, если промежуток времени между первым и вторым событиями очень мал, к примеру пол секунды или меньше.

IPMI проверки

IPMI проверки не будут работать со стандартным пакетом библиотеки OpenIPMI на Debian до 9 (stretch) и Ubuntu до 16.04 (xenial). Чтобы исправить проблему, пересоберите OpenIPMI библиотеку с включенным OpenSSL, как обсуждалось в [ZBX-6139](#) [en].

SSH проверки

Некоторые Linux дистрибутивы такие как Debian, Ubuntu не поддерживают шифрованные приватные ключи (с ключевой фразой) при установке библиотеки libssh2 из пакетов.

Для получения более подробных сведенийсмотрите [ZBX-4850](#).

ODBC проверки

Zabbix сервер или прокси, которые используют MySQL в качестве своей базы данных могут или не могут корректно работать с библиотекой MySQL ODBC в связи с [общей проблемой](#). Пожалуйста, просмотрите [ZBX-7665](#) для получения более подробной информации и возможных способах решения проблемы.

XML данные запрашиваемые с Microsoft SQL сервера могут быть обрезаны до 2033 по причине [проблемы](#) в Microsoft.

HTTPS проверки

Веб-сценарии и HTTP агент использующие https протокол, Zabbix агент проверки `net.tcp.service[https...]` и `net.tcp.service.perf[https...]` могут завершиться неудачей, если целевой сервер настроен на запрет протокола TLS v1.0 или ниже. Пожалуйста, просмотрите [ZBX-9879](#) для получения более подробной информации и возможных способах решения проблемы.

Веб-мониторинг и HTTP агент

Zabbix сервер имеет утечку памяти на CentOS 6, CentOS 7 и, возможно, на других похожих дистрибутивах Linux по причине [ошибки в библиотеке](#) [en] при включенной опции "Проверка SSL узла" в веб-сценариях или HTTP агенте. Пожалуйста, просмотрите [ZBX-10486](#) [en] для получения более подробной информации и о доступных методах обхода.

Простые проверки

Имеется проблема в **fping** утилите в версиях до v3.10 выпуска 2.1.2, которая приводит в дубликатам ответных echo пакетов. Такое поведение может вызвать неожиданные результаты в `icmpping`, `icmppingloss`, `icmppingsec` элементах данных. Рекомендуется использовать последнюю версию **fping**. Пожалуйста,смотрите [ZBX-11726](#) [en] для получения более детальных сведений.

SNMP проверки

Если используется OpenBSD операционная система, проблема использования памяти после освобождения памяти в Net-SNMP библиотеке вплоть до 5.7.3 версии может привести к остановке Zabbix сервера, если SourceIP параметр указан в файле конфигурации Zabbix сервера. Как вариант решения, пожалуйста, не задавайте SourceIP параметр. Эта проблема также применима и к Linux, но она не приводит к остановке работы Zabbix сервера. К пакету net-snmp применен локальный патч на OpenBSD и будет выпущен с OpenBSD 6.3.

Проблема совместимости с PHP 7.0

Было замечено, что с PHP 7.0 импорт шаблона с триггерами от веб-мониторинга может провалиться по причине некорректно добавленных двойных кавычек к элементам данных веб-мониторинга в выражениях триггеров. Проблема исчезает с обновлением PHP до 7.1.

Графики

Результатом изменения на Летнее время (DST) является нарушение отображения подписей к оси X (дублирование данных, пропущенная дата и так далее).

Мониторинг файлов журналов

Элементы данных `log[]` и `logrt[]` многократно перечитывают файл журнала с самого начала, если файловая система заполнена на 100% и файл журнала продолжает заполняться (смотрите [ZBX-10884](#) для получения более подробной информации).

Медленные запросы MySQL

Zabbix сервер генерирует медленные SELECT запросы в случае несуществующих значений у элементов данных. Это обусловлено известной проблемой в MySQL версий 5.6/5.7. Решением такой проблемы является отключение параметра `index_condition_pushdown` оптимизатора в MySQL. Расширенное обсуждение смотрите здесь [ZBX-10652](#).

API

Параметр **output** работает некорректно с методом `history.get`.

API login

Может быть создано большое количество открытых сессий пользователю при использовании пользовательских скриптов с `методом user.login` без последующего `user.logout`.

Проблема с IPv6 адресами в SNMPv3 трапах

По причине проблем в net-snmp, IPv6 адреса могут некорректно отображаться при использовании SNMPv3 в SNMP трапах. Для получения более детальных сведений и возможных путях обхода смотрите [ZBX-14541](#) [en].

4.8 Изменения в шаблонах

На этой странице перечислены все изменения шаблонов, которые поставляются с Zabbix.

Предлагается изменить эти шаблоны на существующих инсталляциях - в зависимости от изменений, это можно сделать либо импортированием последней версии, либо выполнением изменений вручную.

5. Быстрый старт

5.1 Вход и настройка пользователя

Обзор

В этом разделе вы узнаете как войти в Zabbix и настроить пользователя системы.

Вход

Username

Password

Remember me for 30 days

Sign in

or sign in as guest

Это экран “Приветствия” в Zabbix. Введите имя пользователя **Admin** с паролем **zabbix** для входа под [Супер-Администратором Zabbix](#).

Когда вы войдете, вы увидите 'Подключен как Admin' в нижнем правом углу страницы. Доступ к меню *Настройка* и *Администрирование* будет предоставлен.

Защита от атак перебором

В случае пяти последовательных неудачных попыток входа, Zabbix интерфейс будет принудительно делать паузу в 30 секунд для предотвращения атак методом перебора пароля и атак по словарю.

IP адрес, с которого были осуществлены неудачные попытки, будет отображен после успешного входа в систему.

Добавление пользователя

Для просмотра информации о пользователях перейдите в *Администрирование* → *Пользователи*.

Users							User group	All	Create user	
	ALIAS	NAME	SURNAME	USER TYPE	GROUPS	IS ONLINE?	LOGIN	FRONTEND ACCESS	DEBUG MODE	STATUS
<input type="checkbox"/>	Admin	Zabbix	Administrator	Zabbix Super Admin	Zabbix administrators	Yes (2015-08-05 17:25:44)	Ok	System default	Disabled	Enabled
<input type="checkbox"/>	guest			Zabbix User	Guests	Yes (2015-08-05 17:16:38)	Ok	System default	Disabled	Enabled

Изначально в Zabbix только два предустановленных пользователя.

- Пользователь 'Admin' это суперпользователь Zabbix, который имеет все привилегии.
- Пользователь 'Guest' это специальный пользователь по умолчанию. Если пользователь не вошел в систему, тогда он получит доступ с привилегиями пользователя “guest”. По умолчанию, “guest” не имеет разрешений на объекты Zabbix.

Для добавления нового пользователя нажмите *Создать пользователя*.

В диалоге создания пользователя убедитесь, что пользователь принадлежит хотя бы одной [группе пользователей](#), например 'Zabbix администраторы'.

Users

User Media Permissions

* Alias

Name

Surname

* Groups

* Password

* Password (once again)

Language

Theme

Auto-login

Auto-logout (min 90 seconds)

* Refresh (in seconds)

* Rows per page

URL (after login)

По умолчанию у новых пользователей нет добавленных способов оповещения (методов отправки уведомлений). Для их создания перейдите на закладку 'Способы оповещения' и нажмите *Добавить*.

Media

Type Email

* Send to user@domain.tld Remove

Add

* When active 1-7,00:00-24:00

Use if severity Not classified
 Information
 Warning
 Average
 High
 Disaster

Enabled

Add Cancel

В этом всплывающем окне введите для пользователя его e-mail адрес.

Вы можете указать период времени, когда этот способ будет активен (смотрите страницу [Спецификаций периодов времени](#) для получения сведений о формате), по умолчанию способ активен всегда. Вы также можете настроить [важность триггера](#), для которого способ оповещения будет использоваться, но сейчас оставьте все включенным.

Нажмите *Добавить*, затем *Сохранить* в свойствах пользователя. Новый пользователь появится в списке пользователей.

Users									User group	All	Create user
	ALIAS	NAME	SURNAME	USER TYPE	GROUPS	IS ONLINE?	LOGIN	FRONTEND ACCESS	DEBUG MODE	STATUS	
<input type="checkbox"/>	Admin	Zabbix	Administrator	Zabbix Super Admin	Zabbix administrators	Yes (2015-11-05 07:26:26)	Ok	System default	Disabled	Enabled	
<input type="checkbox"/>	guest			Zabbix User	Guests	Yes (2015-11-05 07:25:22)	Ok	System default	Disabled	Enabled	
<input type="checkbox"/>	user	New	User	Zabbix User	Zabbix administrators	No	Ok	System default	Disabled	Enabled	

Добавление прав доступа

По умолчанию, новый пользователь не имеет прав. Для предоставления ему прав нажмите на имя группы в колонке *Группы* (в данном случае - 'Zabbix administrators'). В этом диалоге свойств группы перейдите на закладку *Права*.

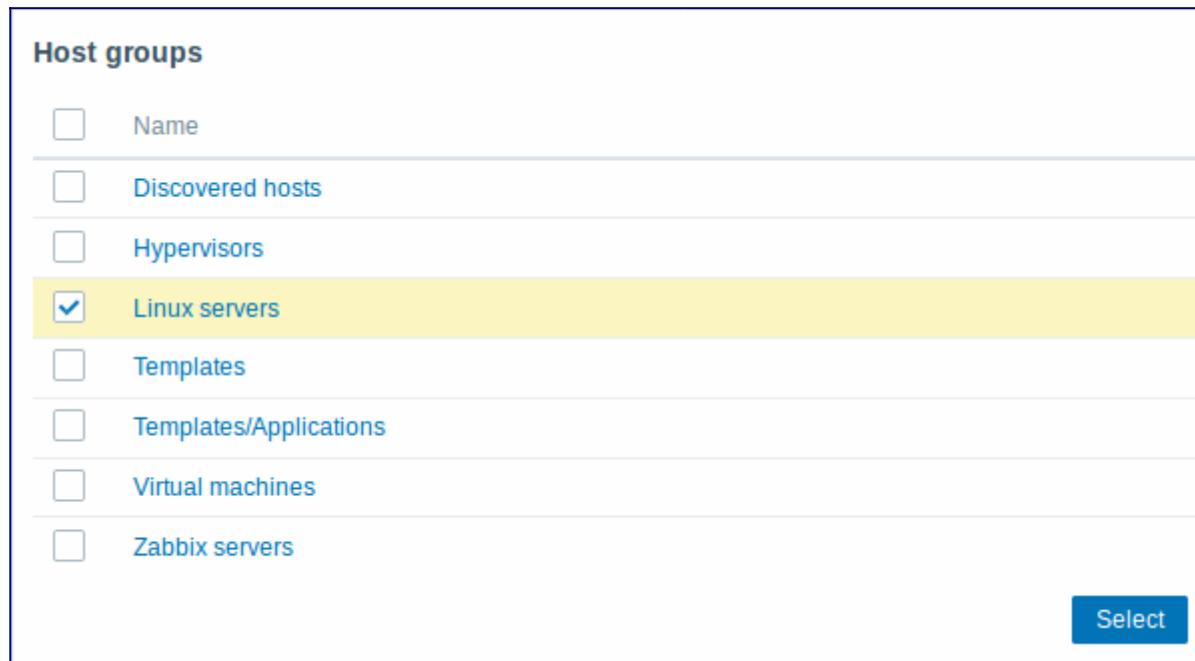
User groups

User group Permissions

Permissions	Host group	Permissions
*		None

Add

Этот пользователь должен иметь доступ только на чтение группы *Linux servers*, поэтому нажмите *Выбрать* сразу после поля выбора имени групп пользователей.



В этом всплывающем окне отметьте 'Linux servers' и затем нажмите *Выбрать*. *Linux servers* должны отображаться в поле выбора. Нажмите на кнопку 'Чтение', чтобы задать уровень прав доступа и затем *Добавить* для добавления группы в список прав доступа. В диалоге свойств группы пользователей, нажмите *Обновить*.

В Zabbix, права доступа к узлам сети назначаются через [группы пользователей](#), а не индивидуально пользователям.

Готово! Теперь вы можете попробовать войти под созданным пользователем.

5.2 Новый узел сети

Обзор

В этом разделе вы узнаете как настроить новый узел сети.

Узел сети в Zabbix - это объект сети (физический, виртуальный), который вы хотите наблюдать. Определение того, что может быть “узлом сети” в Zabbix, очень гибкое. Это может быть физический сервер, сетевой коммутатор, виртуальная машина или какое-нибудь приложение.

Добавление узла сети

Информация о настроенных узлах сети в Zabbix доступна в *Настройка* → *Узлы сети*. Уже существует один предустановленный узел сети, называемый 'Zabbix server', но мы хотим узнать как добавить еще один узел сети.

Для добавления нового узла сети нажмите *Создать*. Это действие покажет нам диалог настройки узла сети.

Hosts

Host Templates IPMI Macros Host inventory Encryption

* Host name

Visible name

* Groups
type here to search

* At least one interface must exist.

Agent interfaces

	IP address	DNS name	Connect to
⋮	127.0.0.1		<input type="button" value="IP"/> <input type="button" value="DNS"/>

SNMP interfaces

JMX interfaces

IPMI interfaces

Description

Monitored by proxy

Enabled

Все обязательные поля ввода отмечены красной звёздочкой.

Как минимум введите:

Имя узла сети

- Введите имя узла сети. Допускаются число-буквенные символы, пробелы, точки, тире и подчёркивания.

Группы

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



Все права доступа назначаются на группы узлов сети, не индивидуально узлам сети.
Поэтому узел сети должен принадлежать хотя бы одной группе.

IP адрес

- Введите IP адрес узла сети. Обратите внимание, Zabbix агенту нужно указать IP адрес Zabbix сервера в файле конфигурации в параметре 'Server'.

Остальные опции подойдут нам на данный момент по умолчанию.

Когда закончите, нажмите на *Добавить*. Ваш новый узел сети должен быть виден в списке узлов сети.



Если иконка *ZBX* в колонке *Доступность* красная, то это значит что имеется какая то ошибка со связью - поместите курсор мышки над этой иконкой и вы увидите всплывающую подсказку о ошибке. Если иконка серая, значит состояние пока не обновилось. Проверьте запущен ли Zabbix сервер и попробуйте обновить страницу немного позже.

5.3 Новый элемент данных

Обзор

В этом разделе вы узнаете как настроить элемент данных.

Элементы данных лежат в основе сбора данных в Zabbix. Без элементов данных, нет данных - потому что только элемент данных определяет одну метрику или какие данные собираются с узла сети.

Добавление элемента данных

Все элементы данных группируются вокруг узлов сети. Именно поэтому для настройки примера элемента данных мы перейдем в *Настройки* → *Узлы сети* и найдем 'Новый узел сети', который мы создали.

Ссылка *Элементы данных* в строке с 'Новым узлом сети' должна отображать количество равное '0'. Нажмите на эту ссылку, и затем нажмите на *Создать элемент данных*. Это покажет нам диалог создания элемента данных.

Item Preprocessing

* Name	CPU Load																
Type	Zabbix agent																
* Key	system.cpu.load																
* Host interface	127.0.0.1 : 10050																
Type of information	Numeric (float)																
Units																	
* Update interval	30s																
Custom intervals	<table border="1"> <thead> <tr> <th>Type</th> <th>Interval</th> <th>Period</th> <th>Action</th> </tr> </thead> <tbody> <tr> <td>Flexible</td> <td>Scheduling</td> <td>50s</td> <td>1-7.00:00-24:00</td> </tr> <tr> <td colspan="4">Remove</td> </tr> <tr> <td colspan="4">Add</td> </tr> </tbody> </table>	Type	Interval	Period	Action	Flexible	Scheduling	50s	1-7.00:00-24:00	Remove				Add			
Type	Interval	Period	Action														
Flexible	Scheduling	50s	1-7.00:00-24:00														
Remove																	
Add																	
* History storage period	90d																
* Trend storage period	365d																
Show value	As is show value mappings																
New application	<input type="text"/>																
Applications	-None-																
Populates host inventory field	-None-																
Description	<input type="text"/>																
Enabled	<input checked="" type="checkbox"/>																
Add Cancel																	

Все обязательные поля ввода отмечены красной звёздочкой.

Введите необходимую информацию, для нашего примера элемента данных:

Имя

- Введите значение *Загрузка CPU*. Это будет видимым именем элемента данных в списках и в других местах.

Ключ

- Вручную введите значение *system.cpu.load*. Это формальное имя элемента данных, которое идентифицирует тип информации, которая будет собираться. Этот ключ является лишь одним из [предопределенных ключей](#), которые поставляются с Zabbix агентом.

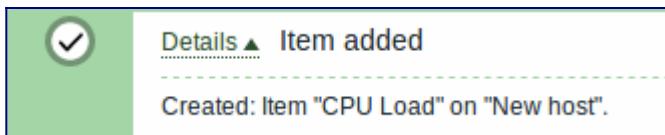
Тип информации

- Здесь выберите *Числовой (с плавающей точкой)*. Этот атрибут определяет формат ожидаемых данных.

Вы также можете захотеть уменьшить количество дней хранения [истории элемента данных](#), до 7 или 14. Это хорошая практика для уменьшения базы данных при хранении большого количества значений истории.

[Остальные опции](#) подойдут нам на данный момент по умолчанию.

Когда закончите, нажмите на *Добавить*. Новый элемент данных должен появится в списке элементов данных. Нажмите на *Детали* выше списка, чтобы просмотреть, что именно было сделано.



Просмотр данных

С элементом данных всё понятно, вам может быть интересно есть ли на самом деле сбор данных. Для этого перейдите в *Мониторинг* → *Последние данные*, выберите 'Новый узел сети' в фильтре и нажмите на *Применить*.

Затем нажмите на + напротив - **другое** - и проверьте есть ли там ваш элемент данных и отображаются ли данные.

HOST	NAME	LAST CHECK	LAST VALUE	CHANGE	
New host	- other - (1 item)				
	CPU Load	2015-08-08 16:00:49	2.24	-0.26	Graph

С учетом сказанного, получение первых данных может занять до 60 секунд. Это время, по умолчанию, как часто сервер читает изменения конфигурации и забирает новые элементы данных для обработки.

Если вы не видите значения в колонке 'Изменен', то возможно к настоящему времени было получено только одно значение. Подождите 30 секунд пока придет другое значение.

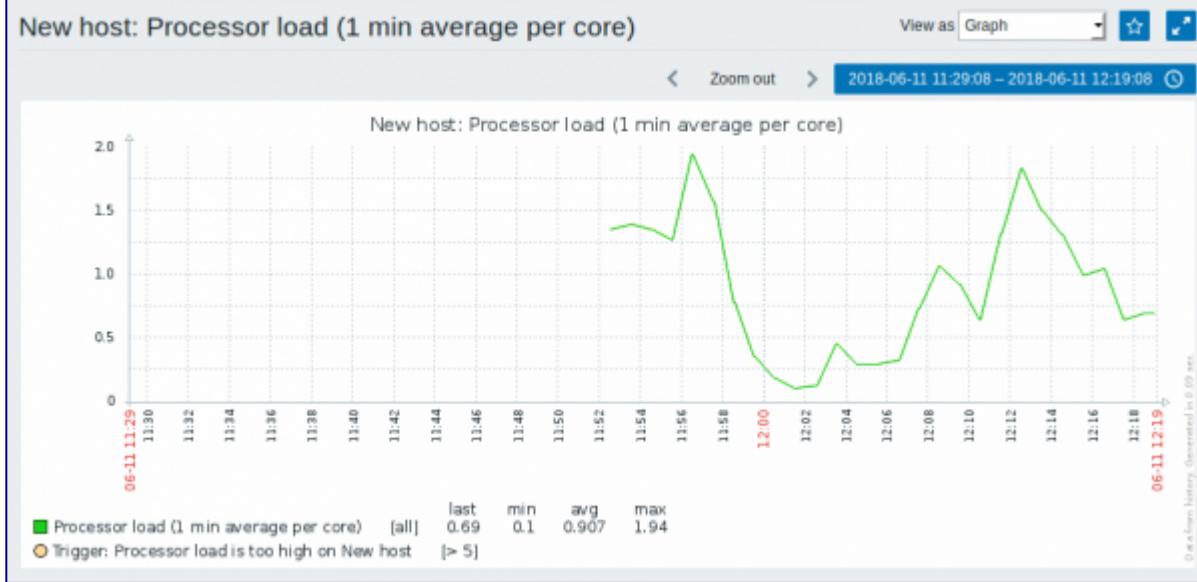
Если вы не видите информации о элементе данных как в снимке экрана, убедитесь что:

- вы указали поля 'Ключ' и 'Тип информации' в элементе данных так же как на снимке экрана
- агент и сервер запущены
- состояние узла сети 'Активирован' и его иконка доступности зеленая
- выберите узел сети в списке узлов сети, элемент данных активен

Графики

При наличии работающего элемента данных какое-то время, возможно настало время для того чтобы увидеть что-то визуальное. [Простые графики](#) доступны для любого наблюдаемого числового элемента данных без каких либо дополнительных настроек. Эти графики создаются в режиме реального времени.

Для просмотра графика, перейдите к *Мониторинг* → *Последние данные* и нажмите на ссылку 'График' у этого элемента данных.



5.4 Новый триггер

Обзор

В этом разделе вы узнаете как настроить триггер.

Элементы данных только собирают данные. Для автоматической оценки приходящих данных нам нужно задать триггеры. Триггер содержит выражение, которое определяет порог являющийся приемлемым уровнем для данных.

Если этот уровень будет превышать пришедшие данные, триггер будет “загораться” или перейдет в состояние 'Проблема' - давая понять, что что-то произошло и может потребовать внимания. Если уровень станет снова приемлемым, то триггер вернется в состояние 'OK'.

Добавление триггера

Для настройки триггера для нашего элемента данных перейдите в *Настройка → Узлы сети*, найдите 'Новый узел сети' и далее нажмите на *Триггеры* и затем на *Создать триггер*. Нам будет отображен диалог добавления триггера.

Trigger [Dependencies](#)

* Name	CPU load too high on "New host" for 3 minutes				
Severity	Not classified	Information	Warning	Average	High
* Expression	{New host:system.cpu.load.avg(3m)}>2				
Expression constructor					
OK event generation	Expression	Recovery expression	None		
PROBLEM event generation mode	Single	Multiple			
OK event closes	All problems	All problems if tag values match			
Tags	<input type="text" value="tag"/> <input type="text" value="value"/> Add				
Allow manual close	<input type="checkbox"/>				
URL					
Description					
Enabled	<input checked="" type="checkbox"/>				
Add Cancel					

Все обязательные поля ввода отмечены красной звёздочкой.

Введите здесь необходимую информацию для нашего триггера:

Имя

- Введите значение *Загрузка CPU слишком высокая у 'New host' в течении 3 минут.* Это будет имя триггера, которое будет отображаться в списках и в других местах.

Выражение

- Введите: {New host:system.cpu.load.avg(3m)}>2

Это выражение триггера. Убедитесь, что выражение введено верно, вплоть до последнего символа. Здесь ключ элемента данных (system.cpu.load) используется для ссылки на элемент данных. По простому данное выражение говорит, что порог проблемы превышается, когда значение средней загрузки CPU в течении 3 минут превышает 2. Вы можете узнать больше о [синтаксисе выражений триггеров](#).

Когда завершите, нажмите на *Сохранить*. Этот новый триггер должен появиться в списке триггеров.

Просмотр состояния триггера

После добавления триггера, вы возможно заинтересуетесь узнать его состояние.

Если загрузка CPU превысит порог, который вы указали в триггере, проблема отобразится в *Мониторинг → Проблемы*.

Time	Severity	Recovery time	Status	Info	Host	Problem	Duration
09:48:39	Not classified		PROBLEM	New host	CPU load too high on New host for 3 minutes		1m 23s

Мигание указывает на недавнее изменение состояние триггера, которое имело место за последние 30 минут.

5.5 Получение оповещения о проблеме

Обзор

В этом разделе вы узнаете как настроить в Zabbix уведомления из диалога оповещений.

При наличии элементов данных, которые собирают данные, и триггеров, переходящих в состояние “Проблема” при проблемных ситуациях, было бы полезно иметь какой-нибудь механизм оповещений, который будет уведомлять нас о важных событиях, тогда когда мы не можем смотреть напрямую веб-интерфейс Zabbix.

Это то, что делают оповещения. E-mail является наиболее популярным способом для отправки уведомлений о проблемах, мы узнаем, как настроить уведомления через e-mail.

Настройки E-mail

Изначально в Zabbix имеется несколько предустановленных [способов доставки](#) оповещений. [E-mail](#) один из них.

Для конфигурирования настроек e-mail, перейдите в *Администрирование → Способы оповещений* и нажмите на *Email* в списке предустановленных способов оповещений.

Media types					Create media type
	NAME	TYPE	STATUS	USED IN ACTIONS	DETAILS
<input type="checkbox"/>	Email	Email	Enabled		SMTP server: "mail.company.com", SMTP helo: "company.com", SMTP email: "zabbix@company.com"
<input type="checkbox"/>	Jabber	Jabber	Enabled		Jabber identifier: "jabber@company.com"
<input type="checkbox"/>	SMS	SMS	Enabled		GSM modem: "devilityS0"

Отобразится диалог определения настроек e-mail.

Media types

Media type Options

* Name	Email
Type	Email
* SMTP server	mail.zabbix.com
SMTP server port	25
* SMTP helo	zabbix.com
* SMTP email	zabbix@zabbix.com
Connection security	None STARTTLS SSL/TLS
Authentication	None Username and password
Enabled	<input checked="" type="checkbox"/>
<button>Update</button> <button>Clone</button> <button>Delete</button> <button>Cancel</button>	

Все обязательные поля ввода отмечены красной звёздочкой.

Установите значения сервера SMTP, SMTP helo и SMTP e-mail в зависимости от вашей среды.

'SMTP email' будет использоваться как адрес 'От' при отправке оповещений от Zabbix.

Нажмите *Обновить*, когда будет готово.

Теперь у вас есть настроенный 'Email' как рабочий способ оповещения. Способ оповещения должен быть связан с пользователями с указанными адресами доставки (как мы это делали, когда [настраивали нового пользователя](#)), в противном случае он не будет использоваться.

Новое действие

Доставка оповещений является одной из задач [действий](#), которые делаются в Zabbix. Поэтому, для настройки оповещений перейдите в *Настройка* → *Действия* и нажмите на *Создать действие*.

Actions

Action Operations Recovery operations Update operations

* Name	Test action		
Conditions	Label	Name	Action
New condition	Trigger name	like	
	Add		
Enabled	<input checked="" type="checkbox"/>		
* At least one operation, recovery operation or update operation must exist.			
<button>Add</button> <button>Cancel</button>			

Все обязательные поля ввода отмечены красной звёздочкой.

В этом диалоге введите имя действия.

В более простом случае, если мы не будем добавлять более конкретных условий, действием будет обработан любой триггер изменивший состояние с 'Ok' на 'Проблема'.

Мы еще должны определить какое действие нужно делать - это настраивается на вкладке *Операции*. Нажмите во вкладке Операции на *Новое*, появится диалог новой операции.

Actions

Action Operations Recovery operations Update operations

* Default operation step duration

Default subject

Default message

Pause operations for suppressed problems

Operations Steps Details Start in Duration Action

1	Send message to users: user (New user) via Email	Immediately	Default	Edit Remove
---	--	-------------	---------	---

Operation details

Steps - (0 - infinitely)

Step duration (0 - use action default)

Operation type

* At least one user or user group must be selected.

Send to User groups [Add](#)

Send to Users [Remove](#) [Add](#)

Send only to

Default message

Conditions

[New](#)

[Update](#) [Cancel](#)

* At least one operation, recovery operation or update operation must exist.

[Add](#) [Cancel](#)

Все обязательные поля ввода отмечены красной звёздочкой.

Здесь, нажмите на *Добавить* в блоке *Отправлять пользователям* и выберите пользователя ('user'), которого мы добавили. Выберите 'Email' значением в *Отправлять только через*. Когда закончите с этим, нажмите на *Добавить*.

Макросы (или переменные) {TRIGGER.STATUS} и {TRIGGER.NAME}, которые видно в полях *Тема по умолчанию* и *Сообщение по умолчанию*, будут заменены актуальными значениями состояния триггера и имени триггера.

Это всё что требуется для настройки простого действия, поэтому нажмите на *Сохранить* в диалоге действия.

Получение оповещения

Теперь, с настроенной отправкой оповещений вам на самом деле может быть интересно получить одно оповещение. Чтобы помочь с этим, мы попробовать увеличить нагрузку на наш узел сети - так, чтобы наш [триггер](#) “зажегся” и мы получили бы оповещение о проблеме.

Откройте консоль на вашем узле сети и выполните:

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

Вы можете запустить один или несколько [этих процессов](#).

Теперь перейдите в *Мониторинг* → *Последние данные* и посмотрите как увеличиваются значения 'Загрузка CPU'. Вспомните, для нашего триггера “поджигание” наступает при значении 'Загрузка CPU' больше '2' в течении 3 минут. Как только это произойдет:

- в *Мониторинг* → *Проблемы* вы должны увидеть триггер с мигающим состоянием 'Проблема'
- вы должны получить оповещение о проблеме на ваш e-mail

Если оповещения не работают:

- Проверьте еще раз, что настройки e-mail и действия были сконфигурированы верно
- Убедитесь, что созданный вами пользователь имеет по крайней мере права на чтение узла сети который генерирует событие, как отмечается в шаге [Добавление пользователя](#). Пользователь, входящий в группу 'Network administrators' должен иметь, по крайней мере, доступ на чтение группы узлов сети 'Linux servers', к которой принадлежит наш узел сети.
- Кроме того, вы можете проверить журнал действий перейдя в *Отчеты* → *Журнал действий*.

5.6 Новый шаблон

Обзор

В этом разделе вы узнаете как настроить шаблон.

Ранее мы узнали как настроить элемент данных, триггер и как получать оповещения о проблемах с этим узлом сети.

Хотя все эти шаги дают большую гибкость как таковые, они могут выглядеть как множество шагов, если потребуется, скажем, добавление их к тысячи узлов сети. Автоматизация процесса была бы кстати.

В этом нам придут на помощь шаблоны. Шаблоны позволяют группировать полезные элементы данных, триггеры и другие сущности, чтобы они могли быть использованы снова и снова простым применением шаблона к узлам сети за один шаг.

Когда шаблон соединен с узлом сети, он унаследует все сущности шаблона. Таким образом, обычно предопределенная куча проверок может быть применена очень быстро.

Добавление шаблона

Для начала работы с шаблонами мы первым делом должны его создать. Чтобы это сделать, перейдите в *Настройка* → *Шаблоны* и нажмите на *Создать шаблон*. Так мы увидим диалог настройки шаблона.

Templates

Template Linked templates Macros

* Template name: New template

Visible name:

* Groups: Templates

Description:

Add Cancel

Все обязательные поля ввода отмечены красной звёздочкой.

Введите здесь необходимые параметры:

Имя шаблона

- Введите имя шаблона. Допускаются число-буквенные символы, пробелы и подчёркивания.

Группы

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

Когда завершите, нажмите на *Добавить*. Ваш новый шаблон должен появиться в списке шаблонов.

Templates Group: all Create template Import

TEMPLATES APPLICATIONS ITEMS TRIGGERS GRAPHS SCREENS DISCOVERY WEB LINKED TEMPLATES LINKED TO

New template Applications Items Triggers Graphs Screens Discovery Web

Как вы можете видеть, шаблон есть, но он ничего не содержит в себе - нет элементов данных, триггеров и других сущностей.

Добавление элемента данных в шаблон

Для добавления элемента данных в шаблон, перейдите в список элементов данных 'Новый узел сети' узла сети. В *Настройки* → *Узлы сети* нажмите на *Элементы данных* рядом с 'Новый узел сети'.

Затем:

- выделите галочкой элемент данных 'Загрузка CPU' из списка
- выберите *Копировать* ниже списка
- выберите шаблон и скопируйте элемент данных туда

* Target type: Templates

Group: Templates

* Target: Template OS Linux

Copy Cancel

Все обязательные поля ввода отмечены красной звёздочкой.

- нажмите на *Копировать*

Если вы сейчас перейдете в *Настройка* → *Шаблоны*, вы должны увидеть 'Новый шаблон' с одним элементом данных.

Сейчас мы остановимся на одном элементе данных, но вы так же можете добавлять любые другие элементы данных, триггеры и другие сущности в шаблон, пока он не будет заполнен набором объектов для конкретных целей (мониторинг ОС, мониторинг одного приложения).

Соединение шаблона с узлом сети

Когда шаблон будет готов, остается только добавить его к узлу сети. Для этого перейдите в *Настройки* → *Узлы сети*, нажмите на 'Новый узел сети', откроется диалог его настроек, теперь перейдите на закладку **Шаблоны**.

Здесь, нажмите на *Добавить*, отметьте шаблон который мы создали ('Новый шаблон') и нажмите на *Выбрать*. Шаблон должен появиться в диалоге.

The screenshot shows the 'Hosts' configuration screen in Zabbix. The top navigation bar includes links for 'All hosts / New host', 'Enabled', and various monitoring protocols like ZBX, SNMP, JMX, and IPMI. Below the navigation is a tabs menu with 'Host' (selected), 'Templates' (highlighted in blue), 'IPMI', 'Macros', and 'Host inventory'. The main content area is titled 'Linked templates' and contains a table with one row: 'Name' (New template) and 'Action' (Unlink). Below this table is a section titled 'Link new templates' with a search input field and a 'Select' button. At the bottom of the screen are several action buttons: 'Update', 'Clone', 'Full clone', 'Delete', and 'Cancel'.

Нажмите *Обновить* в диалоге для сохранения изменений. Теперь шаблон добавлен к узлу сети со всеми своими объектами, которые он имеет.

Как вы уже догадались, этот способ может быть применен и к любым другим узлам сети. Любые изменения элементов данных, триггеров и остальных объектов на уровне шаблона, изменения будут распространяться и на узлы сети с которыми этот шаблон соединен.

Соединение предопределенных шаблонов с узлами сети

Как вы могли заметить, Zabbix поставляется с набором предопределенных шаблонов для различных ОС, устройств и приложений. Чтобы начать наблюдение как можно быстрее, вы можете присоединить соответствующий шаблон к узлу сети, но нужно учесть, что эти шаблоны нужно доработать для вашей среды. Некоторые проверки могут не понадобится и интервалы проверок могут быть слишком частые.

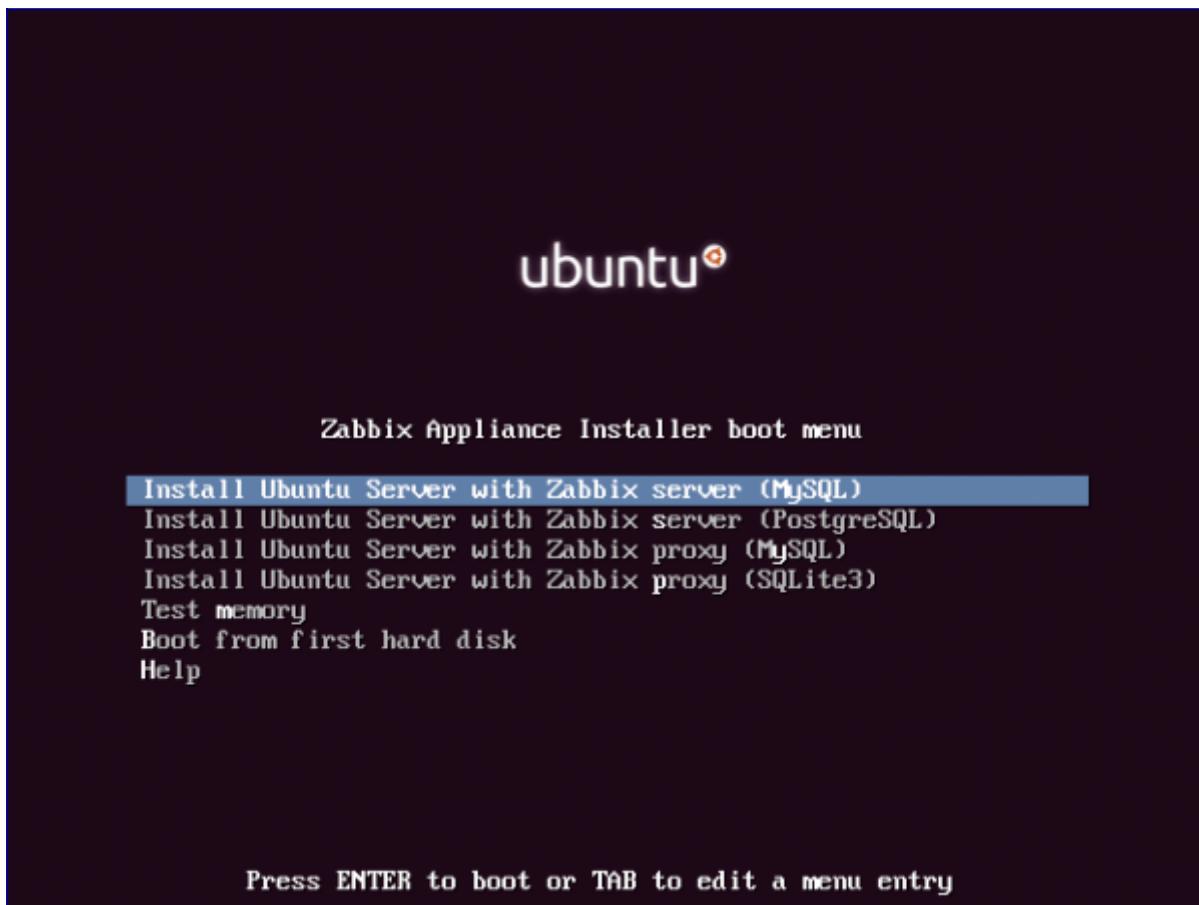
Доступна более подробная информация о [шаблонах](#).

6. Готовое решение Zabbix

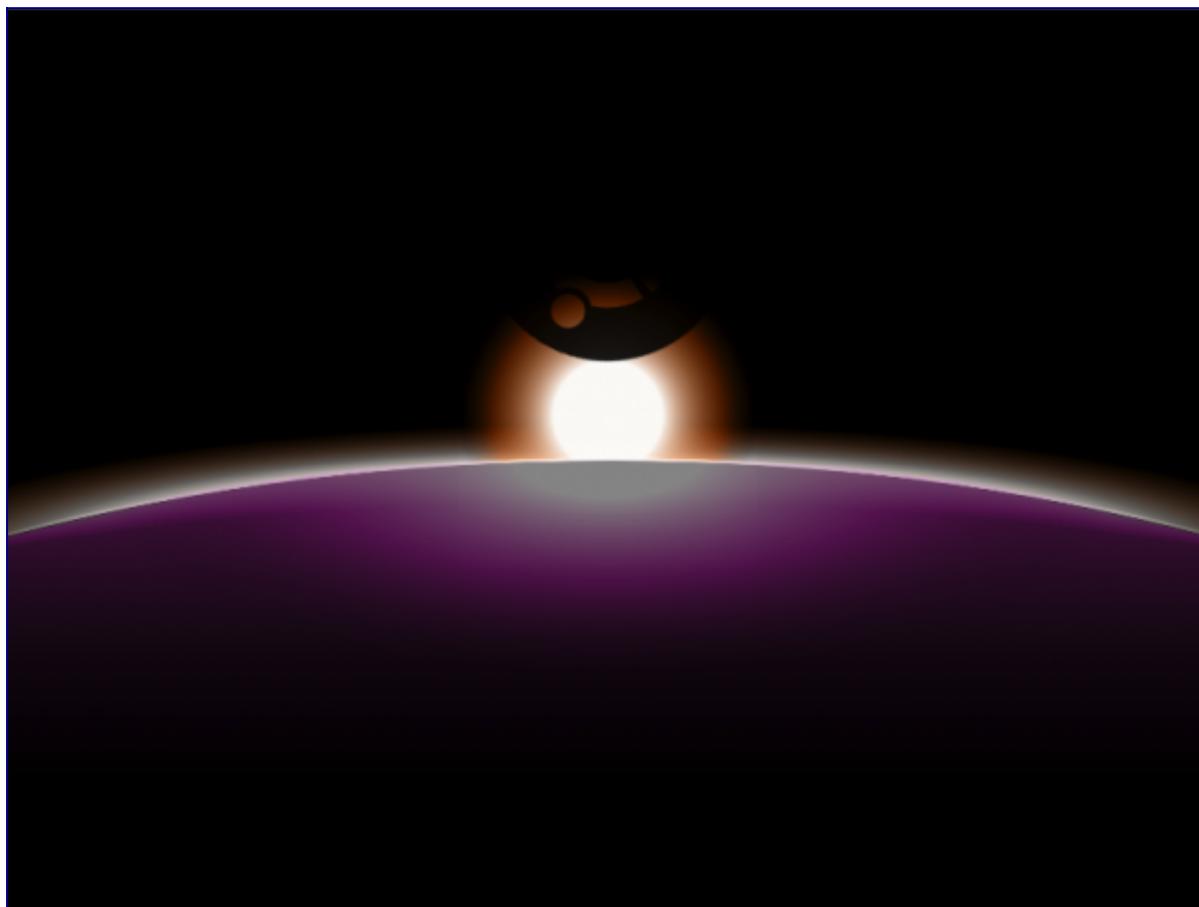
В качестве альтернативы ручной настройке или повторному использованию существующего сервера для Zabbix, пользователи могут [загрузить](#) готовое решение Zabbix или установочный образ CD готового решения Zabbix. Установочный образ CD готового решения Zabbix можно использовать для

быстрого развертывания Zabbix сервера (MySQL), Zabbix сервера (PostgreSQL), Zabbix прокси (MySQL) и Zabbix прокси (SQLite 3).

Виртуальные машины готового решения Zabbix собраны с Zabbix сервером с поддержкой MySQL. Они собраны при помощи установочного образа CD готового решения Zabbix.



Загрузочное меню установки Zabbix CD/DVD



Загрузка готового решения Zabbix

Готовое решение Zabbix и установочный CD основаны на следующих версиях Ubuntu:

Версия готового решения Zabbix Версия Ubuntu

4.0.0

16.04.3

Готовое решение Zabbix доступно в следующих форматах:

- vmdk (VMware/Virtualbox)
- OVF (Open Virtualisation Format)
- KVM
- HDD/flash image, USB stick
- Live CD/DVD
- Xen guest
- Microsoft VHD (Azure)
- Microsoft VHD (Hyper-V)

Для запуска, загрузите готовое решение и перейдите в вашем браузере по IP, который получен по DHCP: http://<ip_хоста>/zabbix

Готовое решение имеет настроенный Zabbix сервер, работающий с базой данных MySQL, также доступен и веб-интерфейс.

Готовое решение создано при помощи стандартной возможности Ubuntu/Debian, называемой Preseed файлами.

6.1 Изменения в настройках Ubuntu

Имеется несколько изменения, которые применены к базовой конфигурации Ubuntu.

6.1.1 Репозитории

Добавлен официальный [репозиторий](#) Zabbix в */etc/apt/sources.list*:

```
## Zabbix repository
deb https://repo.zabbix.com/zabbix/4.0/ubuntu xenial main
deb-src https://repo.zabbix.com/zabbix/4.0/ubuntu xenial main
```

6.1.2 Межсетевой экран

Готовое решение использует брандмауэр iptables с предустановленными правилами:

- Открыт порт SSH (22 TCP);
- Открыты порты Zabbix агента (10050 TCP) и Zabbix траппера (10051 TCP);
- Открыты порты HTTP (80 TCP) и HTTPS (443 TCP);
- Открыт порт SNMP трапов (162 UDP);
- Открыты исходящие подключения на порт DNS (53 UDP) к 8.8.8.8 и 8.8.4.4;
- Открыты исходящие подключения на порт NTP (123 UDP);
- ICMP пакеты ограничены 5 пакетами в секунду;
- Все остальные входящие подключение отбрасываются.

6.1.3 Дополнительные пакеты

Добавлены различные простые утилиты, которые могут сделать работу с Zabbix и мониторингом в целом легче:

- iptables-persistent
- mc
- htop
- snmptrapfmt
- snmp-mibs-downloader

Некоторые из этих пакетов используются Zabbix, некоторые из них установлены, чтобы помочь пользователям настраивать/управлять настройками готового решения.

6.1.4 Использование статического IP адреса

По умолчанию готовое решение получает IP адрес по DHCP. Для того, чтобы указать статический IP адрес выполните:

- Войдите под пользователем root;
- Откройте файл `/etc/network/interfaces` в вашем любимом редакторе;
- `iface eth0 inet dhcp → iface eth0 inet static`
- Измените следующие строки после `iface eth0 inet static`:
 - `address <IP адрес готового решения>`
 - `netmask <маска подсети>`
 - `gateway <ваш адрес сетевого шлюза>`
- Выполните команды `sudo ifdown eth0 && sudo ifup eth0`.

Для получения более подробной информации о других возможных опциях обратитесь к официальной [документации](#) по Ubuntu.

Для настройки DNS, добавьте записи серверов имен в `/etc/resolv.conf`, указывая по одному серверу имен в одной строке: **nameserver 192.168.1.2**.

6.1.5 Изменение часового пояса

По умолчанию готовое решение использует UTC часовой пояс для системного времени. Для изменения часового пояса, скопируйте соответствующий файл из `/usr/share/zoneinfo` в `/etc/localtime`, например:

```
cp /usr/share/zoneinfo/Europe/Riga /etc/localtime
```

6.1.6 Изменения локали

Готовое решение содержит несколько изменений локалей:

- Содержит следующие языки: `en_US.UTF-8, ru_RU.UTF-8, ja_JP.UTF-8, cs_CZ.UTF-8, ko_KR.UTF-8, it_IT.UTF-8, pt_BR.UTF-8, sk_SK.UTF-8, uk_UA.UTF-8, fr_FR.UTF-8, pl.UTF-8`;
- Локаль по умолчанию `en_US.UTF-8`.

Эти изменения требуются для поддержки мультиязычного веб-интерфейса Zabbix.

6.1.7 Другие изменения

- Сеть настроена на получение IP адреса по DHCP;
- Утилита **fping** имеет права 4710 и ей назначен владелец группы **zabbix** - `suid` и только разрешено использование группе **zabbix**;
- `ntpd` настроен на синхронизацию с пулами публичных серверов: `ntp.ubuntu.com`;
- Используется LVM том с `ext4` файловой системой.
- Добавлено “`UseDNS no`” в файл конфигурации SSH сервера `/etc/ssh/sshd_config` для того, чтобы избежать долгих ожиданий при подключении к SSH;
- Демон `snmpd` отключен в файле конфигурации `/etc/default/snmpd`.

6.2 Конфигурация Zabbix

Инсталляция готового решения Zabbix имеет следующие пароли и другие изменения конфигурации:

6.2.1 Учетные данные (имя пользователя:пароль)

Система:

- `appliance:zabbix`

Используйте “`sudo su`” команду вместе с паролем от “`appliance`” пользователя для получения привилегированных root прав.

База данных:

- root:<случайный>
- zabbix:<случайный>

Пароли к базе данных генерируются случайным образом в процессе инсталляции.

Пароль root записан в /root/.my.cnf файле, таким образом не требуется вводить пароль, находясь под аккаунтом “root”.

Веб-интерфейс Zabbix:

- Admin:zabbix

Для изменения пароля пользователя базы данных, потребуются изменения в следующих местах:

- MySQL;
- /etc/zabbix/zabbix_server.conf;
- /etc/zabbix/web/zabbix.conf.php.

6.2.2 Размещения файлов

- Файлы конфигурации расположены в **/etc/zabbix**.
- Файлы журналов (логи) Zabbix сервера, прокси и агента расположены в **/var/log/zabbix**.
- Веб-интерфейс Zabbix расположен в **/usr/share/zabbix**.
- Домашняя папка пользователя **zabbix** - **/var/lib/zabbix**.

6.2.3 Изменения в настройках Zabbix

- Имя сервера для веб-интерфейса Zabbix изменено на “Zabbix Appliance”;
- Часовой пояс веб-интерфейса изменен на Europe/Riga - родина Zabbix (этот параметр можно изменить в файле **/etc/apache2/conf-available/zabbix.conf**);

6.2.4 Сохранение конфигурации

Если вы загрузили готовое решение с Live CD или по какой-то причине не можете иметь постоянного накопителя, вы можете создать архивную копию базы данных, включая все настройки и все собранные данные.

Для создания архивной копии, выполните:

```
mysqldump zabbix | bzip2 -9 > dbdump.bz2
```

Теперь вы можете переместить файл **dbdump.bz2** на другую машину.

Для восстановления из архивной копии, переместите копию в готовое решение и выполните:

```
bzcat dbdump.bz2 | mysql zabbix
```

Убедитесь, что Zabbix сервер не запущен при выполнении восстановления.

6.3 Доступ к веб-интерфейсу

По умолчанию доступ к веб-интерфейсу разрешен отовсюду.

Доступ к веб-интерфейсу может быть получен с *http://<хост>/zabbix*.

Эти параметры могут быть изменены в **/etc/apache2/conf-available/zabbix.conf**. Вы должны перезапустить веб-сервер после изменения этого файла. Чтобы это сделать зайдите на сервер по SSH под root и выполните:

```
service apache2 restart
```

6.4 Межсетевой экран

По умолчанию, открыты только те порты, которые перечислены в изменениях. Чтобы открыть дополнительные порты, просто измените файлы “*/etc/iptables/rules.v4*” или “*/etc/iptables/rules.v6*” и перезагрузите правила межсетевого экрана:

```
service iptables-persistent reload
```

6.5 Возможности мониторинга

Zabbix сервер скомпилирован с поддержкой следующих функций:

- SNMP
- IPMI
- Веб-мониторинг
- VMware мониторинг
- Оповещения по Jabber
- Оповещения по EZ Texting
- ODBC
- SSH2
- IPv6
- SNMP трапы
- Zabbix Java Gateway

6.6 SNMP трапы

Готовое решение Zabbix использует *snmptrapfmt* для обработки SNMP трапов. Утилита настроена на получение всех трапов отовсюду.

Аутентификация не требуется. Если вы желаете включить аутентификацию, вам необходимо изменить файл */etc/snmp/snmptrapd.conf* и указать требуемые настройки аутентификации.

Все трапы записываются в */var/log/zabbix/snmptrapfmt.log* файл. Этот файл ротируется при помощи logrotate до достижения размера файла в 2ГБ.

6.7 Обновление

Пакеты готового решения можно обновлять. Чтобы это сделать, выполните:

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

6.8 Именование, скрипты запуска и другие скрипты

Представлены соответствующие скрипты запуска. Для контроля Zabbix сервера используйте следующие команды:

```
service zabbix-server status
```

Замените **server** на **agent** для демона Zabbix агента или на **proxy** для демона Zabbix прокси.

6.8.1 Увеличение доступного дискового пространства

Создайте резервную копию всех данных до выполнения любого из шагов.

Доступного дискового пространства в готовом решении может быть недостаточно. В этом случае имеется возможность расширить используемый диск. Для этого сначала расширьте блочное устройство в вашей среде виртуализации, затем выполните эти шаги.

Запустите *fdisk* для изменения размера раздела. Выполните под *root*:

```
fdisk /dev/sda
```

Эта команда запустит *fdisk* для диска *sda*. Затем, переключитесь на использование секторов и введите:
u

Не отключайте совместимость с DOS нажатием **c**. При отключении этой совместимости раздел будет поврежден.

Затем удалите существующий раздел и создайте новый с нужным размером. В большинстве случаев вы будете выбирать доступный максимум, что расширит файловую систему до размера, который вы сделали доступным для виртуального диска. Для этого введите следующую последовательность в *fdisk*:

```
d  
n  
p  
1  
(принимаем по умолчанию 63)  
(принимаем по умолчанию max)
```

Если вы хотите оставить пространство для дополнительных разделов (swap и т.д.), вы можете ввести другое значение для *last sector* (последний сектор). По завершении сохраните изменения введя:

```
w
```

После создания раздела (с нового диска или расширения существующего) создайте физический том:
pvcreate /dev/sdb1

Имя раздела */dev/sdb1* используется как пример; в вашем случае имя диска и номер раздела могут быть другими. Вы можете проверить номер раздела, выполнив *fdisk -l /dev/sdb* команду.

Проверьте только что созданный физический том:

```
pvdisplay /dev/sdb1
```

Проверьте доступные физические тома. Должно быть 2 тома *zabbix-vg* и только что созданный:
pvs

Расширьте существующую группу томов при помощи только что созданного физического тома:

```
vgextend zabbix-vg /dev/sdb1
```

Проверьте группу томов “*zabbix-vg*”:

```
vgdisplay
```

Теперь расширьте ваш логический том, заняв все свободное пространство:

```
lvextend -l +100%FREE /dev/mapper/zabbix--vg-root
```

Измените размер вашего корневого тома (можно выполнить на живой системе):

```
resize2fs /dev/mapper/zabbix--vg-root
```

Перезагрузите виртуальную машину (так как раздел, который мы изменили, в настоящее время используется). Теперь файловая система должна быть увеличена до размера раздела. Проверьте “*/dev/mapper/zabbix–vg-root*” том:

```
df -h
```

6.9 Заметки о специфичных форматах

6.9.1 Xen

Конвертация образа для XenServer

Для использования Xen образов в Citrix Xenserver вам необходимо сконвертировать образ диска. Чтобы это сделать:

- Создайте виртуальный диск, который будет по крайней мере такого же размера как и образ
- Узнайте UUID этого диска

```
xe vdi-list params=all
```

- Если имеется много дисков, их можно отфильтровать по параметру имени *name-label* (имя-метка), как назначенные при создании виртуального диска
- Импортируйте образ

```
xe vdi-import filename="image.raw" uuid=<UUID>"
```

Инструкции с *Brian Radford blog*.

6.9.2 VMWare

Образы в формате *vmdk* используются напрямую в продуктах VMWare Player, Server и Workstation. Для использования в ESX, ESXi и vSphere образы должны быть сконвертированы с использованием [конвертер VMWare](#).

6.9.3 Образ HDD/flash (raw)

```
dd if=./zabbix_appliance_4.0.0_x86_64.raw of=/dev/sdc bs=4k conv=fdatasync
```

Замените */dev/sdc* на ваше Flash/HDD устройство диска.

6.10 Известные проблемы

7. Настройка

7.1 Узлы сети и группы узлов сети

Что такое "узел сети"?

Типичные узлы сети в Zabbix - это устройства, которые необходимо мониторить (сервера, рабочие станции, коммутаторы и т.д.).

Создание узлов сети является одной из первоочередных задач мониторинга в Zabbix. Например, если вы хотите мониторить некоторые параметры на сервере “x”, сначала вы должны создать узел сети называемый, скажем, “Сервер X” и только затем вы можете смотреть в сторону добавления элементов данных для мониторинга сервера.

Узлы сети объединяются в группы узлов сети.

Перейти к [созданию и настройке узла сети](#).

7.1.1 Настройка узла сети

Обзор

Для настройки узла сети в веб-интерфейсе Zabbix, сделайте следующее:

- Перейдите в: *Настройка* → *Узлы сети*

- Нажмите на *Создать узел сети* в верхнем правом углу (или на имя узла сети для изменения существующего узла сети)
- В диалоге введите параметры для узла сети

Вы также можете использовать кнопки *Клонировать* и *Полное клонирование* в диалоге существующего узла для создания нового узла сети. Нажатие на *Клонировать* сохранит все параметры узла сети и все соединения с шаблонами (с сохранением всех объектов из этих шаблонов). *Полное клонирование* дополнительно сохранит все напрямую добавленные объекты (элементы данных, триггеры, графики и группы элементов данных).

Обратите внимание: При клонировании узла сети, он наследует все сущности из шаблонов, как они настроены изначально в шаблонах. Любые сделанные изменения в этих сущностях на уровне узла сети (такие как интервал опроса элементов данных, измененное регулярное выражение или добавленные прототипы к правилу низкоуровневого обнаружения) не будут скопированы в новый узел сети; они останутся так как есть в шаблоне.

Настройка

Вкладка **Узел сети** содержит общие атрибуты узла сети

The screenshot shows the 'Hosts' configuration dialog in Zabbix. The 'Host' tab is selected. Key fields include:

- Host name:** Zabbix server_1
- Visible name:** Zabbix server
- Groups:** Linux servers, Zabbix servers, Discovered hosts (with a search bar below)
- Agent interfaces:** IP address: 192.168.3.220, DNS name: (empty), Connect to: IP, Port: 10050. An 'Add' button is available.
- SNMP interfaces:** IP address: 127.0.0.1, DNS name: (empty), Connect to: IP, Port: 161. A checked checkbox 'Use bulk requests' is present, and an 'Add' button is available.
- JMX interfaces:** An 'Add' button is available.
- IPMI interfaces:** An 'Add' button is available.
- Description:** Added on 2018-02-07
- Monitored by proxy:** (no proxy)
- Enabled:** checked

 At the bottom are 'Add' and 'Cancel' buttons.

Все обязательные поля ввода отмечены красной звёздочкой.

Параметр	Описание
Имя узла сети	<p>Введите уникальное имя узла сети. Разрешены буквенно-цифровые символы, пробелы, точки, тире и подчёркивания.</p> <p><i>Обратите внимание:</i> При работающем Zabbix агенте на настраиваемом вами узле сети, параметр <i>Hostname</i> из файла конфигурации агента должен иметь такое же значение, как и введенное здесь имя узла сети. Имя из этого параметра необходимо для обработки активных проверок.</p>

Параметр	Описание
<i>Видимое имя</i>	Если вы укажите это имя, именно оно будет видимо в списках, картах и прочем. Этот атрибут имеет поддержку UTF-8.
<i>Группы</i>	Выберите группы узлов сети к которым будет принадлежать узел сети. Узел сети должен принадлежать по крайней мере одной группе узлов сети. Новую группу можно создать и присоединить к узлу сети, добавив несуществующее имя группы. Для узлов сети поддерживаются несколько типов интерфейсов: <i>агент</i> , <i>SNMP</i> , <i>JMX</i> и <i>IPMI</i> .
<i>Интерфейсы</i>	Чтобы добавить новый интерфейс, нажмите на <i>Добавить</i> в блоке <i>Интерфейсы</i> и введите информацию <i>IP/DNS</i> , <i>Подключаться через</i> и <i>Порт</i> . <i>Обратите внимание:</i> Интерфейсы, которые используются какими-нибудь элементами данных, не могут быть удалены и у них ссылка <i>Удалить</i> отображается серым. Опция <i>Использовать массовые запросы</i> у SNMP интерфейсов позволяет включать/отключать <u>массовую обработку</u> SNMP запросов по каждому интерфейсу отдельно.
<i>IP адрес</i>	IP адрес узла сети (опционально).
<i>DNS имя</i>	DNS имя узла сети (опционально).
<i>Подключаться через</i>	Нажатие на соответствующую кнопку подскажет Zabbix серверу что использовать для получения данных от агентов: IP - Подключение к узлу сети, используя IP адресу (рекомендуется) DNS - Подключение к узлу сети, используя DNS имя
<i>Порт</i>	Номер TCP/UDP порта. Значения по умолчанию: 10050 для Zabbix агента, 161 для SNMP агента, 12345 для JMX и 623 для IPMI.
<i>По умолчанию</i>	Отметьте, чтобы интерфейс считался по умолчанию.
<i>Наблюдение через прокси</i>	Узел сети может наблюдаться через Zabbix сервер или через один из Zabbix прокси: (без прокси) - узел сети наблюдается Zabbix сервером Имя прокси - узел сети наблюдается через Zabbix прокси “Имя прокси”
<i>Активировано</i>	Отметьте, чтобы сделать узел сети активным, готовым к мониторингу. Если не отмечено, узел сети неактивен, таким образом не наблюдается.
Вкладка Шаблоны	позволяет вам присоединить <u>шаблоны</u> к узлу сети. Все объекты (элементы данных, триггеры, графики и группы элементов данных) будут унаследованы из шаблона.

Для присоединения нового шаблона, начните вводить имя шаблона в поле *Соединить с новыми шаблонами* до тех пор пока не появится список совпадающих шаблонов. Прокрутите список и выберете нужное. Когда все шаблоны будут выбраны, нажмите на *Добавить*.

Для отсоединения шаблона используйте одну из двух опций в блоке *Присоединенные шаблоны*:

- **Отсоединить** - отсоединение шаблона, с сохранением всех его элементов данных, триггеров и графиков
- **Отсоединить и очистить** - отсоединение шаблона и удаление всех его элементов данных, триггеров и графиков

Имена перечисленных шаблонов являются ссылками, которые ведут на диалог настройки шаблона.

Вкладка **IPMI** содержит атрибуты управления по IPMI.

Параметр	Описание
<i>Метод аутентификации</i>	Выбор алгоритма аутентификации.
<i>Уровень привилегий</i>	Выбор уровня привилегий.
<i>Имя пользователей</i>	Имя пользователя для аутентификации.
<i>Пароль</i>	Пароль для аутентификации.
Вкладка Макросы	позволяет вам задать <u>пользовательские макросы</u> на уровне узла сети. Вы также здесь можете просмотреть макросы уровня шаблона и глобальные макросы, если выберите опцию

Макросы узла сети и унаследованные. Это то место, где отображаются все определенные пользовательские макросы для этого узла сети со своими раскрытыми значениями, а также информация о том откуда эти макросы.

MACRO	EFFECTIVE VALUE	TEMPLATE VALUE	GLOBAL VALUE (CONFIGURE)
[\$OS]	= test	Change	= "test"
[\$HOST_MACRO]	= snikdjemsc334	Remove	
[\$NESTED_TEMPLATE_MACRO]	= 25864b	Change	= "Template App Zabbix Agent: "25864b"
[\$SNMP_COMMUNITY]	= public	Change	
[\$TEMPLATE_MACRO]	= 25864	Remove	= "public"

[Add](#) [Cancel](#)

Для удобства имеются ссылки на настройку соответствующих шаблонов и глобальных макросов. Также имеется возможность изменить макрос уровня шаблона/глобальный на уровне узла сети, фактически создав копию этого макроса на узле сети.

Вкладка **Инвентарные данные узла сети** позволяет вам вручную ввести информацию о [инвентарных данных](#) узла сети. Вы также можете выбрать *Автоматически заполнение инвентарных данных*, или деактивировать заполнение инвентарных данных для этого узла сети.

Вкладка **Шифрование** позволяет вам требовать [шифрование](#) соединений с узлом сети.

Параметр	Описание
Подключения к узлу сети	Каким образом Zabbix сервер или прокси подключаются к Zabbix агенту на хосте: без шифрования (по умолчанию), используя PSK (pre-shared key) или сертификат.
Соединения с узла сети	Выберите какой тип подключений разрешен с узла сети (то есть с Zabbix агента и Zabbix sender). Можно выбрать несколько типов соединений одновременно (полезно при тестировании и переключении на другой тип соединения). По умолчанию - “Без шифрования”.
Издатель	Разрешенный эмитент сертификата. Сертификат сначала подтверждается CA (центром сертификации). Если он действительный, подписан с помощью CA, тогда можно использовать поле <i>Издатель</i> для более строгого ограничения разрешенных CA. Это поле предназначено для использования, если ваша Zabbix инсталляция использует сертификаты от нескольких CA. Если поле не заполнено, тогда принимается любой CA.
Тема	Разрешенная тема сертификата. Сертификат сначала подтверждается CA. Если он действительный, подписан с помощью CA, тогда можно использовать поле <i>Тема</i> , чтобы разрешить только одно значение строки <i>Тема</i> . Если поле пустое, тогда принимается любой сертификат, подписанный настроенным CA.
Идентификатор PSK	Строка идентификации pre-shared key.
PSK	Pre-shared key (строка в шестнадцатеричном формате). Максимальная длина: 512 шестнадцатеричных цифр (256-байт PSK), если Zabbix использует библиотеки GnuTLS или OpenSSL, 64 шестнадцатеричных цифр (32-байт PSK), если Zabbix использует библиотеку mbedtls (PolarSSL). Пример: 1f87b595725ac58dd977beef14b97461a7c1045b9a1c963065002c5473194952

Создание группы узлов сети

Для создания группы узлов сети в веб-интерфейсе Zabbix, выполните следующее:

- Перейдите в: *Настройка* → *Группы узлов сети*
- Нажмите на *Создать группу узлов сети* в верхнем правом углу экрана
- В диалоге введите параметры группы

Host groups

* Group name Europe/Latvia/Riga/Zabbix servers

Add Cancel

Все обязательные поля ввода отмечены красной звёздочкой.

Параметр	Описание
Имя группы	<p>Введите уникальное имя группы узлов сети.</p> <p>Чтобы создать вложенную группу узлов сети, используйте прямую косую черту '/', например Европа/Латвия/Рига/Zabbix сервера. Вы можете создать эту группу, даже если ни одна из родительских группы узлов сети (Европа/Латвия/Рига) не существуют. В этом случае создание этих родительских групп узлов сети зависит от желания пользователя; они не будут созданы автоматически.</p>
Применить права доступа ко всем подгруппам	<p>Косые черты в начале и в конце строки, несколько косых черт подряд не поддерживаются. Экранирование символа '/' не поддерживается.</p> <p>Вложенное представление групп узлов сети поддерживается начиная с Zabbix 3.2.0.</p> <p>Эта опция доступна только Zabbix Супер Админ пользователям и только при редактировании существующей группы узлов сети.</p> <p>Выберите эту опцию и нажмите на Обновить, чтобы применить одинаковый уровень прав доступа ко всем вложенным группам узлов сети. Для групп пользователей, которые могли иметь разные права доступа, назначенные на вложенные группы узлов сети, будет применён уровень прав доступа родительской группы узлов сети ко всем вложенным группам.</p> <p>Это действие является однократной опцией, которая не хранится в базе данных.</p> <p>Опция поддерживается начиная с Zabbix 3.4.0.</p>

Права доступа к вложенным группам узлов сети

- При создании дочерней группы узлов сети у уже существующей родительской группы узлов сети права доступа [группы пользователей](#) к дочерней группе узлов сети наследуются от родителя (например, при создании Рига/Zabbix сервера, если Рига уже существует)
- При создании родительской группы узлов сети при уже существующей дочерней группе узлов сети права доступа на родительскую группу узлов сети не задаются (например, при создании Рига, если Рига/Zabbix сервера уже существует)

7.1.2 Инвентарные данные

Обзор

Вы можете хранить инвентарные данные сетевых устройств в Zabbix.

В веб-интерфейсе Zabbix существует специальное меню *Инвентарные данные*. Тем не менее, по началу вы не увидите в меню никаких данных и это место не там где вам необходимо ввести 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 *Инвентарные данные* вы можете ввести такие детали как тип устройства, серийный номер, местоположение, ответственное лицо и прочие - данные, которые будут заполнять инвентарные данные.

Если URL включен в инвентарные данные и он начинается с 'http' или 'https', то в результате он будет для нажатия в разделе *Инвентарных данных*.

Автоматический режим

Также инвентарные данные узлов сети могут заполняться автоматически. Чтобы эта функция работала, при настройке узла сети в режиме инвентарных данных на вкладке *Инвентарные данные узла сети* нужно выбрать *Автоматически*.

Затем вы можете [настроить элементы данных узла сети](#), которые будут заполнять любые поля инвентарных данных своими значениями, указав поле назначения с соответствующим атрибутом (называемый *Заполнять поле инвентарных данных*) при настройке элемента данных.

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|cpupnum,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 инвентарных данных

Режим инвентарных данных можно выбрать в диалоге настройки узла сети.

Режим инвентарных данных по умолчанию для новых узлов сети выбирается на основе настройки *Режим инвентаризации узлов сети по умолчанию* в *Администрирование* → *Общие* → [Прочие](#).

По узлам сети добавляемым с помощью действий по сетевым обнаружениям или авторегистрации имеется возможность указать операцию *Установить режим инвентарных данных узлов сети*, выбрав ручной или автоматический режим. Эта операция переопределит настройку *Режим инвентаризации узлов сети по умолчанию*.

Просмотр инвентарных данных

Детали всех указанных инвентарных данных доступны в меню *Инвентарные данные*.

В *Инвентарные данные* → *Обзор* вы можете сгруппировать и просмотреть доступные данные по различным полям инвентарных данных.

В *Инвентарные данные* → *Узлы сети* вы можете увидеть все узлы сети содержащие инвентарные данные. При нажатии на имя узла сети, в диалоге будут раскрыты все детали.

Host inventory

Overview Details

Host name Zabbix server 1
Visible name Zabbix server

Agent interfaces	IP address	DNS name	Connect to	Port	Default
	192.168.3.220		IP DNS	10050	<input checked="" type="radio"/>
SNMP interfaces	127.0.0.1		IP DNS	161	<input type="radio"/>

OS Linux linux-qvvt 3.11.10-21-default #1 SMP Mon Jul 21 15:28:46 U
Description Added on 2015-07-28.
Monitoring Web Latest data Triggers Problems Graphs Screens
Configuration Host Applications 13 Items 81 Triggers 47 Graphs 12 Discovery 3 Web 1

[Cancel](#)

Вкладка **Обзор** отображает:

Параметр	Описание
Имя узла сети	Имя узла сети. Если нажать на имя, появится меню с заданными для этого узла сети скриптами. Имя узла сети отображается с оранжевой иконкой в случае, если узел сети находится в обслуживании.
Видимое имя	Видимое имя узла сети (если задано).
Интерфейсы (Agent, SNMP, JMX, IPMI) узла сети	Этот блок отображает детали 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, Триггеры, Проблемы, Графики, Комплексные экраны.
Настройка	Ссылки на разделы настройки этого узла сети: Узел сети, Группы элементов данных, Элементы данных, Триггеры, Графики, Обнаружение, Веб. Количество добавленных объектов указано в скобках после каждой ссылки.

Вкладка **Детали** отображает все поля инвентарных данных, которые заполнены (не пустые).

Макросы инвентарных данных

Имеются доступные узлу сети макросы инвентарных данных {INVENTORY.*}, которые используются в оповещениях, например:

“Сервер {INVENTORY.LOCATION1} имеет проблему, ответственное лицо {INVENTORY.CONTACT1}, номер телефона {INVENTORY.POC.PRIMARY.PHONE.A1}.”

Для получения более подробных сведений, перейдите на страницу [Макросы поддерживаемые по назначению](#).

7.1.3 Массовое обновление

Обзор

Порой вы можете захотеть изменить какие-нибудь атрибуты для нескольких узлов сети одновременно. Вместо открытия каждого узла сети для редактирования, вы можете воспользоваться функцией массового обновления.

Использование массового обновления

Для массового обновления каких-либо узлов сети, выполните следующее:

- Отметьте узлы сети из списка, которые вы хотите обновить, из списка [узлов сети](#)
- Нажмите на *Массовое обновление* ниже списка
- Перейдите в необходимую вкладку атрибутов (*Узел сети*, *Шаблоны*, *IPMI*, *Инвентаризация* или *Шифрование*)
- Отметьте атрибуты для обновления и введите для них новые значения

The screenshot shows the 'Hosts' configuration dialog. The 'Host' tab is selected. There are three main sections: 'Replace host groups' (checkbox checked, dropdown 'Discovered hosts' with 'type here to search'), 'Add new or existing host groups' (checkbox checked, dropdown 'type here to search'), and 'Remove host groups' (checkbox checked, dropdown 'Zabbix servers' with 'type here to search'). Below these are fields for 'Description' (checkbox unchecked, value 'Original'), 'Monitored by proxy' (checkbox checked, dropdown 'Remote proxy'), and 'Status' (checkbox unchecked, value 'Original'). At the bottom are 'Update' and 'Cancel' buttons.

Заменить группы узла сети уберет узел сети из текущих групп и заменит на указанную(ые) в данном поле.

Добавить новые или существующие группы узлов сети позволяет выбрать дополнительные группы узла сети из существующих или назначить совершенно новые группы узлов сети для выбранных узлов сети.

Удалить группы узлов сети позволяет удалить выбранные группы узлов сети с узлов сети. В случае, когда узлы сети уже находятся в выбранных группах, тогда узлы сети будут удалены из этих групп. В случае, когда узлы сети не находятся в выбранных группах, тогда ничего не будет добавлено или удалено. В случае, когда одни и те же группы узлов сети заменяются и удаляются одновременно, узлы сети фактически остаются без групп.

Эти поля имеют функцию авто-дополнения - начните набирать и в них появится выпадающий список совпадающих групп узлов сети. Если группа узла сети новая, она также появится в выпадающем списке и будет отмечена в конце строки как *(новая)*. Просто прокрутите вниз для выбора.

Host Templates IPMI Inventory Encryption

Link templates

Template SNMP Device

type here to search

Replace
 Clear when unlinking

Для обновления присоединенных шаблонов на вкладке **Шаблоны**, отметьте *Присоединить шаблоны* и начните вводить имя шаблона в поле с автодополнением до появления выпадающего списка предлагающего совпадающие доступные шаблоны. Просто используйте прокрутку для выбора присоединяемого шаблона.

Опция *Заменить* позволяет добавить новый шаблон к узлу сети взамен любых других шаблонов соединенных с узлами сети ранее. Опция *Очистить при отсоединении* позволяет не только отсоединить какие-либо ранее присоединенные шаблоны, но также удалить все элементы унаследованные от них (элементы данных, триггеры, и т.д.).

Host Templates IPMI Inventory Encryption

IPMI authentication algorithm Original

IPMI privilege level Operator

IPMI username Original

IPMI password Original

Host Templates IPMI Inventory Encryption

Inventory mode Disabled Manual Automatic

Type Switch

Type (Full details) Original

Name Original

Alias Original

Чтобы появилась возможность массового обновления полей инвентарных данных, *Режим инвентарных данных* должен быть задан значениями 'Вручную' или 'Автоматически'.

The screenshot shows the Zabbix configuration interface for network encryption. The top navigation bar includes tabs for Host, Templates, IPMI, Inventory, and Encryption, with Encryption being the active tab. A sidebar on the left has a 'Connections' section with a checked checkbox. The main content area is divided into sections for 'Connections to host' and 'Connections from host'. Under 'Connections to host', there are three options: 'No encryption' (selected), 'PSK', and 'Certificate'. Under 'Connections from host', 'No encryption' is checked, while 'PSK' and 'Certificate' are unchecked. Below these sections are two input fields: 'PSK identity' and 'PSK'.

Когда сделаете все необходимые изменения, нажмите на кнопку *Обновить*. Атрибуты будут обновлены соответственно всем выбранным узлам сети.

7.2 Элементы данных

Обзор

Элементы данных являются теми сущностями, кто собирает данные с узла сети.

Как только вы настроили узел сети, вам необходимо добавить какие-нибудь элементы данных для мониторинга, для начала фактического сбора данных.

Каждый элемент данных является отдельной метрикой. Один из способов быстро добавить большое количество элементов данных это присоединить один из предопределенных шаблонов к узлу сети. Для оптимизации производительности системы, вам возможно потребуется выполнить тонкую настройку шаблонов, чтобы у вас было ровно столько элементов данных, а так же насколько часто эти элементы данных будут собирать данные, насколько это действительно необходимо.

В каждом элементе данных вы указываете какого рода данные будут собираться с узла сети.

Для этой цели воспользуйтесь [ключем элемента данных](#). Таким образом 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.

Для того чтобы указать дополнительные параметры для ключа, вы должны включить их в квадратные скобки после имени ключа. Таким образом, **system.cpu.load[avg5]** будет возвращать загрузку процессора за последние 5 минут, а **net.if.in[eth0]** вернет входящий трафик на интерфейсе eth0.

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](#).

7.2.1 Создание элемента данных

Обзор

Для создания элемента данных в веб-интерфейсе Zabbix, выполните следующее:

- Перейдите в: *Настройка* → *Узлы сети*
- Нажмите на *Элементы данных* в строке с узлом сети
- Нажмите на *Создать элемент данных* в правом верхнем углу экрана
- В диалоге введите параметры элемента данных

Вы также можете создать элемент данных, открыв уже существующий элемент данных, после чего нажать на кнопку *Клонировать* и затем сохранить под другим именем.

Настройка

Вкладка **Элемент данных** содержит следующие атрибуты элементов данных.

Item Preprocessing

* Name Incoming network traffic on eth0

Type Zabbix agent

* Key net.if.in[eth0] Select

* Host interface 192.168.6.87 : 10050

Type of information Numeric (unsigned)

Units bps

* Update interval 1m

Custom intervals

Type	Interval	Period	
Flexible	Scheduling	50s	1-7:00:00-24:00
Flexible	Scheduling	{\$FLEX_INTERVAL}	{\$FLEX_PERIOD}
Flexible	Scheduling	wd1-5h9-18	
Flexible	Scheduling	{\$SCHEDULING}	

Add

* History storage period 1w

* Trend storage period 365d

Show value As is show value mappings

New application

Applications

- None-
- CPU
- Filesystems
- General
- Memory
- Network interfaces
- OS
- Performance
- Processes
- Security

Populates host inventory field -None-

Description

Enabled

Add Cancel

Все обязательные поля ввода отмечены красной звёздочкой.

Параметр	Описание
Имя	Имя элемента данных. Поддерживаются следующие макросы, однако, их использование устарело: \$1, \$2...\$9 - ссылка на первый, второй... девятый параметры ключа элемента данных Например: Свободно дискового пространства в \$1

Параметр	Описание
<i>Тип</i>	<p>Если ключ элемента данных “vfs.fs.size[/,free]”, описание автоматически изменится на “Свободно дискового пространства в /”</p> <p>Тип элемента данных. Смотрите отдельные разделы по типам элементов данных.</p> <p>Ключ</p> <p>Ключ элемента данных.</p> <p>Поддерживаемые ключи элементов данных описаны в отдельных разделах по типам элементов данных.</p> <p>Ключ должен быть уникальным в пределах одного узла сети.</p> <p>Если тип ключа 'Zabbix агент', 'Zabbix агент (активный)', 'Простая проверка' или 'Zabbix агрегированный', то значение ключа должно поддерживаться Zabbix агентом или Zabbix сервером.</p> <p>Смотрите также: корректный формат ключа.</p> <p>Интерфейс узла сети</p> <p>Выбор интерфейса узла сети. Это поле доступно при изменении элемента данных на уровне узла сети.</p> <p>Тип хранения данных в базе данных после всех преобразований, если таковые имеются.</p> <p>Числовой (целое положительное) - целое 64битное число</p> <p>Числовой (с плавающей точкой) - число с плавающей точкой</p> <p>Отрицательные значения могут храниться.</p> <p>Допустимые значения -999999999999.9999 до 999999999999.9999.</p> <p>Начиная с Zabbix 2.2, также поддерживается получение значений в научном представлении. Например, 1e+7, 1e-4.</p> <p>Символ - короткие текстовые данные</p> <p>Журнал - длинные текстовые данные с необязательными свойствами для журналов (штамп времени, источник, важность, logeventid).</p> <p>Текст - длинные текстовые данные. Смотрите также ограничения по текстовым данных.</p> <p>Если указан символ единицы измерения, Zabbix добавит пост обработку полученного значения и отобразит его с заданным постфиксом единицы измерения. По умолчанию, если исходное значение превышает 1000, оно делится на 1000 и так отображается. Например, если вы задали <i>bps</i> и полученное значение равно 881764, оно будет отображено как 881.76 Kbps.</p> <p>Для единиц измерения B (байт), Bps (байты в секунду) используется специальная обработка, при которой значение делится на 1024. Таким образом, если единица измерения указана как B или Bps, Zabbix будет отображать:</p> <p>1 как 1B/1Bps 1024 как 1KB/1KBps 1536 как 1.5KB/1.5KBps</p> <p>Специальная обработка используется и для следующих единиц измерения связанных со временем:</p> <p>unixtime - переводится в “гггг.мм.дд чч:мм:сс”. Для корректного перевода, возвращаемое значение должно быть с типом данных Числовой (целое положительное).</p> <p>uptime - переводится в “чч:мм:сс” или в “N дней, чч:мм:сс”</p> <p>Например, если вы получили значение равное 881764 (секунд), оно отобразится как “10 дней, 04:56:04”</p> <p>s - переводится в “ггг мmm дdd ччч мmm сcc мс”; параметр рассматривается как количество секунд.</p> <p>Например, если вы получили значение равное 881764 (секунд), оно будет отображаться как “10д 4ч 56м”</p> <p>Отображаются только 3 верхних основы, такие как “1м 15д 5ч” или “2ч 4м 46с”. В случае, если нет дней, то тогда отображаются только два уровня - “1м 5ч” (минуты, секунды или миллисекунды не будут отображаться). Будет переведено в “< 1 ms”, если значение меньше чем 0.001.</p> <p><i>Обратите внимание</i>, что если единица измерения представлена с ! префиксом, тогда к значениям элементов данных префиксы/обработка единиц измерения применяться не будут. Смотрите чёрный список единиц измерения.</p>
<i>Тип информации</i>	<p>Единица измерения</p>
<i>Единица измерения</i>	<p>unixtime - переводится в “гггг.мм.дд чч:мм:сс”. Для корректного перевода, возвращаемое значение должно быть с типом данных Числовой (целое положительное).</p> <p>uptime - переводится в “чч:мм:сс” или в “N дней, чч:мм:сс”</p> <p>Например, если вы получили значение равное 881764 (секунд), оно отобразится как “10 дней, 04:56:04”</p> <p>s - переводится в “ггг мmm дdd ччч мmm сcc мс”; параметр рассматривается как количество секунд.</p> <p>Например, если вы получили значение равное 881764 (секунд), оно будет отображаться как “10д 4ч 56м”</p> <p>Отображаются только 3 верхних основы, такие как “1м 15д 5ч” или “2ч 4м 46с”. В случае, если нет дней, то тогда отображаются только два уровня - “1м 5ч” (минуты, секунды или миллисекунды не будут отображаться). Будет переведено в “< 1 ms”, если значение меньше чем 0.001.</p> <p><i>Обратите внимание</i>, что если единица измерения представлена с ! префиксом, тогда к значениям элементов данных префиксы/обработка единиц измерения применяться не будут. Смотрите чёрный список единиц измерения.</p>

Параметр	Описание			
Интервал обновления	<p>Получение нового значения по этому элементу данных каждые N секунд. Максимально допустимый интервал обновления - 86400 секунд (1 день). Функции времени поддерживаются, например, 30s, 1m, 2h, 1d.</p> <p>Поддерживаются пользовательские макросы.</p> <p><i>Обратите внимание:</i> Если задано значение равное '0', элемент данных не будет обрабатываться. Однако, если также существует пользовательский интервал (гибкий/по расписанию) с ненулевым значением, элемент данных будет обработан в течении длительности пользовательского интервала.</p> <p><i>Обратите внимание,</i> что у существующего пассивного элемента данных можно выполнить опрос значения немедленно, нажав на кнопку Проверить сейчас.</p> <p>Вы можете создавать пользовательские правила проверки элемента данных:</p> <p>Гибкий - создание исключений из <i>Интервала обновления</i> (интервал с другой частотой обновления)</p> <p>По расписанию - создание пользовательского расписания проверки.</p>			
Пользовательские интервалы	<p>Для получения более подробной информации смотрите Пользовательские интервалы.</p> <p>Функции времени поддерживаются в поле <i>Интервал</i>, например, 30s, 1m, 2h, 1d.</p> <p>Поддерживаются пользовательские макросы.</p> <p>Проверка по расписанию поддерживается начиная с Zabbix 3.0.0.</p> <p><i>Обратите внимание:</i> Недоступно для активных элементов данных Zabbix агента.</p> <p>Количество дней хранения в базе данных детальной истории (от 1 часа до 25 лет). Более старые данные будут удалены с помощью функции автоматической очистки истории базы данных.</p> <p>Хранится в секундах. Функции времени поддерживаются, например, 2h, 1d.</p> <p>Поддерживаются пользовательские макросы.</p> <p>Данное значение можно переопределить глобально в <i>Администрирование</i> → <i>Общие</i> → <i>Очистка истории</i>. Если опция активирована, то вы увидите предупреждение:</p> <table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="padding: 2px;">History storage period</td> <td style="padding: 2px; border-left: none;">1w</td> <td style="padding: 2px; border-left: none;">Overridden by global housekeeping settings (1d)</td> </tr> </table>	History storage period	1w	Overridden by global housekeeping settings (1d)
History storage period	1w	Overridden by global housekeeping settings (1d)		
Период хранения истории	<p>Рекомендуется хранить записанные значения как можно меньшее количество дней для уменьшения размера истории в базе данных. Вместо долговременного хранения истории значений, вы можете хранить более долгий срок данных динамики изменений.</p> <p>Смотрите также История и динамика изменений.</p> <p>Хранение усредненных значений (ежечасные мин, макс, сред, количество) детальной истории N дней в базе данных (от 1 дня до 25 лет). Более старые данные будут удалены с помощью функцией автоматической очистки истории базы данных.</p> <p>Хранится в секундах. Функции времени поддерживаются, например, 24h, 1d.</p> <p>Поддерживаются пользовательские макросы.</p> <p>Данное значение можно переопределить глобально в <i>Администрирование</i> → <i>Общие</i> → <i>Очистка истории</i>. Если опция активирована, то вы увидите предупреждение:</p> <table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="padding: 2px;">Trend storage period</td> <td style="padding: 2px; border-left: none;">365d</td> <td style="padding: 2px; border-left: none;">Overridden by global housekeeping settings (7d)</td> </tr> </table>	Trend storage period	365d	Overridden by global housekeeping settings (7d)
Trend storage period	365d	Overridden by global housekeeping settings (7d)		
Период хранения динамики изменений	<p><i>Обратите внимание:</i> Хранение динамики изменений недоступно для не числовых данных - символ, журнал и текст.</p> <p>Смотрите также История и динамика изменений.</p>			
Отображение значений	<p>Применение преобразования значений к этому элементу данных. Преобразование значений не меняет полученные значения, оно служит только для отображаемых данных.</p> <p>Работает только с целыми числовыми элементами данных.</p> <p>Например, "Windows service states".</p>			
Формат времени журнала	<p>Доступен только для элементов данных типа Журнал. Поддерживаемые значения:</p> <ul style="list-style-type: none"> * y: Год (1970-2038) * M: Месяц (01-12) * d: День (01-31) * h: Час (00-23) 			

Параметр	Описание
	<p>* m: Минута (00-59) * s: Секунда (00-59)</p> <p>Если оставить это поле пустым, то штамп времени не будет обрабатываться. 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 PID, далее дата, время и остальная часть строки.</p> <p>Формат времени журнала для этой строки должен быть “rrrrrr:ууууММдд:hhmmss”.</p> <p>Обратите внимание, что символы “р” и “:” являются лишь заменителями и могут быть какими угодно, кроме “уMdhms”.</p>
<i>Новая группа элементов данных</i>	Введите имя новой группы элементов данных для этого элемента данных.
<i>Группы элементов данных</i>	Соединение элемента данных с одним или несколькими существующими группами элементов данных.
<i>Заполнение поля инвентаря узла сети</i>	Вы можете выбрать поле инвентарных данных, которое будет заполняться значением элемента данных. Функция будет работать, если у узла сети включено автоматическое заполнение инвентарных данных .
<i>Описание</i>	Введите описание элемента данных.
<i>Активировано</i>	Отметьте для активации элемента данных, таким образом он будет обрабатываться. При редактировании элемента данных уровня шаблона на уровне узла сети, некоторое количество полей доступно только для чтения. Вы можете воспользоваться ссылкой из заголовка диалога и перейти на уровень шаблона и отредактировать элемент данных там, однако имейте в виду, что изменение элемента данных коснется всех узлов сети к которым присоединен шаблон.

Предобработка значений элемента данных

Вкладка **Предобработка** позволяет задать правила преобразования полученных значений. Можно использовать одно или несколько правил предобработки до сохранения значений в базу данных. Преобразования выполняются в том порядке, в котором они были добавлены. Вся предобработка выполняется Zabbix сервером.

Смотрите также: [Детальная информация по предварительной обработке](#)

Item Preprocessing

Preprocessing steps	Name	Parameters
Change per second		
Custom multiplier	8	
Regular expression	pattern	output

Text

- Regular expression**
- Trim
- Right trim
- Left trim

Structured data

- XML XPath
- JSON Path

Arithmetic

- Custom multiplier

Change

- Simple change
- Change per second

Numerical systems

- Boolean to decimal
- Octal to decimal
- Hexadecimal to decimal

В параметрах предварительной обработки значений элементов данных поддерживаются [пользовательские макросы](#) и пользовательские макросы с контекстом.

Преобразование

Описание

Совпадение значения с регулярным выражением <шаблона> и замена значения в соответствии с <выводом>. Регулярное выражение поддерживает извлечение до 10 захватенных групп в \N последовательности. Элемент данных станет неподдерживаемым в случае ошибки при поиске соответствия во входящем значении.

Параметры:

Регулярное выражение

шаблон - регулярное выражение

вывод - шаблон форматирования вывода. \N (где N=1..9) - управляющая последовательность заменяется N-нной совпадающей группой.

Управляющая последовательность \0 заменяется совпадающим текстом. Поддерживается начиная с 3.4.0.

Пожалуйста, обратитесь в разделу [регулярных выражений](#) для ознакомления с некоторыми существующими примерами.

Обрезка

Удаление указанных символов с начала или конца значения.

Обрезка справа

Удаление указанных символов с конца значения.

Обрезка слева

Удаление указанных символов с начала значения.

Извлечение значения или фрагмента из XML данных с использованием XPath функционала.

Чтобы эта опция заработала, Zabbix сервер необходимо скомпилировать 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 на то, что пространства имен не поддерживаются.
Поддерживается начиная с 3.4.0.
```

XML XPath

Преобразование

Описание

JSON Path

Извлечение значения или фрагмента с JSON данных с использованием простейшего набора функций JSONPath функционала.

Примеры:

```
$.document.item.value извлечёт 10 с {"document": {"item": {"value": 10}}}  
$.document.item извлечёт {"value": 10} с {"document": {"item": {"value": 10}}}  
$['a document'].item.value извлечёт 10 с {"a document": {"item": {"value": 10}}}  
$.document.items[1].value извлечёт 20 с {"document": {"items": [{"value": 10}, {"value": 20}]}}
```

Обратите внимание, что поддерживаются только прямые пути к одиночным объектам в обозначении точкой или квадратной скобкой.

В JSONPath обозначении точкой (`$.a.b.c`) можно использовать только буквенно-цифровые символы + символ подчёркивания . Если имя объекта JSON содержит другие символы, необходимо использовать обозначение квадратной скобкой (`$['a']['b']['c']`). Допускается смешивание обозначений (`$.a['b'].c`)

Извлечение нескольких значений не поддерживается.

Поддерживается начиная с 3.4.0.

Умножение значения на заданное целое или дробное число.

Пользовательский множитель

Использование этой опции сконвертирует полученные значения в KB, MBps, и т.д. в B, Bps. В противном случае Zabbix не сможет корректно подставить префиксы (К, М, Г и т.д.).

Простое изменение

Начиная с Zabbix 2.2, также поддерживается научное представление. Например, $1e+70$.

Вычисление разницы между текущим и предыдущим значениями.

Вычисляется **значение-пред_значение**, где

значение - текущее значение; **пред_значение** - ранее полученное значение
Эта опция может быть полезной для измерения постоянно растущего значения. Если текущее значение меньше предыдущего, тогда Zabbix отбрасывает это изменение (ничего не записывает) и ждет следующего значения.

На один элемент данных разрешена только одна операция изменения.

Вычисление изменения значения (разница между текущим и предыдущим значениями) скорость в секунду.

Вычисляется **(значение-пред_значение)/(время-пред_время)**, где
значение - текущее значение; **пред_значение** - ранее полученное значение;
Время - текущий штамп времени; **пред_значение** - штамп времени предыдущего значения.

Эта настройка крайне полезна при получении скорости в секунду в случае постоянно увеличивающихся значений. Если текущее значение меньше чем предыдущее значение, Zabbix отбрасывает эту разницу (ничего не записывает) и ждет следующее значение. Такое поведение помогает отрабатывать, например, переполнение 32-битных счетчиков SNMP.

Изменение в секунду

Обратите внимание: Так как результатом этого вычисления может быть число с плавающей точкой, рекомендуется использовать 'Тип информации' - Числовой (с плавающей точкой), даже если входящие значения целые положительные числа. Это особенно актуально для маленьких чисел где дробная часть имеет значение. Если числа с плавающей точкой очень большие и могут превысить длину поля в этом случае введенное значение может быть потеряно, в таком случае предлагается использовать тип Числовой (целое положительное) и это отбросит только дробную часть.
На один элемент данных разрешена только одна операция изменения.

Двоичное в десятичное

Конвертация значения из логического в десятичный формат. Текстовое представление переводится в значения 0 или 1. Таким образом, 'TRUE'

Преобразование

Описание

хранится как 1 и 'FALSE' хранится как 0. Все значения сопоставляются чувствительным к регистру способом. В данное время распознаются значения для:

TRUE - true, t, yes, y, on, up, running, enabled, available, ok, master

FALSE - false, f, no, n, off, down, unused, disabled, unavailable, err, slave

Кроме того, любое ненулевое значение считается TRUE, нулевое считается FALSE.

Следующие значения поддерживаются начиная с 4.0.0: ok, master, err, slave.

Восьмеричное в десятичное

Конвертация значения из восьмеричного в десятичный формат.

Шестнадцатеричное в десятичное

Конвертация значения из шестнадцатеричного в десятичный формат.

Если вы используете пользовательский множитель или хранение значений как *Изменение в секунду* у элементов данных с типом информации *Числовой (целое положительное)* и результирующее вычисленное значение в действительности дробное число, тогда вычисленное значение всё же будет считаться приемлемым, однако дробная часть будет обрезана и значение запишется как целое положительное.

Кнопки диалога

Кнопки в нижней части диалога позволяют выполнить несколько видов операций.

Add

Добавление элемента данных. Эта кнопка доступна только для новых элементов данных.

Update

Обновление свойств элемента данных.

Clone

Создание другого элемента данных на основе свойств текущего элемента данных.

Check now

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.

Clear history and trends

Удаление истории и динамики изменений элемента данных.

Delete

Удаление элемента данных.

Cancel

Отмена изменения свойств элемента данных.

Ограничения текстовых данных

Ограничения текстовых данных зависят от выбранной базы данных. Перед сохранением текстовых значений в базу данных, эти значения обрезаются в соответствии с ограничением типа значений в базе данных:

База данных	Тип информации		
	Символ	Журнал (лог)	Текст
MySQL	255 символов	65536 байт	65536 байт
PostgreSQL	255 символов	65536 символов	65536 символов
Oracle	255 символов	65536 символов	65536 символов
DB2	255 байт	2048 байт	2048 байт

Чёрный список единиц измерения

По умолчанию, указывая единицу измерения элементу данных, вы увидите добавленный префикс множителя - например, значение '2048' с единицей измерения 'B' будет отображаться как '2KB'.

Однако, используя ! префикс, можно избежать конвертации любой единицы измерения, например !В. Чтобы лучше проиллюстрировать то, каким образом это преобразование работает с и без чёрного списка, просмотрите следующие примеры значений и единиц измерения:

```
1024 !B -> 1024 B  
1024 B -> 1 KB  
61 !s -> 61 s  
61 s -> 1m 1s  
0 !uptime -> 0 uptime  
0 uptime -> 00:00:00  
0 !! -> 0 !  
0 ! -> 0
```

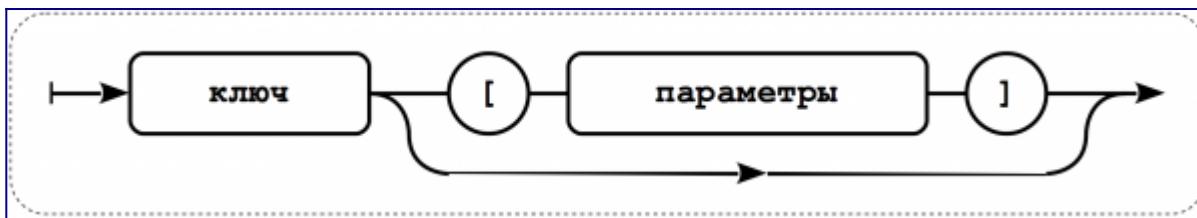
До Zabbix 4.0, имелся зашитый в код чёрный список единиц измерения, который состоял из ms, rpm, RPM, %. Этот чёрный список считается устаревшим, поэтому правильным способом внесения в чёрный список подобных единиц измерения будет !ms, !rpm, !RPM, !%.

Неподдерживаемые элементы данных

Элемент данных может стать неподдерживаемым, если его значение не было получено по некоторым причинам. Такие элементы данных будут перепроверены через определенный интервал времени, настраиваемый в [разделе Администрирования](#).

7.2.1.1 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 следующим образом - если в каком-то блоке может быть достигнута линия, в таком случае разрешено, если линия не может быть достигнута - тогда не разрешено.



Для построения правильного ключа элемента данных, вы должны начать с указания имени ключа, то есть выбрать имеет ли ключ параметры или нет - о чем свидетельствуют две линии, которые вы можете достичь.

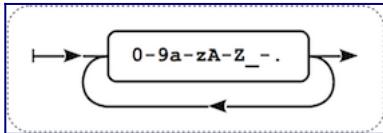
Имя ключа

Имя ключа имеет ограниченный диапазон разрешенных символов, которые просто следуют друг за другом. Разрешенные символы:

0-9a-zA-Z_- .

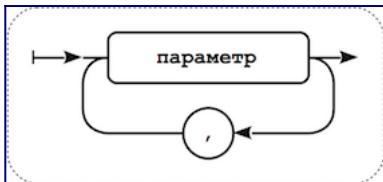
Что означает:

- все цифры;
- все буквы в нижнем регистре;
- все буквы в верхнем регистре;
- подчеркивание;
- дефис;
- точка.



Параметры ключа

Ключ элемента данных может принимать множество параметров, которые должны быть разделены запятой.



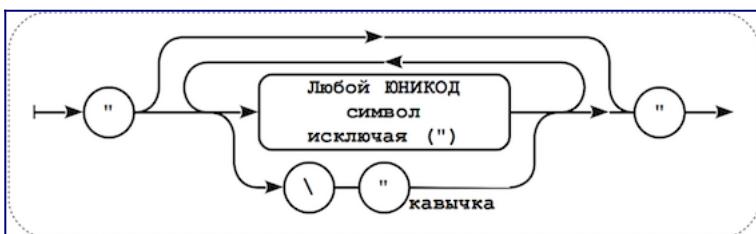
Каждый параметр ключа может быть одним из: строка заключенная в кавычки, строка без кавычек, массив.



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 **icstrping[,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 заключенная в кавычки

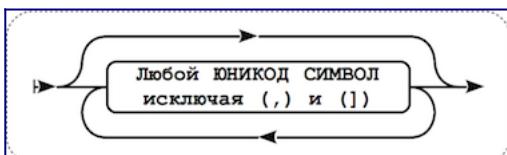
Если параметр ключа это строка, заключенная в кавычки, тогда разрешен любой символ в Юникоде, и если требуется включение двойных кавычек, то они должны быть экранированы обратной наклонной чертой.



Чтобы заключить параметры ключа элемента данных в кавычки, используйте только двойные кавычки. Одинарные кавычки не поддерживаются.

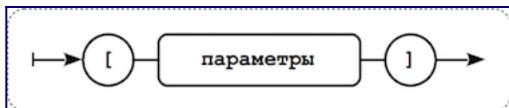
Параметр - строка не заключенная в кавычки

Если параметр ключа это строка без кавычек, тогда разрешен любой символ в Юникоде, за исключением запятой и правой квадратной скобки (]). Параметр, который не заключен в кавычки, не может начинаться с левой квадратной скобки ([).



Параметр - массив

Если параметр ключа это массив, тогда он должен быть заключен в квадратные скобки, в которых каждый индивидуальный параметр следует один за другим, согласно правилам и синтаксису.



Многоуровневые массивы параметров, например `[a, [b, [c, d]], e]`, недопускаются.

7.2.1.2 Пользовательские интервалы

Обзор

Имеется возможность создания пользовательских правил относительно времени, когда элемент данных будет опрашиваться. Для этого есть два способа, *Переменные интервалы*, который позволяет переопределить интервал обновления по умолчанию, и *По расписанию*, посредством чего элемент данных может быть опрошен в конкретное время или последовательность времени.

Гибкие интервалы

Гибкие интервалы позволяют переопределить интервал обновления по умолчанию указанными периодами времени. Гибкий интервал указывается с *Интервал* и *Период*, где:

- *Интервал* – интервал обновления в указанный период времени
- *Период* – период времени, когда гибкий интервал активен (смотрите [периоды времени](#) для подробного описания формата *Период*)

Возможно указать до семи переменных интервалов. Если несколько гибких интервалов перекрываются, то используется *Интервал* с наименьшим значением для перекрывающегося периода. Обратите внимание, что если наименьший интервал из перекрывающих имеет значение “0”, то проверка не будет произведена вообще. В остальных случаях будет использоваться интервал обновления по умолчанию.

Обратите внимание что, если гибкий интервал совпадает с длительностью периода, элемент данных будет опрошен только один раз. Если гибкий интервал больше чем период, элемент данных может быть опрошен один раз или может быть не проверен вовсе (поэтому такая настройка не рекомендуется). Если гибкий интервал меньше чем период, элемент данных будет опрошен по крайней мере один раз.

Если гибкий интервал задан равным '0', элемент данных не будет обрабатываться в течении периода гибкого интервала и опрос возобновится в соответствии с *Интервал обновления* по умолчанию сразу как только период завершится. Примеры:

Интервал	Период	Описание
10	1-5,09:00-18:00	Элемент данных будет опрашиваться каждые 10 секунд в течении рабочего времени.
0	1-7,00:00-7:00	Элемент данных не будет опрашиваться в течении ночи.
0	7-7,00:00-24:00	Элемент данных не будет опрашиваться по Воскресеньям.
60	1-7,12:00-12:01	Элемент данных будет опрошен ровно в 12:00 каждый день. Обратите внимание, что такой формат использовался как обходной путь при опросе по расписанию и начиная с Zabbix 3.0 рекомендуется использовать интервалы по расписанию для таких проверок.

Интервалы по расписанию

Интервалы по расписанию используются для опроса элементов данных в заданное время. Тогда как гибкие интервалы разрабатывались для переопределения интервала обновления элементов данных по умолчанию, интервалы по расписанию используются для обозначения независимого расписания опроса, которое выполняется параллельно.

Интервал по расписанию задается как:

md<фильтр>**wd**<фильтр>**h**<фильтр>**m**<фильтр>**s**<фильтр> where:

- **md** - дни месяца
- **wd** - дни недели
- **h** - часы
- **m** - минуты
- **s** – секунды

<фильтр> используется для того, чтобы указать значения по своему префиксам (дни, часы, минуты, секунды) и задается как: [<от> [-<до>]] [/<шаг>] [, <фильтр>] где:

- <от> и <до> определяют диапазон соответствующих значений (включительно). Если <до> не указывается, тогда фильтр совпадает с диапазоном <от> - <от>. Если <от> также не указан, тогда фильтр совпадает со всеми возможными значениями.
- <шаг> определяет пропуски количества значений через диапазон. По умолчанию значение <шага> равно 1, что означает что все значения указанного диапазона совпадают.

Хотя добавление фильтра опционально, по крайней мере один фильтр должен использоваться. Фильтр должен определять либо диапазон, либо значение <шага>.

Пустой фильтр соответствует либо '0', если фильтр более низкого уровня не определен, либо, в противном случае, все возможные значения. Например, если фильтр часа опущен, тогда будет соответствовать только '0', если фильтры минут и секунд также опущены, в противном случае пустой фильтр часа будет соответствовать всем значениям часа.

Допустимые значения <от> и <до> по их соответствующему префиксам фильтра:

Префикс Описание <от> <до>

md	Дни месяца	1-31	1-31
wd	Дни недели	1-7	1-7
h	Часы	0-23	0-23
m	Минуты	0-59	0-59
s	Секунды	0-59	0-59

Значение <от> должно быть меньше или равно значению <до>. Значение <шага> должно быть больше или равно 1 и меньше или равно <от> - <до>.

К одиночным цифрам значений дней месяца, часов, минут и секунд можно добавлять префикс 0. Например, md01-31 и h/02 являются допустимыми интервалами, но md01-031 и wd01-07 - нет.

В веб-интерфейсе Zabbix, несколько интервалов по расписанию вводятся в отдельных строках. В Zabbix API, они складываются в одну строку с разделителем точкой с запятой ; .

Если время совпадает по нескольким интервалам, опрос выполнится только раз. Например, wd1h9 ; h9 выполнится только один раз в Понедельник в 9 утра.

Примеры:

Интервал	Когда выполнится
m0-59	каждую минуту

Интервал	Когда выполнится
h9-17/2	каждые 2 часа, начиная с 9:00 (9:00, 11:00 ...)
m0,30 или m/30	каждый час в чч:00 и чч:30
m0,5,10,15,20,25,30,35,40,45,50,55 или m/5	каждые пять минут
wd1-5h9	каждый Понедельник до Пятницы в 9:00
wd1-5h9-18	каждый Понедельник до Пятницы в 9:00,10:00,...,18:00
h9,10,11 или h9-11	каждый день в 9:00, 10:00 и 11:00
md1h9m30	каждый 1ый день каждого месяца в 9:30
md1wd1h9m30	каждый 1ый день каждого месяца в 9:30, если этот день Понедельник
h9m/30	каждый день в 9:00, 9:30
h9m0-59/30	каждый день в 9:00, 9:30
h9,10m/30	каждый день в 9:00, 9:30, 10:00, 10:30
h9-10m30	каждый день в 9:30, 10:30
h9m10-40/30	каждый день в 9:10, 9:40
h9,10m10-40/30	каждый день в 9:10, 9:40, 10:10, 10:40
h9-10m10-40/30	каждый день в 9:10, 9:40, 10:10, 10:40
h9m10-40	каждый день в 9:10, 9:11, 9:12, ... 9:40
h9m10-40/1	каждый день в 9:10, 9:11, 9:12, ... 9:40
h9-12,15	каждый день в 9:00, 10:00, 11:00, 12:00, 15:00
h9-12,15m0	каждый день в 9:00, 10:00, 11:00, 12:00, 15:00
h9-12,15m0s30	каждый день в 9:00:30, 10:00:30, 11:00:30, 12:00:30, 15:00:30
h9-12s30	каждый день в 9:00:30, 9:01:30, 9:02:30 ... 12:58:30, 12:59:30
h9m/30;h10	каждый день в 9:00, 9:30, 10:00

7.2.2 Типы элементов данных

Обзор

Типы элементов данных охватывают различные методы получения данных с вашей системы. Каждый тип элемента данных поставляется со своим собственным набором поддерживаемых ключей элементов данных и требуемых параметров.

В настоящее время Zabbix предлагает следующие типы элементов данных:

- [Zabbix агент проверки](#)
- [SNMP агент проверки](#)
- [SNMP тралы](#)
- [IPMI проверки](#)
- [Простые проверки](#)
 - [VMware мониторинг](#)
- [Мониторинг файлов журналов](#)
- [Вычисляемые элементы данных](#)
- [Внутренние проверки Zabbix](#)
- [SSH проверки](#)
- [Telnet проверки](#)
- [Внешние проверки](#)
- [Агрегированные проверки](#)
- [Траппер элементы данных](#)
- [JMX мониторинг](#)

- [ODBC проверки](#)
- [Зависимые элементы данных](#)
- [HTTP проверки](#)

Детальная информация по всем типам элементов данных включена в подстраницы этого раздела. Несмотря на это, некоторые типы элементов данных предлагают большее количество опций по сбору данных, дополнительные опции через [пользовательские параметры](#) или [подгружаемые модули](#).

Некоторые проверки выполняются Zabbix сервером в одиночку (так называемый безагентный мониторинг) в то время как остальные требуют Zabbix агента или даже Zabbix Java gateway (с JMX мониторингом).

Если отдельный тип элемента данных требует определенный интерфейс (например, IPMI проверка требует IPMI интерфейс на узле сети), то этот интерфейс должен существовать в определении этого узла сети.

Можно задвать несколько интерфейсов в определении узла сети: Zabbix агент, SNMP агент, JMX и IPMI. Если элемент данных может использовать более чем один интерфейс, он будет искать доступные интерфейсы у узла сети (в следующем порядке: Агент → SNMP → JMX → IPMI) и будет связан с первым подходящим ему интерфейсом.

Все элементы данных которые возвращают текст (символ, журнал, текстовый типы информации) теперь могут возвращать только пробелы (при необходимости), при этом возвращенное значение становится пустой строкой (поддерживается начиная с 2.0).

7.2.2.1 Zabbix агент

Обзор

Эти проверки используют общение с Zabbix агентом для сбора данных.

Существуют [пассивные и активные](#) проверки. При настройке элемента данных, вы можете выбрать требуемый тип:

- *Zabbix агент* - для пассивных проверок
- *Zabbix агент (активный)* - для активных проверок

Поддерживаемые ключи элементов данных

В таблице приводится подробная информация о ключах элементов данных, которые вы можете использовать элементами данных Zabbix агента.

Смотрите также:

- [Элементы данных поддерживаемые по платформам](#)
- [Специфичные ключи элементов данных для Windows агента](#)

Обязательные и необязательные параметры

Параметры без угловых скобок обязательны. Параметры, отмеченные угловыми скобками < >, необязательны.

Описание	Возвращаемое значение	Ключ	Параметры	Комментарии
Имя хоста агента.	Строка	agent.hostname		Возвращает действительное значение hostname агента из файла конфигурации.

Описание	Возвращаемое значение	Параметры	Комментарии
Проверка доступности агента.	Ничего - недоступен 1 - доступен	agent.ping	Используйте nodata() функцию триггера для проверки недоступности узла сети.
Версия Zabbix агента.	Строка	agent.version	Пример возвращаемого значения: 1.8.2
Максимальное количество открытых файлов поддерживаемое ОС.	Целое число	kernel.maxfiles	
Максимальное количество процессов поддерживаемое ОС.	Целое число	kernel.maxproc	
Мониторинг файлов журналов (логов).	Журнал (лог)	<p>log[файл,<регулярное выражение>,<кодировка>,<макс. кол-во строк>,<режим>,<вывод>,<максзадержка>]</p> <p>файл - абсолютный путь и имя файла журнала</p> <p>регулярное выражение - регулярное выражение описывающее требуемый шаблон содержимого</p> <p>кодировка - <u>идентификатор</u> кодовой страницы</p> <p>макс. кол-во строк - максимальное количество новых строк в секунду, которое агент будет отправлять Zabbix серверу или прокси. Этот параметр заменяет значение 'MaxLinesPerSecond' в <u>zabbix_agentd.conf</u></p> <p>режим - возможные значения: <i>all</i> (по умолчанию), <i>skip</i> - пропуск обработки старых данных (влияет только на недавно созданные элементы данных,</p>	<p>Элемент данных должен быть настроен <u>активной проверкой</u>. Если файл не существует или доступ к нему не разрешен, элемент данных переходит в состояние неподдерживается.</p> <p>Если параметр вывод оставить пустым - будет возвращена вся строка соответствующая регулярному выражению. Заметьте, что все глобальные регулярные выражения исключая 'Результат ИСТИНА' всегда возвращают всю строку соответствующую выражению и параметр вывод игнорируется.</p> <p>Извлечение содержания согласно параметра вывод производится на стороне агента.</p> <p>Примеры: $\Rightarrow \text{log}[/var/log/syslog]$ $\Rightarrow \text{log}[/var/log/syslog,error]$ $\Rightarrow \text{log}[/home/zabbix/logs/logfile,,100]$</p> <p><i>Пример использования параметра вывод для извлечения числа из записи в журнале:</i> $\text{log}[/app1/app.log,"task run [0-9.]+ sec,$</p>

Ключ

Описание	Возвращаемое значение	Параметры	Комментарии
----------	-----------------------	-----------	-------------

Подсчёт количества совпадающих строк в мониторинге файла журнала (логов).	которые еще не получали никаких данных). вывод - дополнительный шаблон форматирования вывода. \0 - управляющая последовательность заменяется найденным текстом, тогда \N (где N=1..9) - управляющая последовательность заменяется N-нной совпадающей группой (или пустой строкой, если N превышает количество найденных групп). максзадержка - максимальная задержка в секундах. Тип: число с плавающей точкой. Значения: 0 - (по умолчанию) никогда не игнорировать строки в файлах журналов; > 0.0 - игнорировать более старые строки с целью получения наиболее новых строк проанализированных в течении "максзадержка" секунд. Перед использованием прочтайте заметки по максзадержка !	processed ([0-9]+) records, [0-9]+ errors",,,,\'1] → будет соответствовать записи в журнале "2015-11-13 10:08:26 task run 6.08 sec, processed 6080 records, 0 errors" и на сервер отправится только число 6080. Так как отправляется число, у этого элемента данных журнала можно изменить "Тип информации" с "Журнал (лог)" на "Числовой (целое положительное)" и это значение можно будет использовать на на графиках, в триггерах и т.д.	Пример использования параметра вывод для изменения вывода 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 RECORDS: \3, ERRORS: \4, DURATION: \2"] → будет соответствовать записи в журнале "2015-11-13 10:08:26 task run 6.08 sec, processed 6080 records, 0 errors" и на сервер отправится измененная запись "2015-11-13 10:08:26 RECORDS: 6080, ERRORS: 0, DURATION: 6.08".
	log.count[файл,<регулярное выражение>,<кодировка>,<макс. кол-во строк>,<режим>,<максзадержка>]	файл - абсолютный путь и имя файла журнала регулярное выражение - регулярное выражение описывающее требуемый шаблон	Параметр режим поддерживается начиная с Zabbix 2.0. Параметр вывод поддерживается начиная с Zabbix 2.2. Параметр максзадержка поддерживается начиная с Zabbix 3.2. Смотрите дополнительную информацию о мониторинге файлов журналов .

Ключ

Описание	Возвращаемое значение	Параметры	Комментарии
Мониторинг файлов журналов (логов) с поддержкой ротации логов.	содержимого кодировка - идентификатор кодовой страницы макс. кол-во строк - максимальное количество новых строк в секунду, которое агент будет анализировать. Значением по умолчанию является $10^{*} \text{MaxLinesPerSecond}$ в zabbix_agentd.conf . режим - возможные значения: <i>all</i> (по умолчанию), <i>skip</i> - пропуск обработки старых данных (влияет только журналов). на недавно созданные элементы данных). максзадержка - максимальная задержка в секундах. Тип: число с плавающей точкой. Значения: 0 - (по умолчанию) никогда не игнорировать строки в файлах журналов; > 0.0 - игнорировать более старые строки с целью получения наиболее новых строк проанализированных в течении “максзадержка” секунд. Перед использованием прочтайте заметки по максзадержка!	Поддерживается начиная с Zabbix 3.2.0.	
	logrt[файл_regexp,<регулярное выражение>,<кодировка>,<макс. кол-во строк>,<режим>,<вывод>,<максзадержка>,<опции>]		
	файл_regexp - абсолютный путь к файлу и регулярное выражение описывающее его имя	Элемент данных должен быть настроен активной проверкой .	
	регулярное выражение - регулярное	Ротация журналов основывается на времени последнего изменения файлов.	
		Если параметр вывод оставить пустым - будет возвращена вся	

Ключ			
Описание	Возвращаемое значение	Параметры	Комментарии
	<p>выражение описывающее требуемый шаблон содержимого</p> <p>кодировка - идентификатор</p> <p>кодовой страницы</p> <p>макс. кол-во строк - максимальное количество новых строк в секунду, которое агент будет отправлять Zabbix серверу или прокси.</p> <p>Этот параметр заменяет значение 'MaxLinesPerSecond' в zabbix_agentd.conf</p> <p>режим - возможные значения: <i>all</i> (по умолчанию), <i>skip</i> - пропуск обработки старых данных (влияет только на недавно созданные элементы данных, которые еще не получали никаких данных).</p> <p>вывод - дополнительный шаблон форматирования вывода:</p> <ul style="list-style-type: none"> \0 - управляющая последовательность заменяется найденным текстом, тогда \N (где N=1..9) - управляющая последовательность заменяется N-ной совпадающей группой (или пустой строкой, если N превышает количество найденных групп). <p>максзадержка - максимальная задержка в секундах. Тип: число с плавающей точкой. Значения: 0 - (по умолчанию) никогда</p>	<p>строка соответствующая регулярному выражению. Заметьте, что все глобальные регулярные выражения исключая 'Результат ИСТИНА' всегда возвращают всю строку соответствующую выражению и параметр вывод игнорируется.</p> <p>Извлечение содержимого согласно параметра вывод производится на стороне агента.</p> <p>Примеры:</p> <p>⇒ logrt["/home/zabbix/logs/^logfile[0-9]{1,3}\$","","100"] → совпадает с файлом наподобии "logfile1" (но не совпадает с ".logfile1")</p> <p>⇒ logrt["/home/user/logfile_*_[0-9]{1,3}","pattern_to_match","UTF-8","100"] → будет собирать данные с файлов таких как "logfile_abc_1" или "logfile__001".</p>	<p><i>Пример использования параметра вывод для извлечения числа из записи в журнале:</i></p> <p>logrt[/app1/^test.*log\$,"task run [0-9].+sec, processed ([0-9]+) records, [0-9]+ errors","","\1"] → будет соответствовать записи в журнале "2015-11-13 10:08:26 task run 6.08 sec, processed 6080 records, 0 errors" и на сервер отправится только число 6080. Так как отправляется число, у этого элемента данных журнала можно изменить "Тип информации" с "Журнал (лог)" на "Числовой (целое положительное)" и это значение можно будет использовать на на графиках, в триггерах и т.д.</p> <p><i>Пример использования параметра вывод для изменения вывода записи журнала перед отправкой на сервер:</i></p> <p>logrt[/app1/^test.*log\$,"([0-9 :-]+) task run ([0-9.]+) sec, processed ([0-9]+) records, ([0-9]+) errors","",""\1 RECORDS: \3, ERRORS: \4, DURATION: \2"] → будет соответствовать записи в журнале "2015-11-13 10:08:26 task run 6.08 sec, processed 6080 records, 0 errors " и на сервер отправится измененная запись "2015-11-13 10:08:26 RECORDS:</p>

Ключ			
Описание	Возвращаемое значение	Параметры	Комментарии
		<p>не игнорировать строки в файлах журналов; > 0.0 - игнорировать более старые строки с целью получения наиболее новых строк проанализированных в течении</p> <p>“максзадержка” секунд. Перед использованием прочтайте заметки по максзадержка!</p> <p>опции - тип ротации файлов журналов.</p> <p>Возможные значения: <i>rotate</i> (по умолчанию), <i>copytruncate</i>.</p> <p>Обратите внимание, что <i>copytruncate</i> нельзя использовать вместе с <i>maxdelay</i>. В этом случае <i>maxdelay</i> должен быть равен 0 или не задан.</p> <p>Смотрите заметки по copytruncate.</p>	<p>6080, ERRORS: 0, DURATION: 6.08".</p> <p>Параметр режим поддерживается начиная с Zabbix 2.0.</p> <p>Параметр вывод поддерживается начиная с Zabbix 2.2.</p> <p>Параметр максзадержка поддерживается начиная с Zabbix 3.2.</p> <p>Параметр опции поддерживается начиная с Zabbix 4.0.</p> <p>Смотрите дополнительную информацию о мониторинге файлов журналов.</p>
Подсчёт количества совпадающих строк в мониторинге файла журнала (логов) с поддержкой ротации логов.	Целое число	<p>файл_regex - абсолютный путь к файлу и регулярное выражение описывающее его имя</p> <p>регулярное выражение - регулярное выражение описывающее требуемый шаблон содержимого</p> <p>кодировка - идентификатор кодовой страницы</p> <p>макс. кол-во строк - максимальное количество новых строк в секунду, которое агент будет анализировать.</p> <p>Значением по умолчанию является $10^* \text{MaxLinesPerSecond}$</p>	<p>Элемент данных должен быть настроен активной проверкой.</p> <p>Ротация журналов основывается на времени последнего изменения файлов.</p> <p>Смотрите дополнительную информацию о мониторинге файлов журналов.</p> <p>Параметр опции поддерживается начиная с Zabbix 4.0.</p> <p>Поддерживается начиная с Zabbix 3.2.0.</p>
logrt.count[файл_regex,<регулярное выражение>,<кодировка>,<макс. кол-во строк>,<режим>,<максзадержка>,<опции>]			

Ключ

Описание	Возвращаемое значение	Параметры	Комментарии
		<p>d' в zabbix_agentd.conf.</p> <p>режим - возможные значения: <i>all</i> (по умолчанию), <i>skip</i> - пропуск обработки старых данных (влияет только на недавно созданные элементы данных).</p> <p>максзадержка - максимальная задержка в секундах.</p> <p>Тип: число с плавающей точкой.</p> <p>Значения: 0 - (по умолчанию) никогда не игнорировать строки в файлах журналов; > 0.0 - игнорировать более старые строки с целью получения наиболее новых строк проанализированных в течении "максзадержка" секунд. Перед использованием прочтайте заметки по максзадержка!</p> <p>опции - тип ротации файлов журналов.</p> <p>Возможные значения: <i>rotate</i> (по умолчанию), <i>copytruncate</i>.</p> <p>Обратите внимание, что <i>copytruncate</i> нельзя использовать вместе с <i>maxdelay</i>. В этом случае <i>maxdelay</i> должен быть равен 0 или не задан.</p> <p>Смотрите заметки по copytruncate.</p>	

net.dns[<ip>,имя,<тип>,<время ожидания>,<количество>,<протокол>]

Проверка, работает ли DNS сервис.	0 - DNS не работает (сервер не ответил или ответ DNS некорректен)	ip - IP адрес DNS сервера (оставьте пустым, чтобы использовать DNS сервера по умолчанию, игнорируется в Windows)	Пример ключа: ⇒ net.dns[8.8.8.8,zabbix.com,MX,2,1]
	1 - DNS работает		Возможные значения для тип: ANY, A, NS, CNAME, MB, MG, MR, PTR, MD, MF, MX, SOA, NULL, WKS (исключая Windows), HINFO, MINFO, TXT, SRV

Описание	Возвращаемое значение	Параметры	Комментарии
		имя - проверяемое имя DNS тип - запрашиваемый тип записи (по умолчанию SOA) время ожидания (игнорируется в Windows) - время ожидания ответа в секундах (по умолчанию 1 секунда) количество (игнорируется в Windows) - количество попыток запросов (по умолчанию 2) протокол - используемый протокол при выполнении DNS запросов: <i>udp</i> (по умолчанию) или <i>tcp</i>	Национальные доменные имена не поддерживаются вместо них, пожалуйста, используйте имена в IDNA кодировке.
net.dns.record[<ip>,имя,<тип>,<время ожидания>,<количество>,<протокол>]			
Выполнение DNS запроса.	Строка символов ip - IP адрес DNS с требуемым типом информации	имя - проверяемое имя DNS тип - запрашиваемый тип записи (по умолчанию SOA) время ожидания (игнорируется в Windows) - время ожидания ответа в секундах (по умолчанию 1 секунда) количество (игнорируется в Windows) - количество попыток запросов (по умолчанию 2) протокол - используемый протокол при выполнении DNS запросов: <i>udp</i> (по умолчанию) или <i>tcp</i>	Пример ключа: ⇒ net.dns.record[8.8.8.8,zabbix.com,MX,2,1]
			Возможные значения для тип : ANY, A, NS, CNAME, MB, MG, MR, PTR, MD, MF, MX, SOA, NULL, WKS (исключая Windows), HINFO, MINFO, TXT, SRV
			Национальные доменные имена не поддерживаются вместо них, пожалуйста, используйте имена в IDNA кодировке.
			Пример ключа: ⇒ net.tcp.dns.query
			Параметр протокол поддерживается начиная с Zabbix 3.0. Тип записи SRV поддерживается в Zabbix агенте начиная с версий 1.8.6 (Unix) и 2.0.0 (Windows).
			Название до Zabbix 2.0 (еще поддерживается): <i>net.tcp.dns.query</i>

Описание	Возвращаемое значение	Параметры	Комментарии
Коллизии за пределами окна (out- Целое число of-window).	умолчанию) или <i>tcp</i> net.if.collisions[if]	if - имя сетевого интерфейса	Поддерживается Zabbix агентом начиная с версии 2.0.
Список сетевых интерфейсов. Используется низкоуровневым обнаружением.	Объект JSON	net.if.discovery	На FreeBSD, OpenBSD и NetBSD поддерживается начиная с версии Zabbix агента 2.2
Статистика по входящему трафику сетевого интерфейса.	Целое число	<p>net.if.in[if,<режим>]</p> <p>if - имя сетевого интерфейса (Unix); полное описание сетевого интерфейса или IPv4 адрес (Windows)</p> <p>режим - возможные значения:</p> <ul style="list-style-type: none"> <i>bytes</i> - количество байт (по умолчанию) <i>packets</i> - количество пакетов <i>errors</i> - количество ошибок <i>dropped</i> - количество отброшенных пакетов <i>overruns (fifo)</i> - количество ошибок FIFO буфера <i>frame</i> - количество ошибок кадрирования пакетов <i>compressed</i> - количество сжатых пакетов, полученных драйвером устройства <i>multicast</i> - количество кадров многоадресного вещания, которые получены драйвером устройства 	<p>В некоторых версиях Windows (например, Server 2008) может потребоваться установка последних обновлений для поддержки не-ASCII символов в именах интерфейсов.</p> <p>В Windows элемент данных берет значения с 64-битный счетчиков, если они доступны. 64-битные счетчики статистики по интерфейсам введены в Windows Vista и Windows Server 2008. Если 64-битный счетчики недоступны, агент будет использовать 32-битные счетчики.</p> <p>Начиная с Zabbix агента 1.8.6 на Windows поддерживаются мультибайтные имена интерфейса.</p> <p>Примеры: ⇒ net.if.in[eth0,errors] ⇒ net.if.in[eth0]</p> <p>Вы можете получить описания сетевых интерфейсов в Windows, используя элементы данных net.if.discovery или net.if.list.</p> <p>Вы можете использовать этот ключ с шагом предобработки <i>Изменение в секунду</i> для получения статистики байт в секунду.</p> <p>net.if.out[if,<режим>]</p>

Описание	Возвращаемое значение	Параметры	Комментарии
Статистика по исходящему трафику сетевого интерфейса.	Целое число	<p>if - имя сетевого интерфейса (Unix); полное описание сетевого интерфейса или IPv4 адрес (Windows)</p> <p>режим - возможные значения:</p> <ul style="list-style-type: none"> <i>bytes</i> - количество байт (по умолчанию) <i>packets</i> - количество пакетов <i>errors</i> - количество ошибок <i>dropped</i> - количество отброшенных пакетов <i>overruns (fifo)</i> - количество ошибок FIFO буфера <i>collisions (colls)</i> - количество коллизий обнаруженных на интерфейсе <i>carrier</i> - количество потерь несущей, обнаруженных драйвером устройства <i>compressed</i> - количество сжатых пакетов, переданных драйвером устройства <p>net.if.total[if,<режим>]</p>	<p>В Windows элемент данных берет значения с 64-битный счетчиков, если они доступны. 64-битные счетчики статистики по интерфейсам введены в Windows Vista и Windows Server 2008. Если 64-битный счетчики недоступны, агент будет использовать 32-битные счетчики.</p> <p>Начиная с Zabbix агента 1.8.6 на Windows поддерживаются мультибайтные имена интерфейса.</p> <p>Примеры: ⇒ net.if.out[eth0,errors] ⇒ net.if.out[eth0]</p> <p>Вы можете получить описания сетевых интерфейсов в Windows, используя элементы данных net.if.discovery или net.if.list.</p> <p>Вы можете использовать этот ключ с шагом предобработки <i>Изменение в секунду</i> для получения статистики байт в секунду.</p>
Суммарная статистика входящего и исходящего трафика на сетевом интерфейсе.	Целое число	<p>if - имя сетевого интерфейса (Unix); полное описание сетевого интерфейса или IPv4 адрес (Windows)</p> <p>режим - возможные значения:</p> <ul style="list-style-type: none"> <i>bytes</i> - количество байт (по умолчанию) <i>packets</i> - количество пакетов <i>errors</i> - количество ошибок <i>dropped</i> - количество отброшенных пакетов <i>overruns (fifo)</i> - количество ошибок FIFO буфера <i>collisions (colls)</i> - количество коллизий обнаруженных на интерфейсе <i>carrier</i> - количество потерь несущей, обнаруженных драйвером устройства <i>compressed</i> - количество сжатых пакетов, переданных драйвером устройства <p>net.if.total[if,<режим>]</p>	<p>В Windows элемент данных берет значения с 64-битный счетчиков, если они доступны. 64-битные счетчики статистики по интерфейсам введены в Windows Vista и Windows Server 2008. Если 64-битный счетчики недоступны, агент будет использовать 32-битные счетчики.</p> <p>Примеры: ⇒ net.if.total[eth0,errors] ⇒ net.if.total[eth0]</p> <p>Вы можете получить описания сетевых интерфейсов в Windows, используя элементы данных net.if.discovery или net.if.list.</p> <p>Вы можете использовать этот ключ с шагом предобработки <i>Изменение в секунду</i> для получения статистики байт в секунду.</p>

Описание	Возвращаемое значение	Параметры	Ключ
Проверка, находится ли TCP порт в состоянии LISTEN.	0 - не находится в состоянии LISTEN 1 - находится в состоянии LISTEN	порт - номер TCP порта	Обратите внимание, отброшенные пакеты поддерживаются только, если оба ключа net.if.in и net.if.out работают с отброщенными пакетами на вашей платформе. net.tcp.listen[порт]
Проверка, возможна ли создание TCP подключения на указанный номер порта.	0 - не удалось подключиться 1 - удалось подключиться	ip - IP адрес (по умолчанию 127.0.0.1) порт - номер порта	В Linux поддерживается Zabbix агентом начиная с версии 1.8.4 Начиная с Zabbix 3.0.0, в ядрах Linux 2.6.14 и более новых, информация о TCP сокетах в состоянии listen берется из интерфейса ядра NETLINK, если возможно. В противном случае, информация берется из файлов /proc/net/tcp и /proc/net/tcp6. net.tcp.port[<ip>,порт]
Проверка, запущен ли сервис и принимает ли он TCP подключения.	0 - сервис недоступен 1 - сервис запущен	сервис - один из: <i>ssh, ntp, ldap, smtp, ftp, http, pop, nntp, imap, tcp, https, telnet</i> (смотри детали) ip - IP адрес (по умолчанию 127.0.0.1) порт - номер порта (по умолчанию используется стандартный номер)	Пример: ⇒ net.tcp.port[,80] → можно использовать для проверки доступности веб-сервера, работающего на 80 порту. Для простого тестирования производительности TCP используйте net.tcp.service.perf[tcp,<ip>,<порт>] Обратите внимание, что эти проверки могут привести к дополнительным записям в системных файлах журналов (обычно сессии SMTP и SSH журналируются). Старое название: check_port[*] net.tcp.service[сервис,<ip>,<порт>]

Описание	Возвращаемое значение	Параметры	Ключ Комментарии
Проверка производительности TCP сервиса.	0 - сервис недоступен секунды - количество секунд потраченное на подключение к сервису	сервис - один из: <i>ssh, ntp, ldap, smtp, ftp, http, pop, nntp, imap, tcp, https, telnet</i> (смотри детали) ip - IP адрес (по умолчанию 127.0.0.1) порт - номер порта (по умолчанию используется стандартный номер порта сервиса)	Проверка шифрованных протоколов (таких как IMAP на 993 порту или POP на 995 порту) в настоящее время не поддерживается. Как решение, пожалуйста, для подобных проверок используйте net.tcp.port.
			Проверка LDAP и HTTPS через Windows агента в настоящее время не поддерживается.
			Обратите внимание, что telnet проверка ищет запрос на вход (с ':' в конце).
			Смотрите также известные проблемы по проверке HTTPS сервиса.
			Сервисы <i>https</i> и <i>telnet</i> поддерживается Zabbix с версии 2.0.
			Старое название: check_service[*]
		net.tcp.service.perf[сервис,<ip>,<порт>]	
			Пример: ⇒ net.tcp.service.perf[ssh] - можно использовать для тестирования скорости изначального ответа от SSH сервера.
			Проверка шифрованных протоколов (таких как IMAP на 993 порту или POP на 995 порту) в настоящее время не поддерживается. Как решение, пожалуйста, для подобных проверок используйте net.tcp.service.perf[tcp,<ip>,<порт>].
			Проверка LDAP и HTTPS через Windows агента в настоящее время не поддерживается.
			Обратите внимание, что telnet проверка ищет запрос на вход (с ':' в конце).
			Смотрите также известные проблемы по проверке HTTPS сервиса.
			Сервисы <i>https</i> и <i>telnet</i> поддерживается Zabbix с версии 2.0.
			Старое название:

Описание	Возвращаемое значение	Параметры	Комментарии
Проверка, находится ли UDP порт в состоянии LISTEN.	0 - не находится в состоянии LISTEN 1 - находится в состоянии LISTEN	порт - номер UDP порта	check_service_perf[*] net.udp.listen[порт] Например: ⇒ net.udp.listen[68]
Проверка, запущен ли сервис и принимает ли он UDP подключения.	0 - сервис недоступен 1 - сервис запущен	сервис - <i>ntp</i> (смотри детали) ip - IP адрес (по умолчанию 127.0.0.1) порт - номер порта (по умолчанию используется стандартный номер порта сервиса)	B Linux поддерживается Zabbix агентом начиная с версии 1.8.4 net.udp.service[сервис,<ip>,<порт>] Пример: ⇒ net.udp.service[ntp,,45] → можно использовать для тестирования доступности NTP сервиса на 45 порту UDP. Этот элемент данных поддерживается начиная с Zabbix 3.0.0, но <i>ntp</i> сервис был доступен в net.tcp.service[] элементе данных и в предыдущих версиях.
Проверка производительности UDP сервиса.	секунды - количество секунд потраченное на подключение к сервису	сервис - <i>ntp</i> (смотри детали) ip - IP адрес (по умолчанию 127.0.0.1) порт - номер порта (по умолчанию используется стандартный номер порта сервиса)	Этот элемент данных поддерживается начиная с Zabbix 3.0.0, но <i>ntp</i> сервис был доступен в net.tcp.service[] элементе данных и в предыдущих версиях. net.udp.service.perf[сервис,<порт>] Пример: ⇒ net.udp.service.perf[ntp] → можно использовать для тестирования времени ответа от NTP сервиса.
Использование CPU процесса в процентах.	Число с плавающей точкой	имя - имя процесса (по умолчанию <i>все процессы</i>) пользователь - имя пользователя (по умолчанию <i>все пользователи</i>) тип - тип использования CPU: <i>total</i> (по умолчанию), <i>user</i> , <i>system</i> cmdline - фильтр по командной строке (является регулярным выражением) режим - режим сбора данных: <i>avg1</i> (по умолчанию), <i>avg5</i> , <i>avg15</i>	Примеры: ⇒ proc.cpu.util[,root] → Использование CPU по всем процессам работающими под “root” пользователем ⇒ proc.cpu.util[zabbix_server,zabbix] → Использование CPU по всем процессам zabbix_server работающими под пользователем zabbix Возвращаемое значение основывается на использовании одного ядра CPU, в процентах. Например, утилизация CPU процесса полного использования двух ядер равна 200%. Данные использования CPU

Описание	Возвращаемое значение	Параметры	Ключ
			<p>собираются коллекторами, которые поддерживают максимально до 1024 уникальных (по имени, пользователю и командной строке) запросов. Запросы, к которым не было обращения в течении последних 24 часов, удаляются из коллектора.</p> <p><i>Обратите внимание</i>, что в случае когда агент скомпилирован на Solaris без поддержки зон, если задать параметр зона значением <i>current</i> (или по умолчанию), и запустить его на более новом Solaris, где зоны поддерживаются, тогда агент вернёт NOTSUPPORTED (агент не может ограничить результаты в пределах только текущей зоны). Однако, значение <i>all</i> в этом случае поддерживается.</p> <p>Этот ключ поддерживается начиная с Zabbix 3.0.0 и также доступен на нескольких платформах (смотри Поддерживаемые элементы данных по платформам).</p>
Количество памяти используемое процессом в байтах.	Целое число	<p>имя - имя процесса (по умолчанию <i>все процессы</i>) пользователь - имя пользователя (по умолчанию <i>все пользователи</i>) режим - возможные значения: <i>avg, max, min, sum</i> (по умолчанию) cmdline - фильтр по командной строке (является регулярным выражением) тип памяти - тип памяти используемый процессом.</p>	<p>proc.mem[<имя>,<пользователь>,<режим>,<cmdline>,<тип памяти>]</p> <p>Примеры:</p> <p>⇒ proc.mem[,root] - память используемая всеми процессами запущенными под пользователем "root"</p> <p>⇒ proc.mem[zabbix_server,zabbix] - память используемая всеми процессами zabbix_server запущенными под пользователем zabbix</p> <p>⇒</p> <p>proc.mem[,oracle,max,oracleZABBIX] - максимальное значение используемой памяти процессами запущенными под пользователем oracle и имеющими oracleZABBIX в содержимом командной строки</p> <p><i>Обратите внимание</i>: Когда несколько процессов используют разделяемую память, сумма памяти используемой процессами в результате может быть большой, нереальной величиной.</p> <p>Смотрите заметки по выбору процессов с параметрами ИМЯ и</p>

Описание	Возвращаемое значение	Параметры	Ключ Комментарии
Количество процессов.	Целое число	<p>proc.num[<имя>,<пользователь>,<состояние>,<cmdline>,<зона>]</p> <p>имя - имя процесса (по умолчанию “все процессы”) пользователь - имя пользователя (по умолчанию “все пользователи”) состояние - возможные значения: <i>all</i> (по умолчанию), <i>disk</i> - непрерывный сон, <i>run</i> - в процессе работы, <i>sleep</i> - прерываемый сон, <i>trace</i> - остановлен, <i>zomb</i> - зомби cmdline - фильтр по командной строке (является регулярным выражением) зона - целевая зона: <i>current</i> (по умолчанию), <i>all</i>. Этот параметр поддерживается только на Solaris платформе.</p>	<p> cmdline (специфика для Linux).</p> <p>Когда этот элемент данных вызывается с командной строки и содержит параметр командной строки (например, при использовании тестового режима агента: <code>zabbix_agentd -t proc.num[,, apache2]</code>), будет засчитан один дополнительный процесс, так как агент посчитает самого себя.</p> <p>Параметр тип памяти поддерживается на нескольких <u>платформах</u> начиная с Zabbix 3.0.0.</p> <p>Примеры ключей: ⇒ <code>proc.num[,mysql]</code> - количество процессов выполняемых под пользователем mysql ⇒ <code>proc.num[apache2,www-data]</code> - количество процессов apache2 выполняемых под пользователем www-data ⇒ <code>proc.num[,oracle,sleep,oracleZABBIX]</code> - количество процессов в спящем состоянии выполняемых под oracle и имеющих oracleZABBIX в содержимом командной строки</p> <p>Смотрите <u>заметки</u> по выбору процессов с параметрами Имя и cmdline (специфика для Linux).</p> <p>В Windows, поддерживаются только параметры Имя и Пользователь.</p> <p>Когда этот элемент данных вызывается с командной строки и содержит параметр командной строки (например, при использовании тестового режима агента: <code>zabbix_agentd -t proc.num[,, apache2]</code>), будет засчитан один дополнительный процесс, так как агент посчитает самого себя.</p> <p>Обратите внимание, что в случае когда агент скомпилирован на Solaris без поддержки зон, если задать</p>

Описание	Возвращаемое значение	Параметры	Ключ	Комментарии
Чтение аппаратного сенсора.	Число с плавающей точкой	устройство - имя устройства сенсор - имя сенсора режим - возможные значения: <i>avg, max, min</i> (если этот параметр не указан, то устройство и сенсор обрабатываются дословно).	sensor[устройство,сенсор,<режим>]	параметр зона значением <i>current</i> (или по умолчанию), и запустить его на более новом Solaris, где зоны поддерживаются, тогда агент вернёт NOTSUPPORTED (агент не может ограничить результаты в пределах только текущей зоны). Однако, значение <i>all</i> в этом случае поддерживается.
Штамп времени (timestamp) загрузки системы.	Целое число (UNIX timestamp)		system.boottime	Значения <i>disk</i> и <i>trace</i> для параметра state поддерживаются начиная с Zabbix 3.4.0.
Список найденных CPU/CPU ядер. Используется низкоуровневым обнаружением.	Объект JSON		system.cpu.discovery	Пример: ⇒ sensor[w83781d-i2c-0-2d,temp1]
Прерывания	Целое число		system.cpu.intr	До Zabbix 1.8.4, использовался формат sensor[temp1] . В Linux 2.6+, читается <i>/sys/class/hwmon</i> . Более подробное описаниесмотрите у элемента данных сенсора в Linux. В OpenBSD, читается MIB <i>hw.sensors</i> . Примеры: ⇒ sensor[cpu0,temp0] - температура одного CPU ⇒ sensor[cpu[0-2]\$temp,avg] - средняя температура первых трех CPU
				Поддерживается в OpenBSD Zabbix агентом начиная с версии 1.8.4.

Описание	Возвращаемое значение	Параметры	Комментарии
устройства.			
<u>Загрузка CPU.</u>	Число с плавающей точкой	<p>system.cpu.load[<cpu>,<режим>]</p> <p>cpu - возможные значения: <i>all</i> (по умолчанию), <i>percpi</i> (общая загрузка делится на количество CPU онлайн)</p> <p>режим - возможные значения: <i>avg1</i> (усреднение за одну минуту, по умолчанию), <i>avg5</i>, <i>avg15</i></p>	<p>Пример: ⇒ system.cpu.load[,avg5]</p> <p>Параметр <i>percpi</i> поддерживается начиная с Zabbix 2.0.0.</p> <p>Старое название: system.cpu.loadX</p>
Количество CPU.	Целое число	<p>system.cpu.num[<тип>]</p> <p>тип - возможные значения: <i>online</i> (по умолчанию), <i>max</i></p>	<p>Пример: ⇒ system.cpu.num</p>
Количество переключений контекста.	Целое число	system.cpu.switches	Старое название: system[switches]
Утилизация CPU в процентах.	Число с плавающей точкой	<p>system.cpu.util[<cpu>,<тип>,<режим>]</p> <p>cpu - номер CPU (по умолчанию все CPU)</p> <p>тип - возможные значения: <i>idle, nice, user</i> (по умолчанию), <i>system</i> (по умолчанию для Windows), <i>iowait, interrupt, softirq, steal, guest</i> (на ядре Linux 2.6.24 и выше), <i>guest_nice</i> (на ядре Linux 2.6.33 и выше).</p> <p>Параметры времени <i>user</i> и <i>nice</i> более не включают время <i>guest</i> и время <i>guest_nice</i> начиная с Zabbix 3.0.14, 3.4.5 и 4.0.0.</p> <p>режим - возможные значения: <i>avg1</i> (усреднение за одну минуту, по умолчанию), <i>avg5</i>, <i>avg15</i></p>	<p>Пример: ⇒ system.cpu.util[0,user,avg5]</p> <p>Старое название: system.cpu.idleX, system.cpu.niceX, system.cpu.systemX, system.cpu.userX</p>
		system.hostname[<тип>]	

Описание	Возвращаемое значение	Параметры	Ключ	Комментарии
Системное имя хоста.	Строка	тип (только для Windows, не должен использоваться на остальных системах) - возможные значения: <i>netbios</i> (по умолчанию) или <i>host</i>	Получение значения либо функцией GetComputerName() (для netbios), либо функцией gethostname() (для host) в Windows и с помощью команды “hostname” на других системах.	Примеры возвращаемых значений: в Linux: ⇒ system.hostname → linux-w7x1 ⇒ system.hostname → www.zabbix.com в Windows: ⇒ system.hostname → WIN-SERV2008-I6 ⇒ system.hostname[host] → Win-Serv2008-I6LonG
Информация о шасси.	Строка	информация - одно из full (по умолчанию), model, serial, type или vendor	Параметр ТИП поддерживается этим элементом данных начиная с версии 1.8.6 .	Смотрите также более детальную информацию .
Информация о CPU.	Строка или целое число	cpu - номер CPU или all (по умолчанию) информация - возможные значения: full (по умолчанию), curfreq, maxfreq, model или vendor	Требуются Права root , потому что значение читается из sysfs или памяти.	Поддерживается Zabbix агентом начиная с версии 2.0.
		system.hw.cpu[<cpu>,<информация>]	Пример: ⇒ system.hw.cpu[0,vendor] → AuthenticAMD	Информация собирается из /proc/cpuinfo и из /sys/devices/system/cpu/[cpunum]/cpufreq/cpuinfo_max_fr

Описание	Возвращаемое значение	Параметры	Ключ
Список PCI или USB устройств.	Текст	тип - pci (по умолчанию) или usb	eq. Если указаны номер CPU и curfreq или maxfreq, то возвращается числовое значение (Гц). Поддерживается Zabbix агентом начиная с версии 2.0.
Список MAC адресов.	Строка	интерфейс - all (по умолчанию) или регулярное выражение формат - full (по умолчанию) или short	system.hw.devices[<тип>] Пример: ⇒ system.hw.devices[pci] → 00:00.0 Host bridge: Advanced Micro Devices [AMD] RS780 Host Bridge [...] Возвращает вывод утилит lspci или lsusb (выполняются без каких либо параметров) Поддерживается Zabbix агентом начиная с версии 2.0.
Системное время.	Целое число - тип как utc Строка - с типом как local.	utc - (по умолчанию) время с начала Эпохи (00:00:00 UTC, Январь 1, 1970), измеренное в секундах. local - время в формате 'тггг-мм-дд,чч:мм:сс.ннн, +чч:мм'	system.hw.macaddr[<интерфейс>,<формат>] Список MAC адресов чьи имена интерфейсов совпадают с заданным в интерфейс регулярным выражением (all список всех интерфейсов). Пример: ⇒ system.hw.macaddr["eth0\$",full] ⇒ [eth0] 00:11:22:33:44:55 Если формат задан как short, имена интерфейсов и одинаковые MAC адреса не попадают в список. Поддерживается Zabbix агентом начиная с версии 2.0.
Выполнение указанной команды	Текст с результатом	команда - выполняемая команда	system.localtime[<тип>] Параметры для этого элемента данных поддерживаются начиная с версии 2.0. Например: ⇒ system.localtime[local] → создайте элемент данных, используя этот ключ, и далее используйте его для отображения времени узла сети в элементе комплексного экрана Часы .
			system.run[команда,<режим>] Возвращается до 512КБ (64КБ до Zabbix 2.0.5), включая пробелы в

Ключ

Описание	Возвращаемое значение	Параметры	Комментарии
на узле сети.	выполнения команды 1 - в режиме <i>nowait</i> (независимо от результата выполнения команды)	режим - один из <i>wait</i> (по умолчанию, ожидать завершения выполнения), <i>nowait</i> (не ожидать)	конце, которые отрезаются. Для корректной обработки, возвращаемые данные должны быть текстом. Пример: ⇒ system.run[ls -l /] → подробный список файлов в папке root. <i>Обратите внимание:</i> Для того чтобы включить этот функционал, в файле конфигурации агента должна быть указана опция EnableRemoteCommands=1.

system.stat[источник,<тип>]

Статистика системы. Целое число или число с плавающей точкой

ent - количество процессорных единиц выделенных на ресурс (с плавающей точкой)
kthr,<тип> - информация о состояниях потоков ядра:
r - среднее количество запускаемых потоков ядра (дробное)
b - среднее количество потоков ядра помещенных в очередь ожидания Менеджера Виртуальной Памяти (с плавающей точкой)
memory,<тип> - информация о использовании виртуальной и реальной памяти:
avm - активные виртуальные страницы (целое)
fre - размер свободного списка (целое)
page,<тип> - информация об ошибках страниц и активности страниц:
fi - операции страничного ввода файлов в секунду (с плавающей точкой)
fo - операции страничного вывода файлов в секунду (с плавающей точкой)
pi - pages paged in from paging space (с плавающей точкой)
po - pages paged out to paging space (с плавающей точкой)
fr - pages freed (page replacement) (с плавающей точкой)
sr - количество сканированных страниц алгоритмом замещения страниц (с плавающей точкой)
faults,<тип> - соотношение ловушек и прерываний:
in - прерывания устройства (с плавающей точкой)
sy - системные вызовы (с плавающей точкой)
cs - переключения контекста потока ядра (с плавающей

Ключ

Описание	Возвращаемое значение	Параметры	Комментарии
		точкой) scri,<тип> - раздельное использование процессорного времени в процентах: <i>us</i> - пользовательское время (с плавающей точкой) <i>sy</i> - системное время (с плавающей точкой) <i>id</i> - время простоя (с плавающей точкой) <i>wa</i> - время простоя в ходе которого система обрабатывала запросы I/O диска/NFS (с плавающей точкой) <i>pc</i> - количество использованных физических процессоров (с плавающей точкой) <i>ec</i> - процентное соотношение размеченного занятого пространства (с плавающей точкой) <i>lbusy</i> - показывает утилизация в процентах для логических процессор(ов), которая произошла при выполнении уровней пользователя и системы (с плавающей точкой) <i>app</i> - отображает количество доступных физических процессоров в распределенном пуле (с плавающей точкой) disk,<тип> - статистика о дисках: <i>bps</i> - отображает суммарное количество данных переданных (чтением или записью) на диск в байтах в секунду (целое) <i>tps</i> - показывает количество передач за секунду, которые произошли на физическом диске/ленте (с плавающей точкой) Этот элемент данных поддерживается начиная с версии 1.8.1 .	
Информация о архитектуре программного обеспечения.	Строка	system.sw.arch	Пример: ⇒ system.sw.arch → i686 Для получения информации используется функция <code>uname()</code> . Поддерживается Zabbix агентом начиная с версии 2.0.
Информация об операционной системе.	Строка	system.sw.os[<информация>] информация - возможные значения: full (по умолчанию), short или name	Информация получается с (обратите внимание, на то что не все файлы и опции представлены во всех дистрибутивах): <code>/proc/version (full)</code> <code>/proc/version_signature (short)</code> Параметр PRETTY_NAME из <code>/etc/os-release</code> на системах, которые его поддерживают, или <code>/etc/issue.net (name)</code> Поддерживается Zabbix агентом начиная с версии 2.0.
		system.sw.packages[<пакет>,<менеджер>,<формат>]	

Описание	Возвращаемое значение	Параметры	Комментарии
Список установленных пакетов.	Текст	<p>пакет - <i>all</i> (по умолчанию) или регулярное выражение</p> <p>менеджер - <i>all</i> (по умолчанию) или конкретный менеджер пакетов</p> <p>формат - <i>full</i> (по умолчанию) или <i>short</i></p>	<p>Список (в алфавитном порядке) установленных пакетов имена которых совпадают с регулярным выражением заданным в пакет (<i>all</i> списки всех).</p> <p>Пример: ⇒ system.sw.packages[mini,dpkg,short] → python-minimal, python2.6-minimal, ubuntu-minimal</p> <p>Поддерживаемые менеджеры пакетов (выполняемые команды): dpkg (dpkg --get-selections) pkgtool (ls /var/log/packages) rpm (rpm -qa) pacman (pacman -Q)</p> <p>Если указан формат равный <i>full</i>, пакеты группируются по менеджерам пакетов (каждый менеджер с отдельной строки, которая начинается с его имени заключенном в квадратные скобки). Если указан формат равный <i>short</i>, пакеты не группируются и отображаются одной строкой.</p>
Статистика файла подкачки в (с устройства в память).	Целое число	<p>устройство - устройство используемое для файла подкачки (по умолчанию <i>all</i>)</p> <p>тип - возможные значения: <i>count</i> (количество swapins), <i>sectors</i> (секторов swapped in), <i>pages</i> (страниц swapped in). Смотрите поддержку по платформам для получения деталей по умолчаниям.</p>	<p>Поддерживается Zabbix агентом начиная с версии 2.0.</p> <p>Пример: ⇒ system.swap.in[,pages]</p> <p>Источником этой информации является: /proc/swaps, /proc/partitions, /proc/stat (Linux 2.4) /proc/swaps, /proc/diskstats, /proc/vmstat (Linux 2.6)</p>
Статистика файла подкачки из (из памяти в устройство).	Целое число.	<p>устройство - устройство используемое для файла подкачки (по умолчанию <i>all</i>)</p>	<p>Пример: ⇒ system.swap.out[,pages]</p> <p>Источником этой информации является:</p>

Ключ			
Описание	Возвращаемое значение	Параметры	Комментарии
Размер файла подкачки в байтах или процентах от общего размера.	Целое число - для количества байт Число с плавающей точкой - для процентов.	тип - возможные значения: <i>count</i> (количество swapouts), <i>sectors</i> (секторов swapped out), <i>pages</i> (страниц swapped out). Смотрите поддержку по платформам для получения деталей по умолчаниям.	/proc/swaps, /proc/partitions, /proc/stat (Linux 2.4) /proc/swaps, /proc/diskstats, /proc/vmstat (Linux 2.6)
		устройство - устройство используемое для файла подкачки (по умолчанию <i>all</i>) тип - возможные значения: <i>free</i> (свободно в файле подкачки, по умолчанию), <i>pfree</i> (свободно в файле подкачки, в процентах), <i>pused</i> (используемое место в файле подкачки, в процентах), <i>total</i> (полный размер файла подкачки), <i>used</i> (используется места в файле подкачки)	Пример: ⇒ system.swap.size[,pfree] → свободно места в файле подкачки в процентах Если <i>устройство</i> не указано, Zabbix агент будет брать во внимание только swap устройства (файлы), физическая память будет игнорироваться. Например, на Solaris системах команда <i>swap -s</i> включает часть физической памяти и swap устройства (в отличии от <i>swap -l</i>). Обратите внимание, этот ключ может возвращать некорректные данные на виртуализированных (VMware ESXi, VirtualBox) Windows платформах. В этом случае используйте ключ <i>perf_counter[700(_Total)\702]</i> для получения корректного использования файла подкачки в процентах.
Идентификация системы.	Строка	system.uname	Старое название: system.swap.free, system.swap.total
			Пример возвращаемого значения (Unix): FreeBSD localhost 4.2-RELEASE FreeBSD 4.2-RELEASE #0: Mon Nov i386
			Пример возвращаемого значения (Windows): Windows ZABBIX-WIN 6.0.6001 Microsoft® Windows Server® 2008 Standard Service Pack 1 x86
			В Unix начиная с Zabbix 2.2.0 значение этого элемента извлекается

Описание	Возвращаемое значение	Параметры	Ключ	Комментарии
				при помощи системного вызова <code>uname()</code> . Ранее эта информация запрашивалась с помощью системного вызова “ <code>uname -a</code> ”. Значение этого элемента данных может отличаться от вывода “ <code>uname -a</code> ” и может не включать дополнительную информацию, так как “ <code>uname -a</code> ” выводит информацию основываясь на других источниках.
				В Windows начиная с Zabbix 3.0 значение этого элемента данных извлекается при помощи <code>Win32_OperatingSystem</code> и <code>Win32_Processor</code> классов WMI. Ранее эта информация запрашивалась при помощи непостоянных вызовов Windows API и недокументированных ключей реестра. Имя ОС (включая редакцию) может быть переведено на пользовательский язык интерфейса. Некоторые версии Windows содержат символы товарных знаков и дополнительные пробелы.
				Обратите внимание, что этот элемент данных в Windows возвращает архитектуру ОС, тогда как в Unix он возвращает архитектуру CPU.
			system.uptime	
Время работы в секундах.	Целое число			При настройке элемента данных , используйте единицы измерения <code>s</code> или <code>uptime</code> для получения читаемых значений.
Количество пользователей находящихся в системе.	Целое число		system.users.num	Для получения значений используется команда <code>who</code> на стороне агента.
Статистика чтения диска.	Целое число при устройство - дисковое устройство (по <code>sectors</code> , <code>operations</code> , <code>bytes</code> умолчанию <code>all</code>) Число с плавающей точкой при тип равном: <code>sps</code> , <code>ops</code> , <code>bps</code>	vfs.dev.read[<устройство>,<тип>,<режим>]	Значения по умолчанию параметра 'Тип' для различных ОС: AIX - operations FreeBSD - bps Linux - sps OpenBSD - operations Solaris - bytes	Пример: ⇒ <code>vfs.dev.read[,operations]</code>

Описание	Возвращаемое значение	Параметры	Ключ	Комментарии
Статистика записи на диск.	умолчания отличаются для разных ОС. <i>sps, ops, bps</i> соответствуют: секторам, операциям, байтам в секунду соответственно режим - возможные значения: <i>avg1</i> (усреднение за минуту, по умолчанию), <i>avg5</i> (усреднение за 5 минут), <i>avg15</i> (усреднение за 15 минут). Третий параметр поддерживается только, если ТИП один из: <i>sps, ops, bps</i> .	умолчания отличаются для разных ОС. <i>sps, ops, bps</i> соответствуют: секторам, операциям, байтам в секунду соответственно режим - возможные значения: <i>avg1</i> (усреднение за минуту, по умолчанию), <i>avg5</i> (усреднение за 5 минут), <i>avg15</i> (усреднение за 15 минут). Третий параметр поддерживается только, если ТИП один из: <i>sps, ops, bps</i> .	<i>ops, bps и sps</i> на поддерживаемых платформах ограничено 8 устройствами (7 отдельных устройств и одно <i>all</i>). Начиная с Zabbix 2.0.1 этот лимит увеличен до 1024 (1023 отдельных устройств и один для <i>all</i>).	По умолчанию используется <i>all</i> первым параметром, который возвращает суммарную статистику, включая: все блочные устройства такие как <i>sda, sdb</i> и их разделы <i>sda1, sda2, sdb3</i> ... и несколько устройств (MD raid) на основе этих блочных устройств/разделов и логические разделы (LVM) на основе этих блочных устройств/разделов. В некоторых случаях возвращаемые значения следует рассматривать как относительные значения (изменяющиеся во времени), но не как абсолютные значения. LVM поддерживается начиная с Zabbix 1.8.6.
	vfs.dev.write[<устройство>,<тип>,<режим>]		До Zabbix 1.8.6, можно было использовать только относительные имена устройств (например, sda), начиная с 1.8.6 можно использовать optionalный префикс /dev/ (например, /dev/sda)	Старое название: <i>io[*]</i>

Ключ			
Описание	Возвращаемое значение	Параметры	Комментарии
		<p>режим - возможные значения: <code>avg1</code> (усреднение за минуту, по умолчанию), <code>avg5</code> (усреднение за 5 минут), <code>avg15</code> (усреднение за 15 минут).</p> <p>Третий параметр поддерживается только если ТИП один из: <code>sps</code>, <code>ops</code>, <code>bps</code>.</p>	<p>По умолчанию используется <code>all</code> первым параметром, который возвращает суммарную статистику, включая: все блочные устройства такие как <code>sda</code>, <code>sbd</code> и их разделы <code>sda1</code>, <code>sda2</code>, <code>sdb3</code> ... и несколько устройств (MD raid) на основе этих блочных устройств/разделов и логические разделы (LVM) на основе этих блочных устройств/разделов. В некоторых случаях возвращаемые значения следует рассматривать как относительные значения (изменяющиеся во времени), но не как абсолютные значения.</p> <p>LVM поддерживается начиная с Zabbix 1.8.6.</p> <p>До Zabbix 1.8.6, можно было использовать только относительные имена устройств (например, <code>sda</code>), начиная с 1.8.6 можно использовать optionalный префикс <code>/dev/</code> (например, <code>/dev/sda</code>)</p> <p>Старое название: <code>io[*]</code></p>
<code>vfs.dir.count[директория,<regex_вкл>,<regex_искл>,<типы_вкл>,<типы_искл>,<макс_глубина>,<мин_размер>,<макс_размер>,<мин_возраст>,<макс_возраст>]</code>	Количество записей в директории	<p>директория - абсолютный путь к директории</p> <p>regex_вкл - регулярное выражение описывающее включаемые файл, директорию и символическую ссылку (если пусто, включаются все файлы, директории и символические ссылки; пустая строка является значением по умолчанию)</p> <p>regex_искл - регулярное выражение описывающее исключаемые файл, директорию и символическую ссылку (если пусто,</p>	<p>Переменные сред, такие как <code>%APP_HOME%</code>, <code>\$HOME</code> и <code>%TEMP%</code>, не поддерживаются.</p> <p>Псевдо-директории <code>."</code> и <code>..</code> никогда не учитываются.</p> <p>Символические ссылки никогда проверяются для обхода директорий.</p> <p>В Windows символические ссылки директорий пропускаются и жесткие ссылки учитываются лишь один раз.</p> <p><code>regex_incl</code> и <code>regex_excl</code> являются Perl совместимыми регулярными выражениями (PCRE). Оба параметра <code>regex_вкл</code> и <code>regex_искл</code> применяются к файлам и директориям при вычислении общего размера, но игнорируются при выборе поддиректорий (если <code>regex_вкл</code> равен <code>"(?i)^.+\.zip\$"</code> и <code>макс_глубина</code> не задана, тогда проход будет выполнен</p>
Целое число			

Ключ

Описание	Возвращаемое значение	Параметры	Комментарии
		<p>ничего не исключать; пустая строка является значением по умолчанию)</p> <p>типы_вкл - перечисление типов записей в директории, которые будут подсчитываться, возможные значения: <i>file</i> - обычный файл, <i>dir</i> - поддиректория, <i>sym</i> - символьическая ссылка, <i>sock</i> - соект, <i>bdev</i> - блочное устройство, <i>cdev</i> - устройство посимвольного ввода-вывода, <i>fifo</i> - FIFO, <i>dev</i> - синонимы к “<i>bdev,cdev</i>”, <i>all</i> - все вышеупомянутые типы, т.е. “<i>file,dir,sym,sock,bdev,cdev,fifo</i>”. Это значение по умолчанию, если параметр оставлен пустым. Несколько типов необходимо разделять запятой и весь список нужно заключать в кавычки “”.</p> <p>типы_искл - перечисление типов записей в директории, которые НЕ будут подсчитываться, те же самые значения и синтаксис что и для типы_вкл. Если некоторые типы записей указаны как в типы_вкл, так и в типы_искл, записи этих типов в директории НЕ будут подсчитываться.</p> <p>макс_глубина - максимальная глубина сканируемых поддиректорий. -1 (по умолчанию) - без ограничения, 0 - без</p>	<p>по всем поддиректориям, но учтены будут только файлы с типом zip). Если имя файла совпадает как с regex_вкл, так и с regex_искл, такой файл не будет учитываться.</p> <p>Время выполнения ограничено значением времени ожидания по умолчанию, которое равно 3 секундам (параметр “Timeout” в файле конфигурации агента). Та как обход большой директории может занять более длительное время чем время ожидания, тогда никакие данные не возвращаются и элемент данных отметится как “Не поддерживается”. Частичный подсчет не возвращается.</p> <p>При фильтрации по размеру только обычные файлы имеют осмысленные размеры. В Linux и BSD директории также имеют не нулевые размеры (в основном несколько Кб). Устройства имеют нулевые размеры, например, размер /dev/sda1 не влияет на соответствующий размер раздела. Поэтому, при использовании параметров <мин_размер> и <макс_размер> рекомендуется указать <типы_вкл> равным “<i>file</i>”, чтобы избежать сюрпризов.</p> <p>Примеры: ⇒ vfs.dir.count[/dev] - мониторинг количества устройств в /dev (Linux) ⇒ vfs.dir.count["C:\Users\ADMINI~1\AppData\Local\Temp"] - мониторинг количества файлов во временной директории (Windows)</p> <p>Поддерживается начиная с Zabbix 4.0.0.</p>

Описание	Возвращаемое значение	Параметры	Комментарии
Размер директории (в байтах).	Целое число	<p>проверки нижестоящих поддиректорий.</p> <p>мин_размер - минимальный размер подсчитываемых файлов. Файлы, которые меньше этого значения не будут подсчитаны. Значение в байтах. Можно использовать суффиксы памяти.</p> <p>макс_размер - максимальный размер подсчитываемых файлов. Файлы, которые больше этого значения не будут подсчитаны. Значение в байтах. Можно использовать суффиксы памяти.</p> <p>мин_возраст - минимальный возраст подсчитываемой записи директории. Записи, которые изменены раньше не будут подсчитаны. Целочисленное значение в виде секунд. Можно использовать суффиксы времени.</p> <p>макс_возраст - максимальный возраст подсчитываемой записи директории. Записи, которые настолько старые не будут подсчитаны (время модификации). Целочисленное значение в виде секунд. Можно использовать суффиксы времени.</p>	<p>Подсчитываются только те директории к которым zabbix пользователь имеет по крайней мере права чтения.</p>

vfs.dir.size[директория,<regex_вкл>,<regex_искл>,<режим>,<макс_глубина>]

Размер директории
(в байтах).

Целое число

директория - абсолютный путь к директории
regex_вкл -

Подсчитываются только те директории к которым zabbix пользователь имеет по крайней мере права чтения.

Ключ

Описание	Возвращаемое значение	Параметры	Комментарии
	регулярное вырыжение описывающее включаемые файл, директорию и символическую ссылку (если пусто, включаются все файлы, директории и символические ссылки; пустая строка является значением по умолчанию)	regex_искл -	В Windows любая символьическая ссылка пропускается и жесткие ссылки принимаются во внимание только один раз.
	регулярное вырыжение описывающее исключаемые файл, директорию и символическую ссылку (если пусто, ничего не исключать; пустая строка является значением по умолчанию)		При наличии больших директорий или медленных дисков этот элемент данных может превысить время ожидания из-за настройки Timeout в файлах конфигурации агента и сервера/прокси . При необходимости увеличьте эти значения времени ожидания.
	режим - возможные значения: <i>apparent</i> (по умолчанию) - получение реальных размеров файлов вместо использования диска (работает как du -sb директория), <i>disk</i> - получение использования диска (работает как du -s -B1 директория). В отличии от du команды, элемент данных vfs.dir.size принимает во внимание скрытые файлы при вычислении размера директории (работает как du -sb .[^.]* * в пределах директории).	режим - возможные значения: <i>apparent</i> (по умолчанию) - получение реальных размеров файлов вместо использования диска (работает как du -sb директория), <i>disk</i> - получение использования диска (работает как du -s -B1 директория). В отличии от du команды, элемент данных vfs.dir.size принимает во внимание скрытые файлы при вычислении размера директории (работает как du -sb .[^.]* * в пределах директории).	Примеры: ⇒ vfs.dir.size[/tmp,log] - вычисление размеров всех файлов в /tmp, которые содержат в своём имени 'log' ⇒ vfs.dir.size[/tmp,log,^.+\old\$] - вычисление размеров всех файлов в /tmp, которые содержат в своём имени 'log', исключая файлы содержащие в своем имени '.old'
	макс_глубина - максимальная глубина сканируемых поддиректорий. -1 (по умолчанию) - без	макс_глубина -	Ограничение размера файлов зависит от наличия поддержки больших файлов .
			Поддерживается начиная с Zabbix 3.4.0.

Ключ			
Описание	Возвращаемое значение	Параметры	Комментарии
Контрольная сумма файла, вычисленная по алгоритму используемому в UNIX cksum.	Целое число	ограничения, 0 - без проверки нижестоящих поддиректорий. vfs.file.cksum[файл]	Пример: ⇒ vfs.file.cksum[/etc/passwd] Пример возвращаемого значения: 1938292000 Старое название: cksum Ограничение размера файла зависит от поддержки больших файлов .
Получение содержимого файла.	Текст	файл - абсолютный путь к файлу кодировка - идентификатор кодовой страницы	Возвращает пустую строку, если файл пустой или содержит только LF/CR символы. Пример: ⇒ vfs.file.contents[/etc/passwd] Этот элемент данных ограничен файлами не превышающими 64 КБайт.
Проверка существования файла	0 - файл не найден 1 - обычный файл или ссылка (символическая или жесткая) на обычный файл существует.	файл - абсолютный путь к файлу	Поддерживается Zabbix агентом начиная с версии 2.0. Пример: ⇒ vfs.file.exists[/tmp/application.pid] Возвращаемое значение зависит от того, что вернет S_ISREG POSIX макрос.
MD5 контрольная сумма файла	Строка (MD5 хэш файла)	файл - абсолютный путь к файлу	Ограничение размера файла зависит от поддержки больших файлов . Пример: ⇒ vfs.file.md5sum[/usr/local/etc/zabbix_agentd.conf] Пример возвращаемого значения: b5052decb577e0fffd622d6ddc017e82 Ограничение (64МБ) на размер файла для этого элемента данных удалено в версии 1.8.6.

Ключ			
Описание	Возвращаемое значение	Параметры	Комментарии
Поиск строки в файле.	<p>vfs.file-regexp[файл,регулярное выражение,<кодировка>,<начальная строка>,<конечная строка>,<вывод>]</p> <p>файл - абсолютный путь к файлу</p> <p>регулярное выражение - Perl совместимое регулярное выражение (PCRE) или расширенное POSIX регулярное выражение до Zabbix 3.4</p> <p>кодировка - идентификатор кодовой страницы</p> <p>начальная строка - номер первой строки для поиска (по умолчанию, первая строка файла).</p> <p>конечная строка - номер последней строки для поиска (по умолчанию, последняя строка файла).</p> <p>вывод - дополнительный шаблон форматирования вывода.</p> <p>\0 - управляющая последовательность заменяется найденным текстом, тогда \N (где N=1..9) - управляющая последовательность заменяется N-нной совпадающей группой (или пустой строкой, если N превышает количество найденных групп).</p>	<p>Возвращается только первая совпадающая строка.</p> <p>Будет возвращена пустая строка, если не найдено совпадений с выражением.</p> <p>Извлечение содержимого при использовании параметра вывод выполняется агентом.</p> <p>Параметры начальная строка, конечная строка и вывод поддерживаются начиная с версии 2.2.</p> <p>Примеры:</p> <p>⇒ 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] → получение ID пользователя zabbix</p>	Ограничение размера файла зависит от поддержки больших файлов .
Поиск строки в файле.	Строка содержащая совпадающую подстроку или то, что определено в дополнительном параметре вывод .		
Поиск строки в файле.	0 - совпадение не найдено	файл - абсолютный путь к файлу	Параметры начальная строка , конечная строка поддерживаются с Zabbix 2.2.

Описание	Возвращаемое значение	Параметры	Комментарии
	1 - найдено	<p>выражение - Perl совместимое регулярное выражение (PCRE) или расширенное POSIX регулярное выражение до Zabbix 3.4</p> <p>кодировка - идентификатор кодовой страницы</p> <p>начальная строка - номер первой строки для поиска (по умолчанию, первая строка файла).</p> <p>конечная строка - номер последней строки для поиска (по умолчанию, последняя строка файла).</p> <p>vfs.file.size[файл]</p>	Например: ⇒ vfs.file.regmatch[/var/log/app.log,error]
Размер файла (в байтах).	Целое число	файл - абсолютный путь к файлу	Пользователь zabbix должен иметь права на чтение файла
Информация о времени файла.	Целое число (Unix timestamp)	<p>vfs.file.time[файл,<режим>]</p> <p>файл - полный путь к файлу</p> <p>режим - возможные значения: <i>modify</i> (по умолчанию) - время последней модификации содержимого файла, <i>access</i> - время последнего чтения файла, <i>change</i> - время последнего изменения свойств файла</p>	<p>Например: ⇒ vfs.file.size[/var/log/syslog]</p> <p>Ограничение размера файла зависит от поддержки больших файлов.</p>
Список примонтированных файловых систем. Используется низкоуровневым обнаружением.	JSON объект	vfs.fs.discovery	<p>Поддерживается Zabbix агентом начиная с версии 2.0.</p> <p>Макрос {#FSDRIVETYPE} поддерживается в Windows начиная с Zabbix агента версии 3.0.</p>

Ключ

Описание	Возвращаемое значение	Параметры	Комментарии
Количество или процент inodes.	Целое число - для количества Число с плавающей точкой - для процентов	vfs.fs.inode[fs,<режим>] fs - файловая система режим - возможные значения: <i>total</i> (по умолчанию), <i>free</i> , <i>used</i> , <i>pfree</i> (свободно, в процентах), <i>pused</i> (использовано, в процентах)	Пример: ⇒ <code>vfs.fs.inode[/,pfree]</code> Старое название: <i>vfs.fs.inode.free[*]</i> , <i>vfs.fs.inode.pfree[*]</i> , <i>vfs.fs.inode.total[*]</i>
Размер диска в байтах или процентах от общего размера.	Целое число - для байт Число с плавающей точкой - для процентов.	vfs.fs.size[fs,<режим>] fs - файловая система режим - возможные значения: <i>total</i> (по умолчанию), <i>free</i> , <i>used</i> , <i>pfree</i> (доступно, в процентах), <i>pused</i> (использовано, в процентах)	В случае примонтированного раздела, будет возвращен размер диска локального файла системы. Пример: ⇒ <code>vfs.fs.size[/tmp,free]</code> Зарезервированное место на файловой системе принимается во внимание и не включено при использовании режима <i>free</i> .
Размер памяти в байтах или в процентах от общего количества.	Целое число - для байт Число с плавающей точкой - для процентов.	vm.memory.size[<режим>] режим - возможные значения: <i>total</i> (по умолчанию), <i>active</i> , <i>anon</i> , <i>buffers</i> , <i>cached</i> , <i>exec</i> , <i>file</i> , <i>free</i> , <i>inactive</i> , <i>pinned</i> , <i>shared</i> , <i>slab</i> , <i>wired</i> . (использовано, в процентах), <i>available</i> , <i>available</i> , <i>available</i> (доступно, в процентах)	Старые названия: <i>vfs.fs.free[*]</i> , <i>vfs.fs.total[*]</i> , <i>vfs.fs.used[*]</i> , <i>vfs.fs.pfree[*]</i> , <i>vfs.fs.pused[*]</i> Этот элемент данных принимает три категории параметров:
Получение содержимого веб-страницы.	Исходная веб-страница как текст (включая заголовки)	web.page.get[хост,<путь>,<порт>] хост - имя хоста путь - путь к HTML документу (по умолчанию /) порт - номер порта	1) <i>total</i> - общего количества памяти. 2) специфичные для платформ типы памяти: <i>active</i> , <i>anon</i> , <i>buffers</i> , <i>cached</i> , <i>exec</i> , <i>file</i> , <i>free</i> , <i>inactive</i> , <i>pinned</i> , <i>shared</i> , <i>slab</i> , <i>wired</i> . 3) оценка на уровне пользователя как много памяти используется и доступно: <i>used</i> , <i>pused</i> , <i>available</i> , <i>available</i> . Смотрите более подробное описание <u>параметров vm.memory.size</u> . Старые названия: <i>vm.memory.buffers</i> , <i>vm.memory.cached</i> , <i>vm.memory.free</i> , <i>vm.memory.shared</i> , <i>vm.memory.total</i>

Ключ			
Описание	Возвращаемое значение	Параметры	Комментарии
Время полной загрузки веб-страницы (в секундах).	Число с плавающей точкой	(по умолчанию 80) web.page.perf[хост,<путь>,<порт>] хост - имя хоста путь - путь к HTML документу (по умолчанию /) порт - номер порта (по умолчанию 80)	web.page.get[www.zabbix.com,index.php,80] Возвращает 0 при ошибке. Например: ⇒ web.page.perf[www.zabbix.com,index.php,80]
Поиск строки на веб-странице.	Совпадающая строка, или как указан опциональный параметр ВЫВОД	web.page regexp[хост,<путь>,<порт>,регулярное выражение,<длина>,<вывод>] хост - имя хоста путь - путь к HTML документу (по умолчанию - /) порт - номер порта (по умолчанию - 80) регулярное выражение - <u>Perl совместимое регулярное выражение</u> (PCRE) или расширенное POSIX регулярное выражение до Zabbix 3.4 длина - максимальное количество возвращаемых символов ВЫВОД - дополнительный шаблон форматирования вывода. \0 - управляющая последовательность заменяется найденным текстом, тогда \N (где N=1..9) - управляющая последовательность заменяется N-нной совпадающей группой (или пустой строкой, если N превышает количество найденных групп).	Возвращается пустая строка, если нет совпадений или при ошибке. Извлечение содержимого согласно дополнительного параметра ВЫВОД осуществляется агентом. Параметр ВЫВОД поддерживается начиная с версии 2.2. Пример: ⇒ web.page.regexp[www.zabbix.com,index.php,80,OK,2]

Заметка специфики для Linux. Zabbix агент должен иметь только права чтения файловой системы /proc. Патчи к ядру для ограничения прав непrivилегированных пользователей вы можете найти на странице www.grsecurity.org.

Доступные кодировки

Параметр Кодировка используется для того, чтобы указать кодировку при обработке соответствующих проверок элементов данных, так чтобы полученные данные не были повреждены. Для получения списка поддерживаемых кодировок (идентификаторов кодовых страниц), пожалуйста, обратитесь к соответствующей документации, такой как libiconv (GNU Project) или Microsoft Windows SDK документации по “Идентификаторам кодовых страницы”.

Если задана пустая Кодировка, тогда по умолчанию используются UTF-8 (языковой стандарт по умолчанию для новых Unix/Linux дистрибутивов, смотрите настройки вашей системы) или ANSI с определенным расширением в системе (Windows).

Поиск проблем с элементами данных агента

- Если используются с пассивным агентом, значение *Timeout* конфигурации сервера, возможно, потребуется выше, чем *Timeout* в файле конфигурации агента. В противном случае элемент данных может не возвращать никаких значений по причине того, что запрос сервера к агенту превысит время ожидания раньше агента.

Специфичные ключи элементов данных для Windows

Ключи элементов данных

В таблице приводится подробная информация о ключах элементов данных, которые вы можете использовать только с Zabbix Windows агентом.

		Ключ	
▲	Описание	Возвращаемое значение	Параметры
	Мониторинг журналов событий.	<code>eventlog[имя,<регулярное выражение>,<важность>,<источник>,<eventid>,<макс. кол-во строк>,<режим>]</code>	<p>имя - имя журнала событий регулярное выражение - регулярное выражение описывающее требуемый шаблон содержимого важность - регулярное выражение описывающее важность Параметр может принимать следующие значения: “Information”, “Warning”, “Error”, “Critical”, “Verbose” (начиная с Zabbix 2.2, работающих на Windows Vista или на более новых версиях) источник - регулярное выражение, описывающее идентификатор источника (регулярное выражение</p> <p>Элемент данных должен быть настроен активной проверкой.</p> <p>Примеры: ⇒ <code>eventlog[Application]</code> ⇒ <code>eventlog[Security,,,"Failure Audit",,529 680]</code> ⇒ <code>eventlog[System,,,"Warning Error"]</code> ⇒ <code>eventlog[System,,,^1\$]</code> ⇒ <code>eventlog[System,,,@TWOSHORT]</code> - здесь используется ссылка на пользовательское регулярное выражение с именем TWOSHORT (заданное с типом Результат ИСТИНА, само выражение равно ^1\$ ^70\$).</p> <p><i>Обратите внимание</i>, агент не может отправлять события из “Пересланные события” журнала.</p> <p>Параметр режим</p>

Описание	Возвращаемое значение	Параметры	Комментарии
Список сетевых интерфейсов (включая тип, состояние, IPv4 адрес, описание интерфейса).	текст	net.if.list поддерживается начиная с версии Zabbix 2.2.0) eventid - регулярное выражение описывающее идентификатор(ы) событий макс. кол-во строк - максимальное количество новых строк в секунду, которое агент будет отправлять Zabbix серверу или прокси. Этот параметр заменяет значение 'MaxLinesPerSecond' в zabbix_agentd.win.conf режим - возможные значения: <i>all</i> (по умолчанию), <i>skip</i> - пропустить обработку старых данных (влияет только на недавно созданные элементы данных).	поддерживается начиная с версии 2.0.0. “Windows Eventing 6.0” поддерживается начиная с Zabbix 2.2.0. Обратите внимание, что выбор не журнального типа информации для этого элемента данных приведет к потере локального штампа времени, а также важности журнала и информации о источнике.
Значение любого счетчика	Целое число, число с	perf_counter[счетчик,<период>] счетчик - путь к счетчику	Поддерживается Zabbix агентом начиная с версии 1.8.1. Начиная с версии 1.8.6 Zabbix агента поддерживаются мультибайтные имена интерфейса. Отключенные интерфейсы не входят в список. Обратите внимание, что включение/отключение некоторых компонентов Windows могут изменить порядок имён интерфейсов в Windows.
			В некоторых версиях Windows (к примеру, Server 2008) может потребоваться установка последних обновления для поддержки не-ASCII символов в именах интерфейсов.

	Описание	Возвращаемое значение	Параметры	Комментарии
▲	производительности Windows.	плавающей точкой, строка или текст (в зависимости от запроса)	период - последние N секунд для сохранения усредненного значения.	производительности для получения списка счетчиков. До версии 1.6 этот параметр возвращал правильное значение только для счетчиков, которые возвращают только одно значение (например, \System\Threads). Параметр не будет работать со счетчиками, которые возвращают более одного значения - например утилизация CPU. Начиная с версии 1.6 используется период, такая проверка каждый раз возвращает среднее значение за последние "период" секунд.
	Различная информация о указанном процессе(ах).	Число с плавающей точкой	Значение период должно быть равно значению с 1 до 900 секунд (включительно), значение по умолчанию 1.	Смотрите также: Счетчики производительности в Windows .
proc_info[<процесс>,<атрибут>,<тип>]				
		<процесс> - имя <атрибут> - запрашиваемый атрибут процесса. <тип> - тип представления (имеет смысл, когда есть более одного процесса с одним именем)	V на настоящий момент поддерживаются следующие атрибуты: <i>vmsize</i> - размер виртуальной памяти процесса в Кбайтах <i>wkset</i> - размер working set процесса (количество физической памяти используемой процессом) в Кбайтах <i>pf</i> - Количество ошибок на страницах <i>ktimes</i> - время ядра процесса в миллисекундах <i>utime</i> - пользовательское время процесса в миллисекундах <i>io_read_b</i> - количество байт чтения процессом в процессе I/O операций <i>io_read_op</i> - количество операций чтения выполненных процессом <i>io_write_b</i> - количество байт записи процессом в процессе I/O операций <i>io_write_op</i> - количество операций записи выполненных процессом <i>io_other_b</i> - количество байт переданных процессу в течении операций отличных от чтения и записи	

Описание	Возвращаемое значение	Параметры	Комментарии
<p>▲</p> <p>Ключ</p> <p>Описание</p> <p>Возвращаемое значение</p> <p>Параметры</p> <p>Комментарии</p> <p><i>io_other_op</i> - количество I/O операций выполненных процессов, отличных от операций чтения и записи <i>gdiobj</i> - количество объектов GDI используемых процессом <i>userobj</i> - количество объектов USER используемых процессом</p> <p>Допустимые типы: <i>min</i> - минимальное значение среди всех процессов с именем <процесс> <i>max</i> - максимальное значение среди всех процессов с именем <процесс> <i>avg</i> - среднее значение среди всех процессов с именем <процесс> <i>sum</i> - сумма значений для всех процессов с именем <процесс></p> <p>Примеры: ⇒ proc_info[iexplore.exe,wkset,sum] - для получения общего количества физической памяти выделенной под все процессы Internet Explorer ⇒ proc_info[iexplore.exe,pf,avg] - для получения среднего количества ошибок на страницах для процессов Internet Explorer</p> <p>Обратите внимание, что для корректной работы этого элемента данных на 64-битной системе потребуется 64-битный Zabbix агент.</p> <p>Обратите внимание: Все атрибуты <i>io_*</i>, <i>gdiobj</i> и <i>userobj</i> доступны только в Windows 2000 и более поздних версиях Windows, не в Windows NT 4.0.</p>			

service.discovery

Список служб
Windows.
Используется
низкоуровневым
обнаружением

		Ключ		
▲	Описание	Возвращаемое значение	Параметры	Комментарии
		service.info[служба,<параметр>]		
		Целое число - с параметром равным <i>state, startup</i>		
		Строка - с параметром равным <i>displayname, path, user</i>		
		Текст - с параметром равным <i>description</i>		
		В частности при <i>state</i> : 0 - запущена, 1 - пауза, 2 - ожидание старта, 3 - ожидание паузы, 4 - ожидание продолжения, 5 - ожидание остановки, 6 - остановлена, 7 - неизвестно, 255 - такой службы не существует	служба - действительное имя службы или её отображаемое имя как в оснастке MMC параметр - state (по умолчанию), <i>displayname, path, user, startup</i> или <i>description</i>	Примеры: ⇒ service.info[SNMPTRAP] - состояние службы SNMPTRAP ⇒ service.info[SNMP Trap] - состояние этой же службы, но указано отображаемое имя ⇒ service.info[EventLog,startup] - состояние запуска при загрузке службы Журнала событий
	Информация о службе.	В частности при <i>startup</i> : 0 - автоматически, 1 - автоматически (отложенный запуск), 2 - вручную, 3 - отключена, 4 - неизвестно, 5 - автоматический запуск по триггеру, 6 - автоматический отложенный запуск по триггеру, 7 - ручной запуск по триггеру		Элементы данных service.info[служба,state] and service.info[служба] вернут одинаковую информацию.
				Обратите внимание, что только параметр равный <i>state</i> у этого элемента данных возвращает значение по несуществующим службам (255).
				Этот элемент данных поддерживается начиная с Zabbix 3.0.0. Его необходимо использовать вместо устаревшего элемента данных service_state[служба].
		services[<тип>,<состояние>,<исключение>]		
	Список служб.	0 - если список	типа - <i>all</i> (по	Примеры:

Ключ				
Описание	Возвращаемое значение	Параметры	Комментарии	
Выполнение WMI запроса и получение первого выбранного объекта.	Служб пуст. Текст - список служб, разделенных новой строкой.	Умолчанию), <i>automatic</i> , <i>manual</i> , <i>disabled</i> состояние - <i>all</i> (по умолчанию), <i>stopped</i> , <i>started</i> , <i>start_pending</i> , <i>stop_pending</i> , <i>running</i> , <i>continue_pending</i> , <i>pause_pending</i> , <i>paused</i> исключение - список служб исключенных из результата. Исключенные службы должны быть указаны в двойных кавычках, разделенные запятой, без пробелов.	⇒ services[,started] - список запущенных служб ⇒ services[automatic, stopped] - список остановленных служб, которые должны быть запущены ⇒ services[automatic, stopped, "service1,service2,service3"] - список остановленных служб, которые должны быть запущены, исключая службы с именами service1,service2 и service3	
Размер виртуального пространства в байтах или в процентах от общего размера.	Целое число, число с плавающей точкой, строка или текст(в зависимости от запроса.)	пространство_имен - название пространства имен WMI запрос - WMI запрос, возвращающий один объект	Параметр исключения поддерживается начиная с версии 1.8.1.	
wmi.get[<пространство_имен>,<запрос>]		Пример: ⇒ wmi.get[root\cimv2,select status from Win32_DiskDrive where Name like '%PHYSICALDRIVE0%'] - возвращает состояние первого физического диска		
vm.vmemory.size[<тип>]		Этот ключ поддерживается начиная с Zabbix 2.2.0.		
		Пример: ⇒ vm.vmemory.size[pavailable] → доступно виртуальной памяти, в процентах		
		Мониторинг статистики виртуальной памяти на основе: Максимального количества памяти, которое может занять Zabbix агент. Текущий предел выделенной памяти в системе или Zabbix агенте, смотря что меньше.		
		Этот ключ поддерживается начиная с Zabbix 3.0.7 и 3.2.3.		

Мониторинг служб Windows

Это руководство содержит пошаговые инструкции по настройке мониторинга служб Windows. Предполагается, что Zabbix сервер и агент уже настроены и работают.

Шаг 1

Узнайте имя службы.

Вы можете получить имя, перейдя в оснастку MMC Службы и открыв свойства службы. На вкладке Общие вы должны увидеть поле называемое 'Имя службы'. Значение которого и будет именем желаемой службы, которое вы будете использовать при настройке элемента данных для наблюдения.

Например, если вы хотите наблюдать службу "workstation", то ваша служба скорее всего будет: **lanmanworkstation**.

Шаг 2

Настройте элемент данных для наблюдения за службой.

Элемент данных service.info[служба,<парам>] возвращает информацию о указанной службе. В зависимости от требуемой вам информации, укажите опцию *парам*, которая принимает следующие значения: *displayname*, *state*, *path*, *user*, *startup* или *description*. Значением по умолчанию является *state*, если *парам* не указан (service.info[служба]).

Тип возвращаемого значения зависит от выбранного *парам*: целое число при *state* и *startup*; строка символов при *displayname*, *path* и *user*; текст при *description*.

Пример:

- Ключ: service.info[lanmanworkstation]
- Тип информации: Целочисленное (положительное)
- Отображение значений: выберите преобразование значений Windows service state

Имеются два преобразования значений Windows service state и Windows service startup type, которые сопоставляют числовое значение в веб-интерфейсе его текстовому представлению.

Обнаружение служб Windows

Низкоуровневое обнаружение дает возможность автоматического создания элементов данных, триггеров и графиков по различных объектам на компьютере. Zabbix может автоматически начать наблюдение за службами Windows на вашей машине, без необходимости знания точного имени службы или создания элементов данных по каждой службе вручную. Можно использовать фильтр для генерирования реальных элементов данных, триггеров и графиков только по интересующим службам.

7.2.2.2 SNMP агент

Обзор

Вы возможно захотите использовать SNMP мониторинг устройств таких как принтеры, сетевые коммутаторы, маршрутизаторы или ИБП, как правило, которые как правило поддерживают SNMP и для которых было бы непрактично пытаться настраивать комплексные системы управления или Zabbix агенты.

Чтобы была возможность получать данные переданные SNMP агентами с этих устройств, Zabbix сервер должен быть изначально сконфигурирован с поддержкой SNMP.

SNMP проверки выполняются только через UDP протокол.

Начиная с версии 2.2.3 демоны Zabbix сервера и прокси опрашивают устройства SNMP множественными значениями за один запрос. Это поведение повлияет на все виды SNMP элементов данных (простые SNMP элементы данных, элементы данных с динамическими индексами и также низкоуровневые SNMP обнаружения) и обработка SNMP элементов данных сейчас должна быть более эффективной. Пожалуйста обратите внимание на раздел с техническими подробностями ниже, описывающий как работает изнутри этот функционал. Начиная с Zabbix 2.4 у каждого интерфейса также имеется настройка "Использовать массовые запросы", которая позволяет отключать массовые запросы у устройств, которые не способны обработать их должным образом.

Начиная с Zabbix 2.2.7 и Zabbix 2.4.2 процессы сервера и прокси будут журналировать строки похожие на следующие в случае получения неправильного/искаженного SNMP ответа:

SNMP response from host "gateway" does not contain all of the requested variable bindings

Пока они не покрывают все возможные проблемные случаи, но они являются удобным удобным идентификатором отдельных SNMP устройств на которых необходимо отключить массовые запросы.

Начиная с версии Zabbix 2.2 демоны сервера и прокси корректно обрабатывают параметр конфигурации Timeout при выполнении SNMP проверок. Дополнительно демоны не выполняют повторных запросов после одного неуспешного (по превышении времени ожидания/неверные настройки учетных данных) SNMP запроса. Ранее на самом деле использовались стандартные для библиотеки SNMP значения времени ожидания и количества повторов (1 секунда и 5 повторов соответственно).

Начиная с версии Zabbix 2.2.8 и Zabbix 2.4.2 демоны сервера и прокси всегда выполняют один повторный запрос: либо через механизм библиотеки SNMP, либо через [внутренний механизм сбора множества значений за один запрос \(bulk\)](#).

Если выполняется мониторинг устройств по SNMPv3, убедитесь что msgAuthoritativeEngineID (также известное как snmpEngineID или “Engine ID”) никогда не будет общим для двух и более устройств. Согласно RFC 2571 (раздел 3.1.1.1) оно должно быть уникальным для каждого устройства.

Настройка мониторинга по SNMP

Для начала мониторинга устройства по SNMP, должны быть выполнены следующие шаги:

Шаг 1

[Создайте узел сети](#) для устройства с SNMP интерфейсом.

Ведите IP адрес. Вы можете использовать один из поставляемых шаблонов SNMP (*Template SNMP Device* и другие), которые автоматически добавят некоторый набор элементов данных. Тем не менее, шаблон может быть не совместим с узлом сети. Нажмите на *Добавить* для сохранения узла сети.

SNMP проверки не используют *Порт агента*, он игнорируется.

Шаг 2

Узнайте строку SNMP (или OID) элемента данных, которую вы хотите мониторить.

Для получения списка строк SNMP, используйте команду **snmpwalk** (часть программного обеспечения [net-snmp](#), которое вы должны были установить как часть инсталляции Zabbix) или эквивалентную утилиту:

```
shell> snmpwalk -v 2c -c public <IP хоста> .
```

'2c' здесь означает версию SNMP, вы также можете заменить его на '1', чтобы использовать 1 версию SNMP на устройстве.

Эта команда должна показать вам список SNMP строк и их последние значения. Если это не произойдет, то возможно что SNMP 'community' отличается от стандартного 'public', в этом случае вам необходимо узнать это имя.

Вы можете пройтись по списку пока не найдете строку которую вы хотите мониторить, например, если вы хотите мониторить входящее количество байт на вашем коммутаторе на 3 порту вы могли бы использовать **IF-MIB::ifInOctets.3** из этой строки:

```
IF-MIB::ifInOctets.3 = Counter32: 3409739121
```

Сейчас вы можете воспользоваться командой **snmpget** для того чтобы определить цифровой OID для 'IF-MIB::ifInOctets.3':

```
shell> snmpget -v 2c -c public -On 10.62.1.22 IF-MIB::ifInOctets.3
```

Обратите внимание, что последнее число в строке это номер порта, который вы ищите для мониторинга. Смотрите также: [Динамические индексы](#).

Вывод команды покажет вам что-то наподобие этого:

```
.1.3.6.1.2.1.2.1.10.3 = Counter32: 3472126941
```

Опять же, последнее число в OID является номером порта.

3COM кажется использует номера портов сотнями, например 1 порт = 101 порт, 3 порт = 103 порт, но в Cisco используются обычные номера, например, 3 порт = 3.

Некоторые из наиболее часто используемых SNMP OID'ов [автоматически конвертируются Zabbix'ом в числовое представление](#).

В последнем примере выше тип значение "Counter32" (32-битный счетчик), что внутренне соответствует типу ASN_COUNTER. Полный список поддерживаемых типов ASN_COUNTER, ASN_COUNTER64, ASN_UNSIGNED, ASN_INTEGER, ASN_UNSIGNED64, ASN_INTEGER64, ASN_FLOAT, ASN_DOUBLE, ASN_TIMETICKS, ASN_GAUGE, ASN_IPADDRESS, ASN_OCTET_STR и ASN_OBJECT_ID (с 2.2.8, 2.4.3). Приведенные типы грубо соответствуют "Counter32", "Counter64", "UInteger32", "INTEGER", "Float", "Double", "Timeticks", "Gauge32", "IpAddress", "OCTET STRING", "OBJECT IDENTIFIER" в выводе **snmpget** утилиты, но могут также отображаться как "STRING", "Hex-STRING", "OID" и другие, в зависимости от наличия полученной подсказки.

Шаг 3

Создайте элемент данных для мониторинга.

Итак, вернитесь назад в Zabbix и нажмите на Элементы данных, выберите созданный ранее узел сети SNMP. В зависимости от того использовали ли вы шаблон при создании узла сети или нет, вы должны будете увидеть список элементов данных SNMP, связанных с вашим узлом сети или попросту окно нового элемента данных. Мы будем исходить из предположения, что вы собираетесь создать элемент данных самостоятельно, с помощью информации, которую вы только что собрали используя **snmpwalk** или **snmpget**, так что введите простое описание на русском языке (или английском) в поле 'Описание' в диалоге нового элемента данных. Убедитесь, что в поле 'Узел сети' находится ваш коммутатор/роутер и измените поле 'Тип' в значение "SNMPv* агент". Введите **community** (обычно **public**) и укажите текстовый или числовой OID, который вы получили ранее, в поле 'SNMP OID', например: .1.3.6.1.2.1.2.1.10.3

Введите 'Порт SNMP' - 161 и 'Ключ' - что-то осмысленное, например, **SNMP-InOctets-Bps**. Установите 'Тип информации' в значение равное Числовой (с плавающей точкой) и шаг предобработки как Скоро в секунду (важно, в противном случае вы будете получать накопленные значения с SNMP устройства вместо последнего изменения). Выберите множитель, если желаете, и укажите 'Интервал обновления', и 'Хранение истории', если вы хотите чтобы значения параметров отличались от умолчаний.

Items

All hosts / Zabbix server Enabled ZBX SNMP JMX IPMI Applications 13 Items 81 Triggers 47

Item Preprocessing

* Name	SNMP: InOctets (Bps)
Type	SNMPv3 agent
* Key	SNMP-InOctets-Bps
* Host interface	127.0.0.1 : 161
* SNMP OID	.1.3.6.1.2.1.2.2.1.10.3
Context name	
Security name	
Security level	authPriv
Authentication protocol	MD5 SHA
Authentication passphrase	
Privacy protocol	DES AES
Privacy passphrase	
Port	161
Type of information	Numeric (float)

Все обязательные поля ввода отмечены красной звёздочкой.

Теперь сохраните элемент данных и перейдите в *Мониторинг → Последние данные*, чтобы увидеть ваши данные SNMP!

Обратите внимание на специфичные опции доступные только для SNMPv3 элементов данных:

Параметр	Описание
Имя контекста	Введите контекстное имя для определения элемента данных в SNMP подсети. Имя контекста поддерживается для SNMPv3 элементов данных с Zabbix 2.2.
Имя безопасности	В данном поле раскрываются пользовательские макросы. Введите имя безопасности. В данном поле раскрываются пользовательские макросы.
Уровень безопасности	Выберете уровень безопасности: noAuthNoPriv - ни аутентификация, ни протокол безопасности не используются AuthNoPriv - используется протокол аутентификации, протокол безопасности нет AuthPriv - используются и протокол аутентификации, и протокол безопасности
Протокол аутентификации	Выберете протокол аутентификации - <i>MD5</i> или <i>SHA</i> .
Фраза-пароль	Введите фразу-пароль для аутентификации

Параметр	Описание
<i>аутентификации</i>	В данном поле раскрываются пользовательские макросы.
<i>Протокол безопасности</i>	Введите протокол безопасности - <i>DES</i> или <i>AES</i> .
<i>Фраза-пароль безопасности</i>	Введите фразу-пароль безопасности.
	В данном поле раскрываются пользовательские макросы.
В случае некорректных учётных данных SNMPv3 (имя безопасности, протокол/фраза-пароль аутентификации, протокол безопасности) Zabbix получит ERROR от net-snmp, за исключением ошибочного <i>Фразы-пароль безопасности</i> , в этом случае Zabbix получит ошибку ВРЕМЕНИОЖИДАНИЯ от net-snmp.	

При изменениях в *Протокол аутентификации*, *Фраза-пароль аутентификации*, *Протокол безопасности* или *Фраза-пароль безопасности*, чтобы эти изменения применились, необходимо перезапустить сервер/прокси.

Пример 1

Общий пример:

Параметр	Описание
Community public	
OID	1.2.3.45.6.7.8.0 (или .1.2.3.45.6.7.8.0)
Ключ	<Уникальная строка, которая используется как ссылка в триггерах> Например, "my_param".

Обратите внимание, что OID можно задать в числовом или строковом представлении. Тем не менее, в некоторых случаях, строковый OID должен быть сконвертирован в числовое представление. Для этого можно использовать утилиту snmpget:

```
shell> snmpget -On localhost public enterprises.ucdavis.memory.memTotalSwap.0
```

Мониторинг SNMP параметров возможен, если указан флаг --with-net-snmp при конфигурировании исходных кодов Zabbix.

Пример 2

Мониторинг времени работы:

Параметр	Описание
Community	public
Oid	MIB::sysUpTime.0
Ключ	router.uptime
Тип информации	Числовой (с плавающей точкой)
Единица измерения	uptime
Множитель	0.01

Обработка массовых SNMP запросов

Начиная с 2.2.3 Zabbix сервер и прокси одним опросом запрашивают множество SNMP элементов данных. Такое поведение затрагивает следующие типы SNMP элементов данных:

- обычные SNMP элементы данных;
- [SNMP элементы данных с динамическими индексами](#);
- [Правила низкоуровневого SNMP обнаружения](#).

Все элементы данных SNMP с одного интерфейса запланированы на опрос в одно время. Первые два типа элементов данных собираются поллерами порциями не более чем по 128 элементов данных, в то время как правила низкоуровневого обнаружения обрабатываются индивидуально как и ранее.

На низком уровне, есть два вида операций выполняемых при опросе значений: получение нескольких заданных объектов и прохождение дерева OID-ов.

Для “получения” используется GetRequest-PDU с не более чем 128 привязанных переменных. Для “прохождения”, используется GetNextRequest-PDU для SNMPv1 и GetBulkRequest с полем “max-repetitions” с наибольшим количеством в 128 полученных значений используется для SNMPv2 и SNMPv3.

Таким образом преимущества массовой обработки для каждого типа SNMP элемента данных описаны ниже:

- простые SNMP элементы данных получают преимущество от улучшения “получения”;
- SNMP элементы данных с динамическими индексами получают преимущество и от улучшений “получения” и “прохождения”: “получение” используется для проверки индексов, а “прохождение” для построения кэша значений;
- правила низкоуровневого SNMP обнаружения получают преимущество от улучшения “прохождения”.

Тем не менее, есть техническая проблема что не все устройства способны вернуть 128 значений за один запрос. Некоторые всегда возвращают корректный ответ, но другие либо отвечают с ошибкой “tooBig(1)”, либо не отвечают вообще, когда потенциальный запрос превышает определенный лимит.

Для вычисления оптимального количества запрашиваемых объектов с устройства, Zabbix использует следующую стратегию. Начинается с осторожного запроса одного значения. Это запрос выполнен успешно, запрашивается 2 значения за один запрос. Если запрос снова выполнен успешно, запрашивается 3 значения за запрос и продолжается аналогично умножением количества запрашиваемых значений на 1.5, в результате получается следующая последовательность размера запросов: 1, 2, 3, 4, 6, 9, 13, 19, 28, 42, 63, 94, 128.

Однако если устройство отказывается от ответа на определенный запрос (к примеру, 42 переменных), Zabbix делает 2 вещи.

Первое, для текущей серии элементов данных Zabbix делит пополам количество элементов данных за один запрос и запрашивает 21 переменных. Если устройство доступно, далее запросы должны работать в большинстве случаев, потому что известно что 28 переменных забиралось, а 21 значительно меньше. Тем не менее если проблема с запросами продолжается, Zabbix уменьшает количество запросов последовательно согласно этому алгоритму. Если и далее проблемы с запросами все еще актуальны, значит устройство определенно не отвечает и количество запросов это не корень проблемы.

Второе дело, которое делает Zabbix для дальнейших порций элементов данных - это, начиная с последнего удачного количества переменных (28 в нашем случае), продолжает увеличивать количество переменных за запрос на 1 до достижения лимита. Например, предположим что максимально возможное количество запросов для данного устройства это 32, последующие запросы будут следующими 29,30,31,32 и 33. Последний запрос будет неудачным и Zabbix никогда более не запросит 33 значения за один запрос. С этого момента, Zabbix всегда будет запрашивать 32 значения для этого устройства.

Если большие запросы неудачно завершаются с определенным количеством переменных, это может означать одно из двух. Точный критерий по которому устройство может ограничивать запросы неизвестен, но мы можем приблизительно рассчитать количество переменных. Первая вероятность - что количество значений примерно равно действительному лимиту размера для данного устройства в общем случае: иногда запросов либо меньше чем лимит, иногда больше. Вторая вероятность, что UDP пакет был потерян. В этом случае, если Zabbix сталкивается с неудачным запросом, он уменьшает максимальное количество запрашиваемых значение за запрос для попытки получения с устройства корректного диапазона, но (начиная с 2.2.8) только до 2 раз.

В примере выше, если запрос с 32 переменными будет неудачен, Zabbix уменьшит количество до 31. Если неудача случиться снова, Zabbix уменьшит количество до 30. Тем не менее, Zabbix не будет

уменьшать количество ниже 30, потому что он предположит, что следующие проблемы по причине потерянных UDP пакетов, чем скорее ограничение устройства.

Если, однако, устройство не может обрабатывать массовые запросы корректно и по другим причинам, начиная с Zabbix 2.4 имеется настройка “Использовать массовые запросы” у каждого интерфейса, которая позволяет отключить массовые запросы у этого устройства.

7.2.2.1 Динамические индексы

Обзор

Хотя вы можете быть и найдете требуемый номер индекса (например, сетевого интерфейса) из всех SNMP OID'ов, иногда вы не можете полностью положиться на номер индекса 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 просматриваться следующим образом:

```
...
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 присоединен к OID'у данных, чтобы получить интересуемое 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 еще одного запроса к наблюдаемому устройству. Обратите внимание, 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 измененный индекс снова проходит и кэширует таблицу индексов SNMP.

7.2.2.2 Специальные OID'ы

Некоторые OID, наиболее часто используемые в SNMP, автоматически преобразовываются в числовые представления самим Zabbix. Например, **ifIndex** преобразовывается в **1.3.6.1.2.1.2.2.1.1**, **ifIndex.0** преобразовывается в **1.3.6.1.2.1.2.2.1.0**.

В таблице содержится перечень специальных OID.

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 о интерфейсе. Эта строка может включать в себя название компании производителя, имя продукта, аппаратную версию 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 байтах
ifSpeed	1.3.6.1.2.1.2.2.1.5	Оценочная скорость интерфейса текущей пропускной способности 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	Полное число полученных байтов, включая символы заголовков.
ifInUcastPkts	1.3.6.1.2.1.2.2.1.11	Количество пакетов одноадресной рассылки, доставленных на верхний уровень стека сетевых протоколов.
ifInNUcastPkts	1.3.6.1.2.1.2.2.1.12	Количество пакетов НЕ одноадресной рассылки (broadcast и multicast), доставленных на верхний уровень стека сетевых протоколов.
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 на верхний уровень стека сетевых протоколов. Одна 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 на верхний уровень стека сетевых протоколов.
ifInUnknownProtos	1.3.6.1.2.1.2.2.1.15	Количество пакетов, полученных через интерфейс, но отвергнутых из-за неизвестного или не поддерживаемого

Специальный OID	Идентификатор	Описание
		протокола.
ifOutOctets	1.3.6.1.2.1.2.2.1.16	Полное количество отправленных октетов с интерфейса, включая символы заголовков.
ifOutUcastPkts	1.3.6.1.2.1.2.2.1.17	Полное количество пакетов, которые пытались отправить верхний уровень стека сетевых протоколов, и которые адресованы НЕ на broadcast или multicast адреса на этом субуровне, включая те которые были отвергнуты или не отправлены.
ifOutNUcastPkts	1.3.6.1.2.1.2.2.1.18	Полное количество пакетов, которые пытались отправить верхний уровень стека сетевых протоколов, и которые адресованы на broadcast или multicast адреса на этом субуровне, 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. Одна 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 могут быть отправлены из-за ошибок.
ifOutQLen	1.3.6.1.2.1.2.2.1.21	Длина очереди исходящих пакетов (в пакетах).

7.2.2.3 МIB файлы

Введение

MIB обозначает Базу Управляющей Информации. MIB файлы позволяют вам использовать текстовое представление OID'a (Идентификатор Объекта).

Например,

`ifHCOutOctets`

является текстовым представлением OID'a

`1.3.6.1.2.1.31.1.1.1.10`

Вы можете использовать оба варианта при мониторинге SNMP устройств с Zabbix, но если вам удобнее пользоваться текстовым представлением, вам нужно установить MIB файлы.

Установка MIB файлов

На системах на основе Debian:

```
# apt install snmp-mibs-downloader
# download-mibs
```

На системах на основе RedHat:

```
# yum install net-snmp-libs
```

Включение MIB файлов

На системах на основе RedHat mib файлы должны быть включены по умолчанию. На системах на основе Debian вам нужно отредактировать файл `/etc/snmp/snmp.conf` и раскомментировать 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 reenable
# loading them by commenting out the following line.
```

```
#mibs :
```

Тестирование MIB файлов

Тестирование snmp MIB можно выполнить с использованием `snmpwalk` утилиты. Если у вас эта утилита не установлена, вы можете использовать следующие инструкции.

На системах на основе Debian:

```
# apt install snmp
```

На системах на основе RedHat:

```
# yum install net-snmp-utils
```

После чего следующая команда не выдаст ошибку при выполнении запроса к сетевому устройству:

```
$ snmpwalk -v 2c -c public <IP СЕТЕВОГО УСТРОЙСТВА> 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 - иметь в виду, что процессы Zabbix не знают об изменениях, которые сделаны в MIB файлы. Поэтому после каждого изменения вам необходимо перезапустить Zabbix сервер или прокси, например:

```
# service zabbix-server restart
```

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.

Скажем, если вы хотите использовать [CISCO-SMI](#) MIB файл. Следующие инструкции загрузят и установят 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/local/share/snmp/mibs" >> /etc/snmp/snmp.conf
# cp /tmp/CISCO-SMI.my /usr/local/share/snmp/mibs
```

Теперь вы сможете 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 получаете 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.



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.

7.2.2.3 SNMP трапы

Обзор

Получение SNMP трапов является полной противоположностью запросам к SNMP устройствам.

В этом случае информация отправляется с SNMP устройства и собирается или “ловится” Zabbix'ом.

Обычно трапы отправляются при наступлении некоторых условий и агент подключается на 162 порт сервера (в отличии от 161 порта на стороне агента, который используется для запросов).

Использование трапов может помочь обнаружить некоторые кратковременные проблемы, которые происходят между интервалами опроса и могут быть пропущены при запросах данных.

Получение SNMP трапов в Zabbix рассчитано на работу с **snmptrapd** и с одним из встроенных механизмов передачи трапов в Zabbix - либо perl скрипт, либо SNMPTT.

Последовательность действий при получении трапа:

1. **snmptrapd** получает трап
2. snmptrapd передает трап в SNMPTT или вызывает получателя трапов Perl
3. SNMPTT или получатель трапов Perl, форматируют и записывают трап в файл
4. Zabbix SNMP траппер читает и анализирует файл с трапами
5. Zabbix ищет все соответствующие элементы данных с типом “SNMP трап” на интерфейсах узлов сети по каждому трапу, которые совпадают с полученным адресом из трапа. Возьмите на заметку, только выбранный “IP” или “DNS” у интерфейса узла сети используется в процессе поиска совпадения.
6. По каждому найденному SNMP интерфейсу, трап сравнивается со всеми регулярными выражениями из элементов данных “snmptrap[регулярное выражение]”. Если совпадение найдено, трап записывается значением для **всех** совпавших элементов данных. Если совпадений не найдено, но существует элемент данных “snmptrap.fallback”, трап записывается значением для этого элемента данных.
7. Если совпадений не было найдено ни с одним из соответствующих SNMP интерфейсов, Zabbix по умолчанию журналирует несовпадающие трапы. (Это поведение настраивается в “Журналировать не совпадающие SNMP трапы” в Администрирование → Общие → Прочие).

1 Настройка SNMP трапов

Настройка следующих полей в веб-интерфейсе является специфичной для этого типа элементов данных:

- Ваш узел сети должен иметь SNMP интерфейс

В **Настройка → Узлы сети**, в поле **Интерфейсы узла сети** добавьте SNMP интерфейс с корректными IP или DNS адресами. Адрес из каждого полученного трапа сравнивается с IP и DNS адресами всех SNMP интерфейсов для поиска соответствующих узлов сети.

- Настройка элемента данных

В поле **Ключ** используйте один из ключей SNMP трапов:

Описание	Ключ	Возвращаемое значение	Комментарии
Отлов всех SNMP трапов , который	snmptrap[регулярное выражение]	SNMP трап	Этот элемент данных можно привязать

Описание	Ключ	Возвращаемое значение	Комментарии
совпадают с регулярным выражением , указанном в поле регулярное выражение . Если регулярное выражение не указано, принимаются все трапы.			только к SNMP интерфейсам. Элемент данных поддерживается начиная с Zabbix 2.0.0 . <i>Обратите внимание:</i> Начиная с Zabbix 2.0.5, в параметре этого ключа элемента данных поддерживаются пользовательские макросы и глобальные регулярные выражения.
Отлов всех SNMP трапов, которые не совпадают ни с одним из элементов данных snmptrap[] для этого интерфейса. Многострочное совпадение по регулярному выражению в данный момент времени не поддерживается.	snmptrap.fallback	SNMP трап	Этот элемент данных можно привязать только к SNMP интерфейсам. Элемент данных поддерживается начиная с Zabbix 2.0.0 .

Укажите **Тип информации** равным 'Журнал (лог)' для обработки штампов времени. Обратите внимание, что другие форматы, такие как 'Числовой' также приемлемы, но для этого может потребоваться пользовательский обработчик трапов.

Для того чтобы мониторинг SNMP трапов работал, он должен быть сначала корректно настроен.

2 Настройка мониторинга SNMP трапов

Настройка Zabbix сервера/прокси

Для чтения трапов, Zabbix сервер или прокси должны быть настроены на запуск процесса SNMP траппера и должны знать абсолютный путь к файлу с трапами, который заполняется при помощи SNMPTT или получателя трапов Perl. Чтобы это сделать, измените файл конфигурации ([zabbix_server.conf](#) или [zabbix_proxy.conf](#)):

1. StartSNMPTrapper=1
2. SNMPTrapperFile=[ФАЙЛ С ТРАПАМИ]

Если используется systemd параметр **PrivateTmp**, этот файл вряд ли заработает в */tmp*.

Настройка SNMPTT

Для начала, snmptrapd необходимо настроить на использование SNMPTT.

Для лучшей производительности, SNMPTT необходимо сконфигурировать демоном с использованием **snmpthandler-embedded** для передачи ему трапов. Смотрите инструкции по настройке SNMPTT на его сайте:

<http://snmptt.sourceforge.net/docs/snmppt.shtml>

При конфигурировании SNMPTT на получение трапов, настройте SNMPTT на журналирование этих трапов:

1. журналирование трапов в файл с трапами, который Zabbix будет читать:

```
log_enable = 1
log_file = [ФАЙЛ С ТРАПАМИ]
```
2. задайте формат времени/даты:

```
date_time_format = %H:%M:%S %Y/%m/%d = [ФОРМАТ ВРЕМЕНИ]
```

Теперь отформатируйте трапы, чтобы они распознавались Zabbix'ом (измените snmptt.conf):

- 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)
- Подробнее о формате SNMP траповсмотрите ниже.

Не используйте неизвестные трапы - Zabbix может их не распознать. Неизвестные трапы могут быть обработаны, если задать общее событие в snmptrap.conf:

EVENT general .* “General event” Normal

Настройка получателя Perl трапов

Требования: Perl, Net-SNMP скомпилированный с --enable-embedded-perl (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 в Zabbix сервер напрямую с snmptrapd. Для его настройки:

- добавьте perl скрипт в файл конфигурации snmptrapd (snmptrapd.conf), Например:
 perl do "[ПОЛНЫЙ ПУТЬ К СКРИПТУ ПОЛУЧАТЕЛЯ PERL]";
- настройте сам получатель, например:
 \$SNMPTrapperFile = '[ФАЙЛ С ТРАПАМИ]';
 \$DateTimeFormat = '[ФОРМАТ ДАТЫ/ВРЕМЕНИ]';

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
```

Формат SNMP трапа

Все пользовательские получатели трапов perl и конфигурация SNMPTT трапов должны форматировать трап следующим образом:

[штамп времени] [трап, часть 1] ZBXTRAP [адрес] [трап, часть 2], где

- [штамп времени] - штамп времени используемый в элементах данных типа 'Журнал (лог)'
- 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:

11:30:15 2011/07/27 .1.3.6.1.6.3.1.1.5.3 Normal “Status Events” localhost - ZBXTRAP 192.168.1.1 Link down on interface 2. Admin state: 1. Operational state: 2

Такое сообщение будет результатом следующего трапа для SNMP интерфейса с IP=192.168.1.1:

11:30:15 2011/07/27 .1.3.6.1.6.3.1.1.5.3 Normal “Status Events” localhost - Link down on interface 2. Admin state: 1. Operational state: 2

3 Требования к системе

Поддержка больших файлов

У Zabbix имеется “Поддержка больших файлов” для файлов SNMP трапов. Максимальный размер файла, который Zabbix может прочитать, это 2^{63} (8 Эбайт). Обратите внимание, что файловая система может иметь меньшее ограничение на максимальный размер файлов.

Ротация журнала

Zabbix не предоставляет какую-либо систему ротации журналов - должно быть обработано пользователем. Ротация журналов должна начинаться с переименования старого файла и только после этого удалять его, чтобы никакие трэпы не пропали:

1. Zabbix открывает файл с трэпами с последней известной позиции и переходит к 3 шагу.
2. Zabbix проверяет был ли скопирован в данный момент открытый файл, сравнивая номера inode с заданным у файла трэпов номером inode. Если нет открытого файла, Zabbix сбрасывает последнюю позицию и переходит к 1 шагу.
3. Zabbix читает данные из открытого в данный момент файла и устанавливает новую позицию.
4. Обрабатываются новые данные. Если этот файл был ротирован, то он закрывается и Zabbix переходит назад ко 2 шагу.
5. Если не было новых данных, Zabbix засыпает на 1 секунду и возвращается ко 2 шагу.

Файловая система

Из-за реализации файла с трэпами, Zabbix'у требуется файловая система с поддержкой inode для того чтобы различать файлы (эта информация берется из вызова stat()).

4 Пример установки

Этот пример использует snmptrapd + SNMPTT для передачи трэпов Zabbix серверу. Установка:

1. **zabbix_server.conf** - настройте Zabbix, чтобы запускался SNMP трэппер процесс и укажите абсолютный путь к файлу с трэпами:
StartSNMPTrapper=1
SNMPTrapperFile=/tmp/my_zabbix_traps.tmp
2. **snmptrapd.conf** - добавьте SNMPTT как обработчик трэпов:
traphandle default snmptt
3. **snmptt.ini** - настройте выходной файл и формат времени:
log_file = /tmp/my_zabbix_traps.tmp
date_time_format = %H:%M:%S %Y/%m/%d
4. **snmptt.conf** - укажите формат трэпов по умолчанию:
EVENT general .* "General event" Normal
FORMAT ZBXTRAP \$aA \$ar
5. Создайте ТЕСТ элемент данных с типом SNMP трэп:
Узел сети с IP адресом SNMP: 127.0.0.1
Ключ: snmptrap["General"]
Формат времени журнала: hh:mm:ss yyyy/MM/dd

В результате:

1. Используйте следующую команду для отправки трэпа:
snmptrap -v 1 -c public 127.0.0.1 '.1.3.6.1.6.3.1.1.5.3' '0.0.0.0' 6 33 '55' '.1.3.6.1.6.3.1.1.5.3 s "teststring000"
2. Полученный трэп:
15:48:18 2011/07/26 .1.3.6.1.6.3.1.1.5.3.0.33 Normal "General event" localhost - ZBXTRAP
127.0.0.1 127.0.0.1
3. Значение ТЕСТ элемента данных:
15:48:18 2011/07/26 .1.3.6.1.6.3.1.1.5.3.0.33 Normal "General event" localhost - 127.0.0.1

В этом примере используется SNMPTT как **traphandle**. Для лучшей производительности на продуктивных системах используйте встроенный Perl для передачи трэпов от snmptrapd к SNMPTT или напрямую Zabbix'у.

5 Смотрите также

- [Руководство по настройке SNMP трапов на CentOS на сайте zabbix.org \[en\]](#)

7.2.2.4 Проверки IPMI

Обзор

Вы можете наблюдать за состоянием и доступностью устройств Intelligent Platform Management Interface (IPMI) в Zabbix. Для выполнения проверок по IPMI Zabbix сервер должен быть изначально [сконфигурирован](#) с поддержкой IPMI.

IPMI стандартизованный интерфейс для удаленного управления “lights-out” или “out-of-band” компьютерными системами. Он позволяет наблюдать за состоянием аппаратного обеспечения напрямую с так называемых карт управления “out-of-band”, независимо от операционной системы или же от наличия питания на машине.

Zabbix IPMI мониторинг работает только с устройствами имеющими поддержку IPMI (HP iLO, DELL DRAC, IBM RSA, Sun SSP и т.п.).

Начиная с Zabbix 3.4, добавлен новый процесс IPMI менеджер, который выполняет распределение проверок IPMI между IPMI поллерами. Теперь, узел сети всегда опрашивается одним и тем же IPMI поллером, таким образом снижается количество открытых подключений к BMC контроллерам. Благодаря этим изменениям можно безопасно увеличивать количество IPMI поллеров, не беспокоясь о перегрузке BMC контроллеров. Процесс IPMI менеджер автоматически запускается, если запускается хотя бы один IPMI поллер.

Смотрите также [известные проблемы](#) по IPMI проверкам.

Настройка

Настройка узла сети

Узел сети необходимо настроить для обработки проверок IPMI. Необходимо добавить интерфейс IPMI, с соответствующими IP адресом и номером порта, и задать параметры аутентификации IPMI.

Смотрите [настройку узлов сети](#) для получения более подробных сведений.

Настройка сервера

По умолчанию, Zabbix сервер не запускает IPMI поллеры, таким образом 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 следующей:

```
StartIPMIPollers=3
```

Сохраните файл и затем перезапустите zabbix_server.

Настройка элемента данных

Для [настройки элемента данных](#) на уровне узла сети:

- Для *Интерфейса узла сети* выберите IPMI IP и порт
- Выберите 'IPMI агент' как *Tin*
- Укажите *IPMI датчик* (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.
- Введите [ключ](#) элемента данных, уникальный в пределах узла сети (скажем, ipmi.fan.rpm)
- Выберите соответствующий тип информации ('Числовой (с плавающей точкой)' в этом случае, для дискретных датчиков - 'Числовой (целый)'), единицы измерения (скорее всего 'rpm') и любые другие требуемые атрибуты элемента данных.

Время ожидания и завершение сессии

Время ожидания IPMI сообщений и количества попыток определены в библиотеке OpenIPMI. В связи с текущим дизайном OpenIPMI, невозможно сделать эти значения настраиваемыми из Zabbix, ни на уровне интерфейса, ни на уровне элемента данных.

Время ожидания неактивности IPMI сессии для LAN равняется 60 +/-3 секунд. В настоящее время невозможно реализовать периодическую отправку команды Активации Сессии в OpenIPMI. Если проверки IPMI элементов данных выполняются от Zabbix к конкретному BMC в течении более чем время ожидания сессии, настроенное в BMW, тогда следующая проверка IPMI после того, как истечет время ожидания, приведет к отдельным сообщениям о превышении времени ожидания, попытка или ошибке при получении. После того, как откроется новая сессия и сделано повторное сканирование BMC. Если вы хотите избежать лишних сканирований BMC, рекомендуется установить интервал опроса IPMI элементов данных ниже времени ожидания неактивности IPMI сессии, которое настраивается в BMC.

Примечания о дискретных датчиках IPMI

Для поиска датчиков на узле сети запустите Zabbix сервер с включенным **DebugLevel=4**. Подождите пару минут и найдите записи об обнаруженных датчиках в журнале Zabbix сервера:

```
$ 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:0x3 ('discrete_state') type:0x7 ('processor')
full_name:'(r0.32.3.0).CATERR'
8358:20130318:111122.170 Added sensor: host:'192.168.1.12:623' id_type:0 id_sz:15
id:'CPU Therm Trip' reading_type:0x3 ('discrete_state') type:0x1 ('temperature')
full_name:'(7.1).CPU Therm Trip'
8358:20130318:111122.171 Added sensor: host:'192.168.1.12:623' id_type:0 id_sz:17
id:'System Event Log' reading_type:0x6f ('sensor specific') type:0x10
('event_logging_disabled') full_name:'(7.1).System Event Log'
8358:20130318:111122.171 Added sensor: host:'192.168.1.12:623' id_type:0 id_sz:17
id:'PhysicalSecurity' reading_type:0x6f ('sensor specific') type:0x5
('physical_security') full_name:'(23.1).PhysicalSecurity'
8358:20130318:111122.171 Added sensor: host:'192.168.1.12:623' id_type:0 id_sz:14
id:'IPMI Watchdog' reading_type:0x6f ('sensor specific') type:0x23 ('watchdog_2')
full_name:'(7.7).IPMI Watchdog'
8358:20130318:111122.171 Added sensor: host:'192.168.1.12:623' id_type:0 id_sz:16
id:'Power Unit Stat' reading_type:0x6f ('sensor specific') type:0x9 ('power_unit')
full_name:'(21.1).Power Unit Stat'
8358:20130318:111122.171 Added sensor: host:'192.168.1.12:623' id_type:0 id_sz:16
id:'P1 Therm Ctrl %' reading_type:0x1 ('threshold') type:0x1 ('temperature')
full_name:'(3.1).P1 Therm Ctrl %'
8358:20130318:111122.172 Added sensor: host:'192.168.1.12:623' id_type:0 id_sz:16
id:'P1 Therm Margin' reading_type:0x1 ('threshold') type:0x1 ('temperature')
full_name:'(3.2).P1 Therm Margin'
8358:20130318:111122.172 Added sensor: host:'192.168.1.12:623' id_type:0 id_sz:13
id:'System Fan 2' reading_type:0x1 ('threshold') type:0x4 ('fan')
full_name:'(29.1).System Fan 2'
8358:20130318:111122.172 Added sensor: host:'192.168.1.12:623' id_type:0 id_sz:13
id:'System Fan 3' reading_type:0x1 ('threshold') type:0x4 ('fan')
full_name:'(29.1).System Fan 3'
8358:20130318:111122.172 Added sensor: host:'192.168.1.12:623' id_type:0 id_sz:14
id:'P1 Mem Margin' reading_type:0x1 ('threshold') type:0x1 ('temperature')
full_name:'(7.6).P1 Mem Margin'
```

```
8358:20130318:111122.172 Added sensor: host:'192.168.1.12:623' id_type:0 id_sz:17
id:'Front Panel Temp' reading_type:0x1 ('threshold') type:0x1 ('temperature')
full_name:'(7.6).Front Panel Temp'
8358:20130318:111122.173 Added sensor: host:'192.168.1.12:623' id_type:0 id_sz:15
id:'Baseboard Temp' reading_type:0x1 ('threshold') type:0x1 ('temperature')
full_name:'(7.6).Baseboard Temp'
8358:20130318:111122.173 Added sensor: host:'192.168.1.12:623' id_type:0 id_sz:9 id:'BB
+5.0V' reading_type:0x1 ('threshold') type:0x2 ('voltage') full_name:'(7.1).BB +5.0V'
8358:20130318:111122.173 Added sensor: host:'192.168.1.12:623' id_type:0 id_sz:14
id:'BB +3.3V STBY' reading_type:0x1 ('threshold') type:0x2 ('voltage')
full_name:'(7.1).BB +3.3V STBY'
8358:20130318:111122.173 Added sensor: host:'192.168.1.12:623' id_type:0 id_sz:9 id:'BB
+3.3V' reading_type:0x1 ('threshold') type:0x2 ('voltage') full_name:'(7.1).BB +3.3V'
8358:20130318:111122.173 Added sensor: host:'192.168.1.12:623' id_type:0 id_sz:17
id:'BB +1.5V P1 DDR3' reading_type:0x1 ('threshold') type:0x2 ('voltage')
full_name:'(7.1).BB +1.5V P1 DDR3'
8358:20130318:111122.173 Added sensor: host:'192.168.1.12:623' id_type:0 id_sz:17
id:'BB +1.1V P1 Vccp' reading_type:0x1 ('threshold') type:0x2 ('voltage')
full_name:'(7.1).BB +1.1V P1 Vccp'
8358:20130318:111122.174 Added sensor: host:'192.168.1.12:623' id_type:0 id_sz:14
id:'BB +1.05V PCH' reading_type:0x1 ('threshold') type:0x2 ('voltage')
full_name:'(7.1).BB +1.05V PCH'
```

Для расшифровки типов датчиков IPMI и их состояния, скачайте экземпляр спецификаций IPMI 2.0 с <http://www.intel.com/content/www/us/en/servers/ipmi/ipmi-specifications.html> (Во время подготовки документации новейшим документом был <http://www.intel.com/content/dam/www/public/us/en/documents/product-briefs/second-gen-interface-spec-v2.pdf>)

Начнем с параметра “тип_чтения” (“reading_type”). Для расшифровки кода “reading_type” используйте раздел “Table 42-1, Event/Reading Type Code Ranges” из спецификации. Большинство датчиков из нашего примера имеют “reading_type:0x1” означающих “порог” датчика. “Table 42-3, Sensor Type Codes” показывает, что “type:0x1” - температурный датчик, “type:0x2” - датчик напряжения, “type:0x4” - датчик частоты вращения вентилятора системы охлаждения и так далее. Пороговые датчики иногда называют “аналоговыми” датчиками, так как они измеряют непрерывные параметры, такие как температуру, напряжение, частоту вращения в минуту.

Другой пример - датчик с “reading_type:0x3”. “Table 42-1, Event/Reading Type Code Ranges” говорит, что считываемые типы кодов 02h-0Ch это “Общий Дискретный” датчик. Дискретные датчики имеют до 15 возможных состояний (другими словами - до 15 значимых бит). К примеру, для датчика 'CATERR' с “type:0x7” “Table 42-3, Sensor Type Codes” показывает, что этот тип обозначает “Процессор” и значение отдельных бит: 00h (наименьший значимый бит) - IERR(внутренняя ошибка процессора), 01h - Перегрев процессора и т.д.

Есть в нашем примере несколько датчиков с “reading_type:0x6f”. Для этих датчиков “Table 42-1, Event/Reading Type Code Ranges” советует использовать “Table 42-3, Sensor Type Codes” для расшифровки значений бит. Например, датчик 'Power Unit Stat' имеет тип “type:0x9”, который означает “Блок питания”. Смещение 00h означает “Выключено/Обесточено”. Другими словами, если младший значимый бит равен 1, то сервер выключен. Для проверки этого бита можно воспользоваться функция **band** с маской 1. Выражение триггера может выглядеть следующим образом

```
{www.zabbix.com:Power Unit Stat.band(#1,1)}=1
```

для предупреждения о выключенном сервере.

Заметки о 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), присоединенный к концу 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 следующее именование “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 обновлен в новом Linux дистрибутиве, который использует OpenIPMI-2.0.19 (или более позднюю), элементы данных с такими IPMI дискретными датчиками перейдут в “NOT SUPPORTED”. Вам потребуется изменить их имена *IPMI датчик* (удалить с конца '0') и подождать некоторое время пока они станут “Активированными” снова.

Заметки о одновременной доступности пороговых и дискретных датчиках

Некоторые IPMI агенты предоставляют одновременно пороговые и дискретные датчики под одним именем. В версиях Zabbix до 2.2.8, выбирался первый полученный датчик. Начиная с версии 2.2.8, предпочтение всегда отдается 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/удален, узел сети в обслуживании и так далее), то в этом случае Zabbix сервер/прокси продолжит сбор данных по IPMI с этого узла сети до перезагрузки сервера/прокси.

7.2.2.5 Простые проверки

Обзор

Простые проверки в основном используются для удаленных безагентных проверок сервисов.

Обратите внимание, что для простых проверок Zabbix агент не требуется. За обработку (созданием внешних подключений и т.д.) простых проверок 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]
```

Поля *Имя пользователя* и *пароль* в конфигурации простых элементов данных используются для элементов данных мониторинга VMware; иначе игнорируются.

Поддерживаемые простые проверки

Список поддерживаемых простых проверок:

Смотрите также:

- [Ключи элементов данных для мониторинга VMware](#)

▲	Описание	Возвращаемое значение	Ключ	Параметры	Комментарии
	Доступность хоста через пинг по ICMP.	0 - ошибка при пинге по ICMP 1 - успешный пинг по ICMP	icmpping[<цель>,<пакеты>,<интервал>,<размер>,<время ожидания>]	цель - IP хоста или DNS имя пакеты - количество пакетов между успешными пакетами в	Пример: ⇒ icmpping[4] → если по крайней мере один пакет из четырех вернется, элемент данных возвратит 1. Смотрите также таблицу со

Ключ

▲ Описание	Возвращаемое значение	Параметры	Комментарии
		миллисекундах размер - размер пакета в байтах время ожидания - время ожидания в миллисекундах	значениями по умолчанию .
	icmppingloss[<цель>,<пакеты>,<интервал>,<размер>,<время ожидания>]	цель - IP хоста или DNS имя пакеты - количество пакетов интервал - время между успешными пакетами в миллисекундах размер - размер пакета в байтах время ожидания - время ожидания в миллисекундах	
Процентное отношение потерянных пакетов.	Число с плавающей точкой.		Смотрите также таблицу со значениями по умолчанию .
	icmppingsec[<цель>,<пакеты>,<интервал>,<размер>,<время ожидания>,<режим>]	цель - IP хоста или DNS имя пакеты - количество пакетов интервал - время между успешными пакетами в миллисекундах размер - размер пакета в байтах время ожидания - время ожидания в миллисекундах режим - один из min, max, avg (по умолчанию)	Если хост недоступен (превышено время ожидания), элемент данных вернет 0. Если элемент данных “icmppingsec” вернет значение меньше 0.0001 секунд, значение будет равно 0.0001 секунд.
Время ответа на пинг по ICMP (в секундах).	Число с плавающей точкой		Смотрите также таблицу со значениями по умолчанию .
	net.tcp.service[сервис,<ip>,<порт>]	сервис - один из ssh, ntp, ldap, smtp, ftp, http, pop, nntp, imap, tcp, https, telnet (смотри детали) ip - IP адрес или DNS имя (по умолчанию, используется IP/DNS узла сети) порт - номер порта сервиса (по умолчанию для сервиса используется порт, указываемый в параметре)	Пример: ⇒ net.tcp.service[ftp,,45] → можно использовать для проверки доступности FTP сервера на 45 порту TCP. Обратите внимание, для сервиса tcp обязательно нужно указывать порт. Эти проверки могут привести к дополнительным записям в системных лог файлах (обычно для сессии SMTP и SSH журналируются). Проверка шифрованных
Проверка запущен ли сервис и отвечает ли на TCP подключения.	0 - сервис недоступен 1 - сервис работает		

Описание	Возвращаемое значение	Параметры	Комментарии
Проверка производительности сервиса.	<code>net.tcp.service.perf[сервис,<ip>,<порт>]</code>	сервис - один из ssh, ntp, ldap, smtp, ftp, http, pop, nntp, imap, tcp, https, telnet (смотри детали) ip - IP адрес или DNS имя (по умолчанию, используется IP/DNS узла сети) порт - номер порта (по умолчанию для сервиса используется стандартный номер порта).	протоколов (таких как IMAP на 993 порту или POP на 995 порту) в настоящее время не поддерживается. Как решение, пожалуйста, для подобных проверок используйте <code>net.tcp.service[tcp,<ip>,порт]</code> . Сервисы <i>https</i> и <i>telnet</i> поддерживаются Zabbix начиная с версии 2.0. Пример ключа: ⇒ <code>net.tcp.service.perf[ssh]</code> → можно использовать для проверки скорости начального ответа от SSH сервера.
Проверка запущен ли сервис и отвечает ли на UDP подключения.	<code>net.udp.service[сервис,<ip>,<порт>]</code>	сервис - возможные значения: <i>ntp</i> (смотри детали) ип - IP адрес или DNS имя (по умолчанию, используется IP/DNS узла сети) порт - номер порта (по умолчанию для сервиса используется стандартный номер порта).	Обратите внимание, для сервиса <i>tcp</i> обязательно нужно указывать порт. Проверка шифрованных протоколов (таких как IMAP на 993 порту или POP на 995 порту) в настоящее время не поддерживается. Как решение, пожалуйста, для подобных проверок используйте <code>net.tcp.service.perf[tcp,<ip>,порт]</code> . Сервисы <i>https</i> и <i>telnet</i> поддерживаются Zabbix начиная с версии 2.0. Назывался <i>tcp_perf</i> до Zabbix 2.0.
Проверка производительности	<code>net.udp.service.perf[service,<ip>,<port>]</code>	сервис - возможные значения: <i>ntp</i> (смотри детали) ип - IP адрес или DNS имя (по умолчанию, используется IP/DNS узла сети) порт - номер порта (по умолчанию для сервиса используется стандартный номер порта).	Пример: ⇒ <code>net.udp.service[ntp,,45]</code> → можно использовать для тестирования доступности NTP сервиса на 45 порту UDP.
			Этот элемент данных поддерживается начиная с Zabbix 3.0, но <i>ntp</i> сервис был доступен в <code>net.tcp.service[]</code> элементе данных и в предыдущих версиях.

Ключ				
▲	Описание	Возвращаемое значение	Параметры	Комментарии
		0.000000 - сервис недоступен	значения: <i>ntp</i> (смотри детали) ip - IP адрес или DNS имя (по умолчанию, используется IP/DNS узла сети) порт - номер порта (по умолчанию для сервиса используется стандартный номер порта).	можно использовать для тестирования времени ответа от NTP сервиса.
	UDP сервиса.	секунды - количество секунд прошедшее на ожидания ответа от сервиса		Этот элемент данных поддерживается начиная с Zabbix 3.0, но <i>ntp</i> сервис был доступен в <code>net.tcp.service[]</code> элементе данных и в предыдущих версиях.

Обработка времени ожидания

Zabbix не будет обрабатывать простую проверку дольше Timeout (времени ожидания) секунд, заданных в файле конфигурации Zabbix сервера/прокси.

ICMP пинг

Для обработки ICMP пинг Zabbix использует внешнюю утилиту **fping**.

Эта утилита не является частью дистрибутива Zabbix и должна быть установлена дополнительно. Если утилиты нет, у нее выставлены неверные разрешения и её размещение не совпадает с размещением заданным в файле конфигурации Zabbix сервера/прокси (параметры 'FpingLocation'), ICMP пинг (**icmpping**, **icmppingloss**, **icmppingsec**) не будет обрабатываться.

Смотрите также: [известные проблемы](#)

fping должен быть выполняемым под пользователем Zabbix демонов и должен иметь setuid root. Выполните эти команды из под **root** для выставления корректных разрешений:

```
shell> chown root:zabbix /usr/sbin/fping
shell> chmod 4710 /usr/sbin/fping
```

После выполнения этих двух команд выше проверьте владельца исполняемого файла **fping**. В некоторых случаях владелец может сброситься при выполнении chmod команды.

Также проверьте, принадлежит ли пользователь zabbix к группе zabbix, запустив команду:

```
shell> groups zabbix
```

и если нет добавьте следующей командой:

```
shell> usermod -a -G zabbix zabbix
```

Значения по умолчанию, ограничения и описания значений для параметров ICMP проверок:

Параметр	Ед. изм	Описание	Флаг у fping	Значения по умолчанию у fping	Разрешенные ограничения в Zabbix
пакеты	количество	количество требуемых пакетов -C к цели		3	1 10000
интервал	миллисекунды	время ожидания между успешными -р пакетами		1000	20 без ограничений

Параметр	Ед. изм	Описание	Флаг	Значения по умолчанию	Разрешенные ограничения в Zabbix	
			у fping	у fping	Zabbix мин	макс
размер	байты	размер пакета в байтах 56 байт в x86, 68 байт в x86_64	-b	56 или 68	24	65507
время ожидания	миллисекунды	fping v3.x - время ожидания после отправки последнего пакета (влияет с "-C" флагом) fping v4.x - отдельное время ожидания для каждого пакета	-t	fping v3.x - 500 fping v4.x - наследуется от -p флага, но не более чем 2000	50	без ограничений

Предупреждение: Значения по умолчанию для fping могут различаться в зависимости от платформы и версии - если сомневаетесь, проверьте документацию по fping.

Zabbix записывает проверяемые IP адреса во временный файл по всем трем *icmpping** ключам, который затем передается утилите **fping**. Если элементы данных имеют различные параметры ключа, то только элементы данных с идентичными параметрами ключа записываются в один файл.

Все записанные в один файл IP адреса проверяются fping утилитой в параллельном режиме, таким образом процесс Zabbix icmp pinger тратит фиксированное время вне зависимости от количества IP адресов в файле.

7.2.2.5.1 Ключи элементов данных для мониторинга VMware

Ключи элементов данных

Таблица содержит подробные сведения о простых проверках, которые можно использовать для мониторинга VMware сред.

▲	Описание	Возвращаемое значение	Ключ	Параметры	Комментарии
	Обнаружение кластеров VMware.	JSON объект	vmware.cluster.discovery[<url>]		
	Состояние кластера VMware.	Целое число 0 - серый; 1 - зеленый; 2 - желтый; 3 - красный.	vmware.cluster.status[<url>, <имя>]	url - URL службы VMware имя - имя кластера VMware	
	Журнал событий VMware.	Лог (журнал)	vmware.eventlog[<url>]	url - URL службы VMware	
	Полное имя сервиса VMware.	Строка	vmware.fullname[<url>]	url - URL службы VMware	
			vmware.hv.cluster.name[<url>,<uuid>]		

Ключ				
Описание	Возвращаемое значение	Параметры	Комментарии	
Имя кластера гипервизора VMware.	Строка	url - URL службы VMware uuid - имя хоста гипервизора VMware		
Использование процессора гипервизором VMware (Гц).	Целое число	url - URL службы VMware uuid - имя хоста гипервизора VMware		
Имя центра обработки данных гипервизора VMware.	Строка	url - URL службы VMware uuid - имя хоста гипервизора VMware		
Обнаружение хранилищ у гипервизора VMware	JSON объект	url - URL службы VMware uuid - имя хоста гипервизора VMware		
Среднее время операций чтения с хранилища данных (в Целое число ² миллисекундах)	Целое число	url - URL службы VMware uuid - имя хоста гипервизора VMware хранилище - имя хранилища режим - задержка (по умолчанию)		
Размер пространства VMware хранилища в байтах или в процентах от общего размера.	Целое число - для байт Число с плавающей точкой - для процентов	url - URL службы VMware uuid - имя хоста гипервизора VMware хранилище - имя хранилища режим - возможные значения: total (по умолчанию), free, pfree (свободного места, в процентах), uncommitted	Доступно начиная с Zabbix версий 3.0.6, 3.2.2	
Среднее время операций записи в хранилище данных (в Целое число ² миллисекундах)	Целое число	url - URL службы VMware uuid - имя хоста гипервизора VMware хранилище - имя хранилища режим - задержка (по умолчанию)		
Обнаружение гипервизоров VMware.	JSON объект	url - URL службы VMware		
Имя гипервизора VMware.	Строка	url - URL службы VMware uuid - имя хоста гипервизора VMware		

Описание	Возвращаемое значение	Параметры	Комментарии
Частота процессора гипервизора VMware (Гц).	Целое число	url - URL службы VMware uuid - имя хоста гипервизора VMware	
Модель процессора гипервизора VMware.	Строка	url - URL службы VMware uuid - имя хоста гипервизора VMware	
Количество ядер процессора у гипервизора VMware.	Целое число	url - URL службы VMware uuid - имя хоста гипервизора VMware	
Количество потоков процессора у гипервизора VMware.	Целое число	url - URL службы VMware uuid - имя хоста гипервизора VMware	
Общее количество памяти гипервизора VMware (в байтах).	Целое число	url - URL службы VMware uuid - имя хоста гипервизора VMware	
Модель гипервизора VMware.	Строка	url - URL службы VMware uuid - имя хоста гипервизора VMware	
UUID BIOS гипервизора VMware.	Строка	url - URL службы VMware uuid - имя хоста гипервизора VMware	
Имя поставщика гипервизора VMware.	Строка	url - URL службы VMware uuid - имя хоста гипервизора VMware	
Размер раздутой памяти гипервизора VMware (в байтах).	Целое число	url - URL службы VMware uuid - имя хоста гипервизора VMware	
Размер используемой памяти гипервизором VMware (в байтах).	Целое число	url - URL службы VMware uuid - имя хоста гипервизора VMware	
Статистика по входящему трафику на сетевом интерфейсе гипервизора VMware (в байтах в секунду).	Целое число 2	url - URL службы VMware uuid - имя хоста гипервизора VMware режим - bps	
		vmware.hv.network.out[<url>,<uuid>,<режим>]	

		Ключ		
	Описание	Возвращаемое значение	Параметры	Комментарии
▲	Статистика по исходящему трафику на сетевом интерфейсе гипервизора VMware (в байтах в секунду).	Целое число 2	url - URL службы VMware uuid - имя хоста гипервизора VMware режим - bps	
		vmware.hv.perfcounter[<url>,<uuid>,<путь>,<экземпляр>]	url - URL службы VMware uuid - имя хоста гипервизора VMware путь - путь к счетчику производительности 1 экземпляр - экземпляр счетчика производительности. Оставьте пустым для получения агрегированных значений (по умолчанию)	
	Значение счетчика производительности гипервизора VMware.	Целое число 2		Доступно начиная с Zabbix 2.2.9, 2.4.4
		vmware.hv.sensor.health.state[<url>,<uuid>]	url - URL службы VMware uuid - имя хоста гипервизора VMware	Доступно начиная с Zabbix 2.2.16, 3.0.6, 3.2.2
	Датчик накопительного состояния гипервизора VMware.	Целое число: 0 - серый; 1 - зеленый; 2 - желтый; 3 - красный		
		vmware.hv.status[<url>,<uuid>]	url - URL службы VMware uuid - имя хоста гипервизора VMware	Используется свойство общего состояния хост системы начиная с Zabbix 2.2.16, 3.0.6, 3.2.2
	Состояние гипервизора VMware.	Целое число: 0 - серый; 1 - зеленый; 2 - желтый; 3 - красный.		
		vmware.hv.uptime[<url>,<uuid>]	url - URL службы VMware uuid - имя хоста гипервизора VMware	
	Время работы гипервизора VMware (в секундах).	Целое число		
		vmware.hv.version[<url>,<uuid>]	url - URL службы VMware uuid - имя хоста гипервизора VMware	
	Версия гипервизора VMware.	Строка		
		vmware.hv.vm.num[<url>,<uuid>]	url - URL службы VMware uuid - имя хоста гипервизора VMware	
	Количество виртуальных машин на гипервизоре VMware.	Целое число		
		vmware.version[<url>]	url - URL службы VMware	
	Версия сервиса VMware.	Строка		
		vmware.vm.cluster.name[<url>,<uuid>]	url - URL службы VMware uuid - имя хоста виртуальной машины	
	Имя виртуальной машины VMware.	Строка		

Описание	Возвращаемое значение	Параметры	Комментарии
Количество процессоров на виртуальной машине VMware.	Целое число	VMware	
		vmware.vm.cpu.num[<url>,<uuid>]	url - URL службы VMware uuid - имя хоста виртуальной машины VMware
Время (в миллисекундах), когда виртуальная машина была готова, но не запланирована на выполнение на физическом CPU. Время готовности CPU зависит от количества виртуальных машин на хосте и их загрузки CPU (%).	Целое число 2	url - URL службы VMware uuid - имя хоста виртуальной машины VMware	Доступно начиная с Zabbix версии 3.0.0
		vmware.vm.cpu.ready[<url>,<uuid>]	
Использование процессора виртуальной машины VMware (Гц).	Целое число	url - URL службы VMware uuid - имя хоста виртуальной машины VMware	
		vmware.vm.cpu.usage[<url>,<uuid>]	
Имя центра обработки данных виртуальной машины VMware.	Строка	URL службы VMware uuid - имя хоста виртуальной машины VMware	
		vmware.vm.datacenter.name[<url>,<uuid>]	
Обнаружение виртуальных машин VMware.	JSON объект	url - URL службы VMware	
		vmware.vm.discovery[<url>]	
Имя гипервизора виртуальной машины VMware.	Строка	url - URL службы VMware uuid - имя хоста виртуальной машины VMware	
		vmware.vm.hv.name[<url>,<uuid>]	
Общий размер памяти виртуальной машины VMware (в байтах).	Целое число	url - URL службы VMware uuid - имя хоста виртуальной машины VMware	
		vmware.vm.memory.size[<url>,<uuid>]	
Размер раздутой памяти виртуальной машины VMware(в байтах).	Целое число	url - URL службы VMware uuid - имя хоста виртуальной машины VMware	
		vmware.vm.memory.size.balloon[<url>,<uuid>]	
Размер сжатой памяти виртуальной машины VMware(в байтах).	Целое число	url - URL службы VMware uuid - имя хоста виртуальной машины VMware	
		vmware.vm.memory.size.compressed[<url>,<uuid>]	

Описание	Возвращаемое значение	Параметры	Комментарии
Размер приватной памяти виртуальной машины VMware (в байтах).	vmware.vm.memory.size.private[<url>,<uuid>]	url - URL службы VMware uuid - имя хоста виртуальной машины VMware	VMware
Размер разделяемой памяти виртуальной машины VMware (в байтах).	vmware.vm.memory.size.shared[<url>,<uuid>]	url - URL службы VMware uuid - имя хоста виртуальной машины VMware	VMware
Размер памяти выгруженной в файл подкачки виртуальной машины VMware (в байтах).	vmware.vm.memory.size.swapped[<url>,<uuid>]	url - URL службы VMware uuid - имя хоста виртуальной машины VMware	VMware
Размер используемой памяти гостевой виртуальной машиной VMware (в байтах).	vmware.vm.memory.size.usage.guest[<url>,<uuid>]	url - URL службы VMware uuid - имя хоста виртуальной машины VMware	VMware
Размер используемой памяти гостевой виртуальной машиной VMware (в байтах).	vmware.vm.memory.size.usage.host[<url>,<uuid>]	url - URL службы VMware uuid - имя хоста виртуальной машины VMware	VMware
Обнаружение сетевых интерфейсов виртуальной машины VMware.	vmware.vm.net.if.discovery[<url>,<uuid>]	url - URL службы VMware uuid - имя хоста виртуальной машины VMware	VMware
Статистика по входящему трафику на сетевого интерфейсе виртуальной машины VMware (в байтах/пакетах в секунду).	vmware.vm.net.if.in[<url>,<uuid>,<экземпляр>,<режим>]	url - URL службы VMware uuid - имя хоста виртуальной машины VMware экземпляр - экземпляр сетевого интерфейса режим - bps (по умолчанию)/pps - байты/пакеты в секунду	2
Статистика по исходящему трафику на сетевом интерфейсе виртуальной машины VMware (в байтах/пакетах в секунду).	vmware.vm.net.if.out[<url>,<uuid>,<экземпляр>,<режим>]	url - URL службы VMware uuid - имя хоста виртуальной машины VMware экземпляр - экземпляр сетевого интерфейса режим - bps (по	2

Ключ				
Описание	Возвращаемое значение	Параметры	Комментарии	
Значение счетчика производительности у виртуальной машины VMware.	Целое число ²	умолчанию)/pps - байты/пакеты в секунду	vmware.vm.perfcounter[<url>,<uuid>,<путь>,<экземпляр>] url - URL службы VMware uuid - имя хоста гипервизора VMware путь - путь к счетчику производительности ¹ экземпляр - экземпляр счетчика производительности. Оставьте пустым для получения агрегированных значений (по умолчанию)	
Состояние питания виртуальной машины VMware.	Целое число: 0 - выключено; 1 - включено; 2 - приостановлено.	url - URL службы VMware uuid - имя хоста виртуальной машины VMware	Доступно начиная с Zabbix 2.2.9, 2.4.4	
Объем записанных данных на хранилище данных виртуальной машины VMware (в байтах).	Целое число	vmware.vm.storage.committed[<url>,<uuid>] url - URL службы VMware uuid - имя хоста виртуальной машины VMware		
Объем не записанных данных на хранилище данных виртуальной машины VMware (в байтах).	Целое число	vmware.vm.storage.uncommitted[<url>,<uuid>] url - URL службы VMware uuid - имя хоста виртуальной машины VMware		
Размер неразделяемого пространства хранилища виртуальной машины VMware (в байтах).	Целое число	vmware.vm.storage.unshared[<url>,<uuid>] url - URL службы VMware uuid - имя хоста виртуальной машины VMware		
Время непрерывной работы виртуальной машины VMware (в секундах).	Целое число	vmware.vm.uptime[<url>,<uuid>] url - URL службы VMware uuid - имя хоста виртуальной машины VMware		
Обнаружение дисковых устройств виртуальной машины VMware.	JSON объект	vmware.vm.vfs.dev.discovery[<url>,<uuid>] url - URL службы VMware uuid - имя хоста виртуальной машины VMware		
Статистика чтения с	Целое число ²	vmware.vm.vfs.dev.read[<url>,<uuid>,<экземпляр>,<режим>] url - URL службы VMware		

	Описание	Возвращаемое значение	Параметры	Комментарии
	дискового устройства виртуальной машины VMware (байт/операций в секунду).		uuid - имя хоста виртуальной машины VMware экземпляр - экземпляр дискового устройства режим - bps/ops - байт/операций в секунду	
		vmware.vm.vfs.dev.write[<url>,<uuid>,<экземпляр>,<режим>]	url - URL службы VMware uuid - имя хоста виртуальной машины VMware экземпляр - экземпляр дискового устройства режим - bps/ops - байт/операций в секунду	
	Статистика записи на дисковое устройство виртуальной машины VMware Целое число ² (байт/операций в секунду).			
		vmware.vm.vfs.fs.discovery[<url>,<uuid>]		
	Обнаружение файловых систем виртуальной машины VMware.	JSON объект	url - URL службы VMware uuid - имя хоста виртуальной машины VMware	Необходимо установить VMware Tools на гостевую виртуальную машину.
	Статистика файловой системы виртуальной машины VMware Целое число (в байтах/процентах).		fsname - имя файловой системы режим - total/free/used/pfree/pused	Необходимо установить VMware Tools на гостевую виртуальную машину.
		vmware.vm.vfs.fs.size[<url>,<uuid>,<fsname>,<режим>]		

Примечания

¹ Путь к счетчику производительности VMware имеет следующий формат **группа/счетчик[rollup]**, где:

- **группа** - группа счетчика производительности, например *cpu*
- **счетчик** - имя счетчика производительности, например *usagemhz*
- **rollup** - тип возвращаемого значения счетчика производительности, например *average*

Таким образом пример выше может быть следующим счетчиком: *cpu/usagemhz[average]*

Описания групп, имен счетчиков производительности и возвращаемые ими типы значений можно найти в [документации к VMware](#).

² Значения по этим элементам данных получаются от счётчиков производительности VMware и [параметр](#) **VMwarePerfFrequency** используется для обновления данных по этим элементам данных в Zabbix VMware кэше:

- *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 [мониторинг виртуальных машин](#) для получения более подробной информации о настройке Zabbix для мониторинга сред VMware.

6 Мониторинг файлов журналов

Обзор

Zabbix можно использовать для централизованного мониторинга и анализа файлов журналов с/без поддержки ротации журналов.

Можно использовать оповещения для предупреждения пользователей, когда файл журнала содержит конкретные строки или шаблоны строк.

Для наблюдения за файлом журнала у вас должно быть:

- Работающий Zabbix агент на узле сети
- Настроенный элемент данных для мониторинга журнала

Максимальный размер наблюдаемого файла журнала зависит от [поддержки файлов большого объема](#).

Настройка

Проверка параметров агента

Убедитесь, что в [файле конфигурации агента](#):

- Параметр 'Hostname' совпадает с именем узла сети в веб-интерфейсе
- Указаны сервера в параметре 'ServerActive' для обработки активных проверок

Настройка элемента данных

Настройте [элемент данных](#) для мониторинга журнала.

* Name	Log item
Type	Zabbix agent (active)
* Key	log[/var/log/syslog,error]
Type of information	Log
* Update interval	30s
* History storage period	3600
Log time format	ppppddphh:mm:ss

Все обязательные поля ввода отмечены красной звёздочкой.

Специально для элементов данных наблюдения за журналами вы должны указать:

Тип

Здесь выберите **Zabbix агент (активный)**.

Укажите:

log[/путь/к/файлу/имя_файла,<регулярное выражение>,<кодировка>,<макс. кол-во строк>,<режим>,<вывод>,<максзадержка>]

или

logrt[/путь/к/файлу/регулярное_выражение_описывающее_шаблон_имени_файла,<регулярное выражение>,<кодировка>,<макс. кол-во строк>,<режим>,<вывод>,<максзадержка>]

Zabbix агент фильтрует записи из файла журнала по регулярному выражению, если оно указано.

Если требуется только количество совпадающих строк укажите:

Ключ

log.count[/путь/к/файлу/имя_файла,<регулярное выражение>,<кодировка>,<макс. кол-во строк>,<режим>,< максзадержка >]

или

logrt.count[/путь/к/файлу/регулярное_выражение_описывающее_шаблон_имени_файла,<регулярное выражение>,<кодировка>,<макс. кол-во строк>,<режим>,< максзадержка >].

Убедитесь, что у файла имеются права на чтение для пользователя 'zabbix', в противном случае состояние элемента данных будет 'unsupported'.

Для получения более подробных сведенийсмотрите информацию о ключах **log**, **log.count**, **logrt** и **logrt.count** в разделе [поддерживаемых ключей элементов данных Zabbix агентом](#).

Выберите здесь Журнал (лог) для элементов данных **log** и **logrt** или Числовой (целое положительное) для элементов данных **log.count** и **logrt.count**.

Тип информации

Если используется опциональный параметр ВЫВОД, вы можете выбрать подходящий тип информации, отличный от "Журнал (лог)".

Обратите внимание, что выбор не журнального типа информации приведет к потере локального штампа времени.

Интервал обновления (в сек)

Этот параметр задает 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.

В этом поле вы можете опционально задать шаблон для анализа штампа времени строки журнала.

Если оставить пустым, штамп времени не будет анализироваться.

Поддерживаемые значения:

- * **y:** Год (0001-9999)
- * **M:** Месяц (01-12)
- * **d:** День (01-31)
- * **h:** Час (00-23)

Формат времени журнала

- * **m:** Минута (00-59)
- * **s:** Секунда (00-59)

Например, рассмотрим следующую строку из файла журнала Zabbix агента:

"23480:20100328:154718.045 Zabbix agent started. Zabbix 1.8.2 (revision 11211)."

Она начинается шестью символами обозначающими PID, далее следует дата, время, и остальная часть строки.

Форматом времени журнала для этой строки является
"rrrrrr:ууууММdd:hhmmss".

Обратите внимание, что символы "р" и ":" являются лишь заменителями и могут быть чем угодно, за исключением "уMdhms".

Важные замечания

- Сервер и агент следят за размером наблюдаемого журнала и временем последней модификации (для logrt) двумя счетчиками. Дополнительно:
 - Также агент использует номера inode (на UNIX/GNU/Linux), индексы файлов (на Microsoft Windows) и MD5 суммы первых 512 байт файла журнала для улучшения выбора в случае когда файлы журнала усекаются и ротируются.

- На системах UNIX/GNU/Linux предполагается, что файловые системы где хранятся файлы журналов, сообщают числа inode, которые могут быть использованы для слежения за состоянием файлов.
- На системах Microsoft Windows Zabbix агент определяет тип файловой системы на которой находятся файлы журналов:
 - На файловой системе NTFS 64-битные файловые индексы.
 - На файловых системах ReFS (только Microsoft Windows Server 2012) 128-битные файловые ID.
 - На файловых системах где файловые индексы меняются (т.е. FAT32, exFAT) используется запасной алгоритм для получения разумного подхода в неопределенных условиях, когда сжатие файла журнала приводит в результате к множеству файлов журналов с одинаковым временем изменения.
- Номера inode, индексы файлов и суммы MD5 собираются Zabbix агентом. Они не передаются Zabbix серверу и теряются в случае остановки Zabbix агента.
- Не меняйте время последней модификации файлов журналов, используя утилиту 'touch', не копируйте файл журнала с последующим восстановлением его имени (это изменит идентификатор иноды файла). В обоих случаях файл будет рассматриваться как другой и будет проанализирован с самого начала, что может привести к дубликатам оповещений.
- Если есть несколько совпадающих файлов журналов для элемента данных `logrt[]` и Zabbix агент следит за наиболее новым из них и этот более новый файл журнал удаляется, предупреждающее сообщение будет записано "`there are no files matching "<regexp mask>" in "<directory>"`". Zabbix агент игнорирует файлы журналы с временем изменения меньше чем последнее время модификации полученное агентом во время проверки элемента данных `logrt[]`.
- Агент начинает читать файл журнала с той позиции, на которой он остановился последний раз.
- Количество байт уже проанализированное (счётчик размера) и время последней модификации (счетчик времени) хранятся в базе данных Zabbix и отправляются агенту, для уверенности, что агент начнет читать файл журнала с этой позиции в случаях, когда агент только что был запущен или агент получил элементы данных, которые были ранее деактивированы или не поддерживались. Однако, если агент получает ненулевой размер счётчика от сервера, но элементы данных `logrt[]` или `logrt.count[]` не найдены и не удается найти соответствующие файлы, счётчик размера сбрасывается в 0, чтобы начать анализ сначала, если файлы появятся позже.
- Всякий раз, когда файл журнала становится меньше, чем значение счетчика размера известное агенту, счетчик обнуляется и агент начинает читать файл журнала с самого начала, принимая во внимание счетчик времени.
- Если есть несколько файлов журналов, с одинаковым последним временем модификации файла в соответствующей папке, агент пытается корректно проанализировать все файлы журналы с одинаковым временем модификации и избежать пропущенных данных или проанализировать данные дважды, несмотря на это невозможно охватить все возможные ситуации. Агент не предполагает какую либо определенную схему ротации файлов журналов, либо определяет ее. Когда есть несколько файлов журналов с одинаковым последним временем изменения, агент будет обрабатывать их лексикографически в порядке убывания. Таким образом, для некоторых схем ротации файлы журналы будут проанализированы в их оригинальном порядке. Для других же схем ротации журналов первоначальный порядок файла журнала не будет соблюдать, что может привести к получению найденных по шаблону строк файла журнала в измененном порядке (проблема не случится, если файлы журнала имеют разное время последнего изменения).
- Zabbix агент обрабатывает новые записи файла журнала один раз за *Период обновления* секунд.

- Zabbix агент отправляет не более чем **макс. кол-во строк** записей из файла журнала за секунду. Это ограничение предотвращает перегрузку сети и ресурсов процессора и переопределяет значение по умолчанию предусмотренное параметром **MaxLinesPerSecond** в [файле конфигурации агента](#).
- Для поиска необходимой строки Zabbix обрабатывает в 10 раза больше строк, чем указано в параметре **MaxLinesPerSecond**. Таким образом, например, если элемент данных `log[]` или `logrt[]` имеет *Интервал обновления* 1 секунда, по умолчанию агент будет анализировать не более чем 400 строк файла журнала и будет отправлять не более чем 200 совпадающих записей Zabbix серверу за одну проверку. Увеличением параметра **MaxLinesPerSecond** в файле конфигурации агента или указанием параметра **макс. кол-во строк** в ключе элемента данных, лимит можно увеличить вплоть до 10000 проанализированных записей в журнале и 1000 совпадающих записей для отправки Zabbix серверу за одну проверку. Если *Интервал обновления* указан значением в 2 секунды, лимиты для одной проверки могут быть увеличены в два раза больше, чем для *Интервала обновления* в 1 секунду.
- Кроме того, данные из файлов журналов всегда ограничены 50% размера буфера отправки у агента, даже если в буфере нет значений не связанных с данными из файлов журналов. Таким образом, значения **макс. кол-во строк** будут отправлены за одно соединение (а не в нескольких соединений), параметр [**BufferSize**](#) агента должен быть по крайней мере равен макс. кол-во строк x 2.
- При отсутствии данных для элементов данных журналов весь размер буфера используется для значений не связанных с данными из журналов. Когда появляются значения от файлов журналов они заменяют устаревшие данные не связанные с файлами журналов, если требуется, до максимального уровня 50%.
- Для записей в файле журнала длиннее 256КБ, только первые 256КБ сопоставляются с регулярным выражением, остальная часть игнорируется. Однако, если Zabbix агент был остановлен в процессе обработки длинной строчки, внутреннее состояние агента теряется и длинная строчка может быть проанализирована иначе после запуск агента.
- Специальное примечание для разделителей пути “\”: если формат файла представлен как “file\log”, тогда там не должно быть папки “file”, поскольку невозможно однозначно определить, экранируется ли это символ “.” или это первый символ в имени файла.
- Регулярные выражения для `logrt[]` поддерживаются только в именах файлов, совпадение регулярного выражения с папкой не поддерживается.
- В UNIX элементы данных `logrt[]` становятся НЕПОДДЕРЖИВАЕМЫМ, в случае если папка не существует где файл журнала должен был бы находиться.
- В Microsoft Windows, если папка не существует элемент данных не переводится в состояние НЕПОДДЕРЖИВАЕТСЯ (например, если в ключе элемента данных папка указана с ошибкой)
- Отсутствие файла журнала для элемента данных `logrt[]` не переводит его в состояние НЕПОДДЕРЖИВАЕТСЯ.
- Ошибки чтения файлов журналов для элемента данных `logrt[]` записываются в журнал агента как предупреждения, но не переводят элемент данных в состояние НЕПОДДЕРЖИВАЕТСЯ.
- Журнал Zabbix агента может быть очень полезен для поиска причин почему элементы данных `log[]` или `logrt[]` становятся НЕПОДДЕРЖИВАЕМЫМИ. Zabbix может мониторить свой файл журнала, за исключением случая когда он в режиме `DebugLevel=4`.

Извлечение совпадающей части регулярного выражения

Иногда мы можем захотеть извлечь только интересующие значения из требуемого файла вместо того, чтобы получать всю строку, в случае когда найдено совпадение с регулярным выражением.

Начиная с Zabbix 2.2.0, элементы данных файлов журналов расширены возможностью получения извлечения требуемых значений из строк файла. Добавился дополнительный параметр **'вывод'** у элементов данных **log** и **logrt**.

Использование параметра 'вывод' позволяет обозначить подгруппу совпадения в которой мы можем быть заинтересованы.

И так, например

```
log[/path/to/the/file,"large result buffer allocation.*Entries: ([0-9]+),,\1]
```

должно позволить получить количество записей со следующего содержания:

```
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
```

Причина, почему Zabbix вернет только одно число, потому что параметр 'вывод' здесь определен как \b ссылка только на первую интересующую подгруппу: **([0-9]+)**

Вместе с возможностью извлечения и получения числа, значение можно использовать в определениях триггеров.

Использование параметра максзадержка

Параметр 'максзадержка' в элементах данных журналов позволяет игнорировать более старые строки с целью получения наиболее новых строк проанализированных в течении "максзадержка" секунд.

Параметр 'maxdelay' > 0, может привести к **игнорированию важных записей в файлах журналов и пропуску оповещений**. Используйте этот параметр осторожно и на свой страх и риск, только в случае необходимости.

По умолчанию элементы данных мониторинга журналов забирают все новые строки появляющиеся в файлах журналов. Однако, имеются приложения, которые в некоторых ситуациях начинают записывать огромное количество сообщений в свои файлы журналов. Например, если база данных или DNS сервер недоступны, то такие приложения могут флудить файлы журналов тысячами практически идентичных сообщений об ошибке до тех пор пока не восстановится нормальный режим работы. По умолчанию, все эти сообщения добросовестно анализируются и совпадающие строки отправляются на сервер, как настроено в элементах данных **log** и **logrt**.

Встроенная защита от перегрузов состоит из настраиваемого параметра 'макс. кол-во строк' (защищающий сервер от слишком большого количества приходящих совпадающих строк в журнале) и ограничения в 4*'макс. кол-во строк' (защищает CPU и I/O хоста от перегрузки агентам одной проверкой). Тем не менее имеется 2 проблемы со встроенным механизмом защиты. Первая, на сервер будет отправлено большое количество потенциально не так информативных сообщений, которые займут место в базе данных. Вторая, по причине ограниченного количества строк анализируемых в секунду агент может отставать на часы от самых новых записей в журнале. Вполне вероятно, что вы захотите как можно быстрее быть информированным о текущей ситуации в файлах журналов вместо ковыряния часами старых записей.

Решение этих двух проблем является использование параметра 'максзадержка'. Если параметр 'maxdelay' > 0, во время каждой проверки измеряются количество обработанных байт, количество оставшихся байт и время обработки. Отталкиваясь от этих значений, агент вычисляет оценочную задержку - как много секунд может потребоваться, чтобы проанализировать все оставшиеся записи в файле журнала.

Если задержка не превышает 'максзадержка', тогда агент поступает с анализом файла журнала как обычно.

Если задержка больше чем 'максзадержка', тогда агент **игнорирует часть файла журнала, "перепрыгивая" эту часть** к новой оценочной позиции таким образом, чтобы оставшиеся строки можно было проанализировать за 'максзадержка' секунд.

Обратите внимание, что агент даже не читает проигнорированные строки в буфер, но вычисляет приблизительную позицию для прыжка в файле.

Сам факт пропуска строк в файле журнала записывается в файл журнала агента, примерно 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
```

Количество "to byte" является оценочным, потому что после "прыжка" агент скорректирует позицию в файл к началу строки в журнале, которая может быть в файле чуть дальше или раньше.

В зависимости от того как скорость роста соотносится к скорости анализа файла журнала, вы можете не увидеть "прыжков", а можете увидеть редкие или частые "прыжки", большие или маленькие "прыжки", или даже маленькие "прыжки" каждую проверку. Колебания загрузки системы и сетевые задержки также влияют на вычисления задержки и, следовательно, "прыжки" вперед чтобы не отставать от параметра "максзадержка".

Не рекомендуется указывать 'максзадержка' < 'интервал обновления' (это может привести к частым маленьким "прыжкам").

Заметки по обработке ротации 'copytruncate' файлов журналов

`logrt` с опцией `copytruncate` подразумевает, что разные файлы журналов имеют разные записи (по крайней мере штампы времени в них отличаются), поэтому MD5 суммы начальных блоков (до первых 512 байт) будут отличаться. Два файла с одинаковыми MD5 суммами начальных блоков означают, что один из них оригинал, а второй - копия.

`logrt` с опцией `copytruncate` делает попытку правильной обработки копий файлов журналов без дублирующих сообщений. Тем не менее, такие варианты как создание нескольких копий файлов журналов с одинаковыми штампами времени, ротация файлов журналов чаще чем интервал обновления `logrt[]` элемента данных, частый перезапуск агента не рекомендуются. Агент пытается справиться со всеми этими ситуациями, но хорошие результаты не гарантируются при всех обстоятельствах.

Действия, если произошла ошибка связи между агентом и сервером

Каждая совпадающая строка с элементов данных `log[]` и `logrt[]` и результат проверки каждого элемента данных `log.count[]` и `logrt.count[]` требует свободный слот в выделенной 50% области буфера отправки в агенте. Элементы буфера регулярно отправляются серверу (или прокси) и слоты буфера становятся снова пустыми.

Пока имеются свободные слоты в выделенной области для журналов в буфере отправки в агенте и связь между агентом и сервером (или прокси) нарушена, результаты мониторинга журналов накапливаются в буфере отправки. Такое поведение позволяет смягчить кратковременные нарушения связи.

Во время длительных нарушений связи все слоты журналов становятся занятыми и выполняются следующие действия:

- Проверки элементов данных `log[]` и `logrt[]` останавливаются. Когда связь восстановится и появятся свободные слоты, проверки вернутся к предыдущей позиции. Не совпадающие строки потеряются. Совпадающие строки не будут потеряны, они просто отправятся позже.

- Проверки `log.count[]` и `logrt.count[]` останавливаются, если `maxdelay = 0` (по умолчанию). Поведение похоже на элементы данных `log[]` и `logrt[]`, описанное выше. Обратите внимание, что потеря связи может повлиять на результаты `log.count[]` и `logrt.count[]`: например, одна проверка насчитает 100 совпадающих строк в файле журнала, но по причине отсутствия свободных слотов в буфере проверка будет остановлена. Когда связь восстановится агент насчитает те же 100 совпадающих строк, а также 70 новых совпадающих строк. После чего агент отправит количество = 170, так как они найдены за одну проверку.
- Проверки `log.count[]` и `logrt.count[]` при `maxdelay > 0`: если не было “прыжка” во время проверки, тогда поведение аналогично описанному выше. Если всё же был “прыжок” через строки файла журнала, тогда позиция после “прыжка” сохранится и подсчитанный результат будет отброшен. Таким образом, агент пытается не отставать от увеличивающегося файла журнала, даже в случае проблем со связью.

7.2.2.7 Вычисляемые элементы данных

1 Обзор

При помощи вычисляемых элементов данных вы можете выполнять подсчеты на основании других элементов данных.

Таким образом, вычисляемые элементы данных являются одним из способов создания виртуальных источников данных. Значения будут периодически вычисляться на основе арифметического выражения. Все вычисления выполняются Zabbix сервером - Zabbix агенты и прокси не участвуют в подсчете вычисляемых элементов данных и не имеют никаких данных для этого.

Полученные данные записываются в базу данных Zabbix, как и для любого другого элемента данных - это означает хранение значений истории и динамики изменений для быстрой генерации графиков. Созданные элементы данных можно использовать в выражениях триггеров, сославшись макросами или другими объектами, также как и на любой другой тип элементов данных.

Для использования вычисляемых элементов данных, выберите тип элемента данных **Вычисляемое**.

2 Настраиваемые поля

Ключ уникальный идентификатор элемента данных (в пределах узла сети). Вы можете создать любое имя ключа используя поддерживаемые символы.

Формула расчета необходимо ввести в поле **Формула**. Не существует практически никакой связи между формулой и ключем. В любом случае параметры ключа не используются в формуле.

Корректный синтаксис простой формулы:

`функция(<ключ>|<узел сети:ключ>, <параметр1>, <параметр2>, ...)`

Где:

АРГУМЕНТ	ОПРЕДЕЛЕНИЕ
функция	Одна из функций поддерживаемых в выражениях триггеров: <code>last</code> , <code>min</code> , <code>max</code> , <code>avg</code> , <code>count</code> и остальные
ключ	Ключ другого элемента данных, данные которого вы хотите использовать. Его можно задать как ключ или узел сети:ключ . <i>Обратите внимание:</i> Настоятельно рекомендуется заключать весь ключ в двойные кавычки (“...”), во избежании неправильного разбора из-за пробелов или запятых в ключе. Также если в ключе имеются параметры заключенные в кавычки, то двойные кавычки должны быть экранированы с помощью обратной косой чертой (\). Смотрите ниже Пример 5 .

АРГУМЕНТ

ОПРЕДЕЛЕНИЕ

параметр(ы) Параметр(ы) функций, если требуются.



Все элементы данных, на которые ссылается формула вычисляемого элемента данных, должны существовать и собирать данные ([исключения в функциях и неподдерживаемых элементах данных](#)). Также, если вы измените ключ элемента данных на который ссылается элемент данных, то вы должны вручную обновить все формулы в которых используется этот ключ.



[Пользовательские макросы](#) в формуле будут раскрыты, если используется ссылка на параметр функции или константу. Пользовательские макросы НЕ будут раскрыты, если используется ссылка на функцию, имя узла сети, параметры ключей элементов данных или оператор.

Для более сложной формулы можно использовать комбинацию функций, операторов и скобок. Вы можете использовать все функции и [операторы](#), которые поддерживаются в выражениях триггеров. Обратите внимание, что синтаксис немного отличается, однако логика и приоритет операций в точности такой же.

В отличии от выражений триггеров, Zabbix обрабатывает вычисляемые элементы данных в соответствии со временем обновления элемента данных, а не при получении нового значения.



Если результатом вычисления будет число с плавающей точкой, оно будет обрезано до целого числа, в случае если типом информации элемента данных указан Числовой (целое положительное).

Вычисляемый элемент данных может перейти в неподдерживаемое состояние в нескольких случаях:

1. элемент(ы) данных, на которые имеется ссылка
 - не найден
 - деактивирован
 - принадлежит деактивированному узлу сети
 - не поддерживается (смотрите исключения в разделах [Функции и неподдерживаемые элементы данных](#), [Выражения с неподдерживаемыми элементами данных и неизвестными значениями](#) и [Операторы](#))
2. нет данных для вычисления функции
3. деление на ноль
4. использован некорректный синтаксис

Поддержка вычисляемых элементов данных была введена в Zabbix 1.8.1.

Начиная с Zabbix 3.2 вычисляемые элементы данных в некоторых случаях могут использовать неподдерживаемые элементы данных как описано в [Функции и неподдерживаемые элементы данных](#), [Выражения с неподдерживаемыми элементами данных и неизвестными значениями](#) и [Операторы](#).

3 Примеры использования

Пример 1

Вычисление процентного отношения свободного места на диске для '/'.

Используйте функцию **last**:

```
100*last("vfs.fs.size[/,free]"/last("vfs.fs.size[/,total]"))
```

Zabbix будет брать последние значения полного и свободного объема диска и вычислять процентное отношение в соответствии с данной формулой.

Пример 2

Вычисление 10-минутного усреднения количества значений обрабатываемых Zabbix.

Используйте функцию **avg**:

```
avg("Zabbix Server:zabbix[wcache,values]", 600)
```

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]"))
```

Пример 5

В вычисляемом элементе данных возможно использование агрегированных элементов данных.

Возьмите на заметку, каким образом двойные кавычки экранируются в заключенном кавычками ключе:

```
last("grpsum[\"video\", \"net.if.out[eth0,bytes]\", \"last\", \"0\"]") / last("grpsum[\"video\", \"nginx_stat.sh[active]\", \"last\", \"0\"]")
```

7.2.2.8 Внутренние проверки

Обзор

Внутренние проверки позволяют наблюдать за внутренним процессами Zabbix. Другими словами, вы можете наблюдать что происходит с Zabbix сервером или Zabbix прокси.

Внутренние проверки вычисляются:

- на Zabbix сервере - если узел сети наблюдается через сервер
- на Zabbix прокси - если узел сети наблюдается через прокси

Внутренние проверки обрабатываются сервером или прокси вне зависимости от состояния обслуживания узла сети (начиная с Zabbix 2.4.0).

Для использования этого элемента данных выберите тип элемента данных **Zabbix внутренний**.

Внутренние проверки обрабатываются Zabbix поллерами.

Поддерживаемые ключи

- Параметры без угловых скобок являются константами - например, 'host' и 'available' в `zabbix[host,<тип>,available]`. Используйте их в ключе элемента данных без изменения (*как есть*).
- Значения элементов данных и параметры элементов данных, которые “не поддерживаются на прокси” можно собирать только на узлах сети, которые наблюдаются через сервер. И наоборот,

значения “не поддерживается на сервере” можно собирать только, если узел сети наблюдается через прокси.

▲	Описание	Ключ	Возвращаемое значение	Комментарии
	Время запуска процесса Zabbix сервера в секундах.	zabbix[boottime]	Целое число.	
	Количество значений хранимых в таблице HISTORY	zabbix[history]	Целое число.	Не используйте с MySQL InnoDB, Oracle или PostgreSQL! <i>(не поддерживается прокси)</i>
	Количество значений хранимых в таблице HISTORY_LOG	zabbix[history_log]	Целое число.	Не используйте с MySQL InnoDB, Oracle или PostgreSQL! Этот элемент данных поддерживается начиная с версии 1.8.3 . <i>(не поддерживается прокси)</i>
	Количество значений хранимых в таблице HISTORY_STR	zabbix[history_str]	Целое число.	Не используйте с MySQL InnoDB, Oracle или PostgreSQL! <i>(не поддерживается прокси)</i>
	Количество значений хранимых в таблице HISTORY_TEXT	zabbix[history_text]	Целое число.	Не используйте с MySQL InnoDB, Oracle или PostgreSQL! Этот элемент данных поддерживается начиная с версии 1.8.3 . <i>(не поддерживается прокси)</i>
	Количество значений хранимых в таблице HISTORY_UINT	zabbix[history_uint]	Целое число.	Не используйте на MySQL InnoDB, Oracle или PostgreSQL! Этот элемент данных поддерживается начиная с версии 1.8.3 . <i>(не поддерживается прокси)</i>
	Количество активированных элементов данных (поддерживаемых и неподдерживаемых) у узла сети.	zabbix[host,,items]	Целое число.	Этот элемент данных поддерживается начиная с Zabbix 3.0.0 .
	Количество активированных неподдерживаемых элементов данных у узла сети.	zabbix[host,,items_unsupported]	Целое число.	Этот элемент данных поддерживается начиная с Zabbix 3.0.0 .

Ключ	Описание	Возвращаемое значение	Комментарии
zabbix[host,,maintenance]	Возвращает текущее состояние обслуживания узла сети.	0 - узел сети в нормальном состоянии, 1 - узел сети в обслуживании со сбором данных, 2 - узел сети в обслуживании без сбора данных.	Данный элемент данных всегда обрабатывается Zabbix сервером вне зависимости от настроек узла сети (мониторится через сервер или прокси). Прокси не получает этот элемент данных при получении своей конфигурации. Второй параметр должен быть пустым и зарезервирован для использования в будущем. Данный элемент данных поддерживается начиная с Zabbix 2.4.0.
zabbix[host,discovery,interfaces]	Детали по всем добавленным интерфейсам к узлу сети в веб-интерфейсе Zabbix.	JSON объект.	Этот элемент данных можно использовать в низкоуровневом обнаружении . Этот элемент данных поддерживается начиная с Zabbix 3.4.0. <i>(не поддерживается прокси)</i>
zabbix[host,<тип>,available]	Доступность определенного типа проверок на узле сети. Значение этого элемента соответствует иконкам доступности в списке узлов сети.	0 - недоступен, 1 - доступен, 2 - неизвестно.	Допустимые типы: agent , snmp , ipmi , jmx . Значение элемента данных вычисляется согласно параметрам конфигурации доступности/недоступности соответствующего узла сети.
zabbix[hosts]	Количество наблюдаемых узлов сети	Целое число.	Этот элемент данных поддерживается начиная с Zabbix 2.0.0.
zabbix[items]	Количество активированных элементов данных (поддерживаемых и не поддерживаемых)	Целое число.	Данный элемент данных поддерживается начиная с Zabbix 2.2.0.
zabbix[items_unsupported]	Количество не поддерживаемых элементов данных	Целое число.	

	Описание	Ключ	
		Возвращаемое значение	Комментарии
		zabbix[java,,<параметр>]	
	Получение информации связанной с Zabbix Java gateway.	<p>Если <параметром> является ping, возвращается “1”. Можно использовать для проверки доступности Java gateway, используя функцию триггера <code>nodata()</code>.</p> <p>Если <параметром> является version, возвращается версия Java gateway. Пример: “2.0.0”.</p>	<p>Второй параметр должен быть пустым, т.к. зарезервирован для использования в будущем.</p> <p>Этот элемент данных поддерживается начиная с Zabbix 2.0.0.</p>
		zabbix[preprocessing_queue]	
	Количество значений, помещенных в очередь предварительной обработки.	Целое число.	<p>Этот элемент данных можно использовать для мониторинга размера очереди предварительной обработки.</p> <p>Этот элемент данных поддерживается начиная с 3.4.0.</p>
		zabbix[process,<тип>,<режим>,<состояние>]	
	Время конкретного процесса Zabbix или группы процессов (указываются с помощью <тип> и <режим>), потраченное на <состояние> в процентах. Расчет производится только за последнюю минуту.	<p>Если в поле <режим> указан номер процесса Zabbix, который не запущен (например, при 5 запущенных поллерах, в <режиме> указано 6), такой элемент данных переходит в состояние <i>не поддерживается</i>. Минимум и максимум относится к процентному использованию одним процессом. Таким образом, если в группе из 3 поллеров процентное использование составило 2, 18 и 66, тогда min должен вернуть 2, а max должен вернуть 66. Процессы сообщают то, что они делают, используя разделяемую память, и процесс самодиагностики резюмирует эти данные каждую секунду. Изменения состояний (<i>busy/idle</i>) регистрируются при изменении - таким образом,</p>	<p>На данный момент поддерживаются следующие типы процессов:</p> <ul style="list-style-type: none"> alerter - процесс отправки уведомлений (<i>не поддерживается прокси</i>) alert manager - менеджер задач оповещения configuration syncer - процесс управления кэшем данных конфигурации в оперативной памяти data sender - процесс отправки данных с прокси (<i>не поддерживается сервером</i>) discoverer - процесс обнаружения устройств escalator - процесс эскалации действий (<i>не поддерживается прокси</i>) heartbeat sender - процесс

Ключ

▲	Описание	Возвращаемое значение	Комментарии
	<p>процессы, которые становятся занятыми, регистрируют это состояние и не меняют и не обновляют его пока процесс не станет свободным. Такое поведение гарантирует, что даже полностью зависшие процессы будут корректно зарегистрированы как 100% занятые. В настоящее время, “busy” означает “не спит”, но в будущем могут добавиться дополнительные состояния - ожидание блокировки, выполнение запросов в базу данных, и т.д.</p> <p>В Linux и в большинстве других систем, точность составляет 1/100 секунды.</p>		уведомления сервера прокси сервером о состоянии прокси (<i>не поддерживается сервером</i>)
			history syncer - процесс, который записывает историю в БД
			housekeeper - процесс удаления старых данных истории
			http poller - поллер веб-мониторинга
			icmp pinger - поллер проверок icmp ping
			ipmi manager - менеджер IPMI поллеров
			ipmi poller - поллер для проверок по IPMI
			java poller - поллер для Java проверок
			poller - обычный поллер для пассивных проверок
			preprocessing manager - менеджер задач предобработки
			preprocessing worker - процесс предобработки данных
			proxy poller - поллер для пассивных прокси (<i>не поддерживается прокси</i>)
			self-monitoring - процесс сбора внутренней статистики сервера
			snmp trapper - траппер сбора/обработки SNMP трапов
			task manager - процесс для удаленного выполнения задач, которые запрашиваются другими компонентами (например, возможности закрытия проблем, подтверждения проблемы, принудительной проверки значения элемента данных, удаленной команды)
			timer - процесс обработки обслуживаний
			trapper - процесс-улавливатель для активных проверок, трапов, связей между нодами и прокси
			unreachable poller - поллер недоступных устройств



Ключ

Описание	Возвращаемое значение	Комментарии
	vmware collector - коллектор данных VMware, ответственный за сбор данных со служб VMware	
	Примечание: Вы также можете увидеть все эти типы процессов в файле журнала сервера.	
	Допустимые режимы: avg - среднее значение по всем процессам указанного типа (по умолчанию) count - количество форков указанного типа процесса, <состояние> не должно быть указано max - максимальное значение min - минимальное значение <номер процесса> - номер процесса (от 1 до количества префорк процессов). Например, если запущено 4 траппера, тогда значение от 1 до 4.	
	Допустимые состояния: busy - процесс в занятом состоянии, например, обработка запроса (по умолчанию). idle - процесс в свободном состоянии, ничего не делающий.	
	Примеры: ⇒ zabbix[process,poller,avg,busy] → среднее время, потраченное процессами поллеров, которые что-либо делали за последнюю минуту ⇒ zabbix[process,"icmp pinger",max,busy] → максимальное время, потраченное любыми процессами ICMP pinger, которые что-либо делали за последнюю минуту ⇒ zabbix[process,"history syncer",2,busy] → время, потраченное на что-то	



Ключ

Описание

Возвращаемое значение

Комментарии

Информация доступности по Zabbix прокси.

Целое число.

процессом синхронизации истории номер 2 в течении последней минуты
⇒

zabbix[process,trapper,count]
→ общее количество запущенных процессов
trapper

Этот элемент данных поддерживается начиная с Zabbix **1.8.5**.

zabbix[proxy,<имя>,<параметр>]

<имя> - Имя прокси
Список поддерживаемых параметров (<параметр>):
lastaccess – штамп времени последнего сообщения о доступности от прокси

Например, ⇒
zabbix[proxy,"Germany",lastaccess]

Можно использовать [функцию триггеров fuzzytime\(\)](#) для проверки доступности прокси.
Начиная с Zabbix 2.4.0 этот элемент данных всегда обрабатывается Zabbix сервером независимо от настроек узла сети (наблюдается через сервер или прокси).

zabbix[proxy_history]

Количество значений в таблице истории прокси ожидающих отправку на сервер

Целое число.

Данный элемент данных поддерживается начиная с Zabbix **2.2.0**
(не поддерживается сервером)

zabbix[queue,<от>,<до>]

Количество наблюдаемых элементов данных в очереди, которые задерживаются <от> и <до> секунд включительно.

Целое число.

<от> - по умолчанию: 6 секунд
<до> - по умолчанию: бесконечно
Для этих параметров поддерживаются [символы времени](#) (s,m,h,d,w).
Параметры от и до поддерживаются начиная с Zabbix **1.8.3**.

	Описание	Ключ	
	Возвращаемое значение	Комментарии	
▲			
Статистика доступности кэша конфигурации	zabbix[rcache,<кэш>,<режим>]	Целое число (для размера): Целое с плавающей точкой (для процентов).	Кэш: buffer Режим: total - полный размер буфера free - размер свободного места в буфере pfree - процент свободного места в буфере used - размер использованного места в буфере
Требуемое быстродействие Zabbix сервера, имеется ввиду новые значения в секунду.	zabbix[requiredperformance]	Целое с плавающей точкой.	Приблизительно соответствует значению “Требуемое быстродействие сервера, новые значения в секунду” со страницы <i>Отчеты</i> → <u>Информация о системе</u> . Поддерживается начиная с Zabbix 1.6.2 .
Количество значений хранимых в таблице TRENDS	zabbix[trends]	Целое число.	Не используйте с MySQL InnoDB, Oracle или PostgreSQL! (не поддерживается прокси)
Количество значений хранимых в таблице TRENDS_UINT	zabbix[trends_uint]	Целое число.	Не используйте с MySQL InnoDB, Oracle или PostgreSQL! Этот элемент данных поддерживается начиная с Zabbix 1.8.3 . (не поддерживается прокси)
Количество активированных триггеров в базе данных Zabbix, со всеми активированными элементами данных, которые упомянуты в выражениях триггеров на активированных узлах сети.	zabbix[triggers]	Целое число.	(не поддерживается прокси)
Время непрерывной работы процесса Zabbix сервера в секундах.	zabbix[uptime]	Целое число.	(не поддерживается прокси)
Статистика доступности кэша значений.	zabbix[vcache,buffer,<режим>]	Целое число (для размера); Целое с плавающей точкой (для процентов).	Режим: total - полный размер буфера free - размер свободного места в буфере pfree - процент свободного



Ключ

Описание

Возвращаемое значение

Комментарии

Статистика эффективности кэша значений Zabbix.

zabbix[vcache,cache,<параметр>]

Целое число.

С параметром равным **mode**:
0 - нормальный режим,
1 - режим низкой памяти

места в буфере
used - размер использованного места в буфере
pused - процент использованного места в буфере

Данный элемент данных поддерживается с Zabbix **2.2.0**. (*не поддерживается прокси*)

Параметр:
requests - общее количество запросов
hits - количество попаданий в кэш (история берется из кэша)
phits - процент попаданий в кэш
misses - количество непопаданий в кэш (история берется из базы данных)
mode - режим работы кэша значений

Данный элемент данных поддерживается с Zabbix **2.2.0** и параметр **mode** поддерживается начиная с Zabbix **3.0.0**. (*не поддерживается прокси*)

Вы можете использовать этот ключ с шагом предобработки *Изменение в секунду* при необходимости получения статистики значений в секунду.

zabbix[vmware,buffer,<режим>]

Статистика доступности кэша VMware.

Целое число (для размера);
Целое с плавающей точкой (для процентов).

free - размер свободного места в буфере
pfree - процент свободного места в буфере
used - размер использованного места в буфере
pused - процент использованного места в

Режим:
total - полный размер буфера

Ключ

	Описание	Возвращаемое значение	Комментарии
	zabbix[<i>wcache</i>,<кэш>,<режим>]		буфере
	Статистика и доступность кэша записи Zabbix.		Данный элемент данных поддерживается с Zabbix 2.2.0.
Кэш Режим			
	values		
	all (по умолчанию)	Количество значений обрабатываемых Zabbix сервером или Zabbix прокси, исключая не поддерживающие	Целое число.
	float	Количество обработанных значений с плавающей точкой.	Целое число.
	uint	Количество обработанных целочисленных значений.	Целое число.
	str	Количество обработанных символьных/строковых значений.	Целое число.
	log	Количество обработанных значений из файлов журналов.	Целое число.
	text	Количество обработанных текстовых значений.	Целое число.
	not supported	Количество неподдерживаемых элементов данных.	Целое число. Режим <i>not supported</i> поддерживается начиная с Zabbix 1.8.6.
history			
	pfree (по умолчанию)	Свободное место в буфере истории в процентах.	Число с плавающей точкой.
	index		
	free	Размер свободного места в буфере истории.	Целое число.
	total	Полный размер буфера.	Целое число.
	used	Размер занятого места в буфере истории.	Целое число.
	pfree (по умолчанию)	Свободное место в буфере индексов истории.	Целое с плавающей точкой.
			Кэш индексов истории используется для индексирования значений записываемых в кэш истории. Кэш индекса

		Ключ	
	Описание	Возвращаемое значение	Комментарии
	free	Свободное место в буфере индексов истории.	Целое число.
	total	Полный размер буфера индексов истории.	Целое число.
	used	Размер занятого места в буфере индексов истории.	Целое число.
	pfree (по умолчанию)	Свободное место в буфере динамики изменений в процентах.	Число с плавающей точкой. Кэш динамики изменений агрегируется за текущий час для всех элементов данных, которые получают данные. (не поддерживается прокси)
trend	free	Свободное место в буфере динамики изменений.	Целое число. (не поддерживается прокси)
	total	Полный размер буфера динамики изменений.	Целое число. (не поддерживается прокси)
	used	Размер занятого места в буфере динамики изменений.	Целое число (не поддерживается прокси)

7.2.2.9 Проверки через SSH

Обзор

SSH проверки выполняются без какого-либо агента. Zabbix агент не требуется для проверок выполняемых по SSH.

Для выполнения SSH проверок Zabbix сервер должен быть [изначально сконфигурирован](#) с поддержкой SSH2.

Минимально поддерживаемой версией библиотеки libssh2 является версия 1.0.0.

Настройка

Аутентификация парольной фразой

Проверки SSH предоставляют два метода аутентификации, пара логин пользователя/пароль и на основе ключа-файла.

Если вы не собираетесь использовать ключ, то никакой дополнительной настройки не требуется, при компиляции из исходных кодов, необходима также привязка библиотеки libssh2 к Zabbix.

Аутентификация на базе ключа

Для использования элементов данных SSH на основе аутентификации по ключу необходимо произвести некоторые изменения в конфигурации сервера.

Откройте файл конфигурации Zabbix сервера ([zabbix_server.conf](#)) из под `root` и найдите следующую строку:

```
# SSHKeyLocation=
```

Раскомментируйте её и укажите полный путь к папке, где размещены публичные и приватные ключи:
`SSHKeyLocation=/home/zabbix/.ssh`

Затем сохраните файл и перезапустите zabbix_server.

Где `/home/zabbix` домашняя папка для аккаунта `zabbix` пользователя и `.ssh` папка, куда будут по умолчанию сгенерированы с помощью команды [ssh-keygen](#) публичные и приватные ключи.

Обычно при установке пакетов zabbix-server на разных дистрибутивах ОС создается аккаунт `zabbix` пользователя с домашней папкой в не очень известных местах (как для системных аккаунтов). Например, для CentOS папка `/var/lib/zabbix`, для Debian она `/var/run/zabbix`.

До начала генерирования ключей, рассмотрите вариант перемещения домашней папки в более известное место (интуитивно ожидаемое). Этот вариант будет соответствовать параметру *SSHKeyLocation* конфигурации Zabbix сервера, упомянутого выше.

Эти шаги можно пропустить, если аккаунт `zabbix` добавлен вручную в соответствии с [разделом установки](#), потому что в этом случае домашняя папка, скорее всего, уже расположена в `/home/zabbix`.

Для изменения этой настройки у аккаунта zabbix пользователя все работающие процессы, которые его используют должны быть остановлены:

```
# service zabbix-agent stop  
# service zabbix-server stop
```

Для изменения размещения домашней папки с попыткой переместить её (если папка существует), вы должны выполнить команду:

```
# usermod -m -d /home/zabbix zabbix
```

Вполне возможно, что домашняя папка не существует в старом месте (в CentOS, например), поэтому её необходимо создать в новом месте. Безопасная попытка, чтобы сделать это:

```
# test -d /home/zabbix || mkdir /home/zabbix
```

Чтобы быть уверенным что всё безопасно, можно выполнить дополнительные команды для установки разрешений к домашней папке:

```
# chown zabbix:zabbix /home/zabbix  
# chmod 700 /home/zabbix
```

Теперь можно запустить ранее остановленные процессы:

```
# service zabbix-agent start  
# service zabbix-server start
```

Теперь шаги генерирования публичных и приватных ключей можно выполнить командой:

```
# 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):
Created directory '/home/zabbix/.ssh'.
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 . . . * .               |
|                           00 0+ |
```

+-----+

Возьмите на заметку: публичные и приватные ключи (*id_rsa.pub* и *id_rsa* соответственно) генерируются по умолчанию в папку */home/zabbix/.ssh*, которая соответствует параметру конфигурации *SSHKeyLocation* Zabbix сервера.

Ключи других типов кроме “rsa” могут быть использованы, если поддерживаются утилитой *ssh-keygen* и библиотекой *libssh2*, которая используется Zabbix.

Диалог настройки командной строки

Этот шаг необходимо выполнить только один раз на каждом хосте, который будет наблюдаться с использованием SSH проверок.

При использовании следующей команды, файл **публичного** ключа будет установлен на удаленный хост *10.10.10.10*, для того чтобы потом можно было выполнять SSH проверки при помощи аккаунта *root*:

```
# sudo -u zabbix ssh-copy-id root@10.10.10.10
The authenticity of host '10.10.10.10 (10.10.10.10)' can't be established.
RSA key fingerprint is 38:ba:f2:a4:b5:d9:8f:52:00:09:f7:1f:75:cc:0b:46.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added '10.10.10.10' (RSA) to the list of known hosts.
root@10.10.10's password:
Теперь попытайтесь зайти на машину с помощью "ssh 'root@10.10.10.10'" и проверьте там:
  .ssh/authorized_keys
чтобы убедиться, что мы не добавили дополнительные ключи, которые нежелательны.
```

Теперь можно проверить вход по SSH с использованием приватного ключа по умолчанию (*/home/zabbix/.ssh/id_rsa*) у аккаунта *zabbix* пользователя:

```
# sudo -u zabbix ssh root@10.10.10.10
```

Если вход успешен, то часть настройки в командной строке завершена и удаленное соединение по SSH можно закрыть.

Настройка элемента данных

Фактически выполняемые команда(ы) необходимо поместить в поле **Выполняемый скрипт** при настройке элемента данных.

Можно выполнять несколько команд одну за другой, размещая каждую на новой строке. В этом случае возвращаемые значения также будут отформатированы как многострочные.

* Name	SSH test check (whithout passphrase)
Type	SSH agent
* Key	ssh.run[clear]
* Host interface	127.0.0.1 : 10051
Authentication method	Public key
* User name	root
* Public key file	id_rsa.pub
* Private key file	id_rsa
Key passphrase	
* Executed script	service mysql-server status
Type of information	Numeric (unsigned)
Units	
* Update interval	30s

Все обязательные поля ввода отмечены красной звёздочкой.

Поля, которые требуют специфичную информацию SSH элементов данных:

Параметр элемента данных	Описание	Комментарии
Тип	Здесь выберите SSH агент.	<уникальное короткое описание> обязательно и должно быть уникальным у всех элементов данных типа SSH в рамках одного узла сети
Ключ	Уникальный (в пределах узла сети) ключ элемента данных в формате ssh.run[<уникальное короткое описание>,<ip>,<порт>,<кодировка>]	Порт 22 по умолчанию, а не порт указанный в интерфейсе узла сети к которому этот элемент данных назначен
Метод аутентификации	Один из “Пароль” или “Публичный ключ”	
Имя пользователя	Имя пользователя для аутентификации на удаленном хосте. Требуется	
Файл публичного ключа	Имя файла публичного ключа, если Метод аутентификации “Публичный ключ”. Требуется	Пример: <i>id_rsa.pub</i> - имя файла публичного ключа по умолчанию сгенерированного командой ssh-keygen
Файл приватного ключа	Имя файла приватного ключа, если Метод аутентификации “Публичный ключ”. Требуется	Пример: <i>id_rsa</i> - имя файла приватного ключа
Пароль или Парольная фраза	Пароль при аутентификации или Парольная фраза, если была	Оставьте поле Парольная фраза пустым, если фраза не используется

Параметр элемента данных	Описание	Комментарии
	использована фраза для приватного ключа	Смотрите также известные проблемы по поводу использования парольных фраз
Выполняемый скрипт	Выполняемые команды командной строки при использовании удаленной сессии SSH	Примеры: <i>date +%s service mysql-server status ps auxww grep httpd wc -l</i>



Библиотека libssh2 может обрезать выполняемые скрипты до ~32КБ

7.2.2.10 Проверки через Telnet

Обзор

Telnet проверки выполняются без какого-либо агента. Zabbix агент не нужен для проверок через Telnet.

Настраиваемые поля

Действующая команда(ы) для выполнения должны быть размещены в поле **Выполняемый скрипт** при настройке элемента данных.

Можно выполнять несколько команд одну за другой, размещая каждую на новой строке. В этом случае результирующее значение также будет отформатировано как многострочное.

Поддерживаемые символы, которыми строка может заканчиваться:

- \$
- #
- >
- %



Строка telnet с запросом, которая заканчивается одним из этих символов, будет удалена из возвращаемого значения, но только для первой команды из списка команд, то есть только в начале telnet сессии.

Ключ	Описание	Комментарии
<code>telnet.run[<уникальное короткое описание>,<ip>,<порт>,<кодировка>]</code>	Выполнение команды на удаленном устройстве с использованием telnet соединения	



Если telnet проверка возвращает значение с не-ASCII символами и в не-UTF8 кодировке, то необходимо указать корректный параметр ключа `<кодировка>`. Смотрите страницу [о кодировке возвращаемых значений](#) для получения более подробных сведений.

7.2.2.11 Внешние проверки

Обзор

Внешняя проверка исполняется Zabbix сервером [выполнением shell скрипта](#) или бинарного файла. Однако, когда узлы сети наблюдаются через Zabbix прокси, внешние проверки выполняются через этот прокси.

Внешние проверки не требуют на наблюдаемом узле сети какого-либо агента.

Синтаксис ключа элемента данных:

скрипт[<параметр1>, <параметр2>, . . .]

Где:

АРГУМЕНТ

ОПРЕДЕЛЕНИЕ

скрипт Имя shell скрипта или бинарного файла.

параметр(ы) Опциональные параметры командной строки.

Если вы не хотите передавать какие-нибудь параметры скрипту, вы можете использовать:

script[] или
script

Zabbix сервер заглянет в папку указанную как размещение внешних скриптов (параметр 'ExternalScripts' в [файле конфигурации Zabbix сервера](#)) и выполнит заданную команду. Команда будет выполнена от имени пользователя под которым запущен Zabbix сервер, так что любые права доступа или переменные среды должны быть обработаны в оболочке скрипта, если необходимо, и права доступа на команду должны разрешать этому пользователю выполнение скрипта. Для выполнения доступны только те команды, которые имеются в наличии в указанной папке.



Не злоупотребляйте внешними проверками! Каждый скрипт требует запуска процесса форка Zabbix сервером, запуск множества скриптов может значительно снизить производительность Zabbix.

Пример использования

Выполнение скрипта **check_oracle.sh** с первым параметром '-h'. Второй параметр будет заменен IP адресом или DNS именем узла сети в зависимости от выбранного в настройках узла сети.

`check_oracle.sh[-h", "{HOST.CONN}"]`

Предположим, что узел сети настроен на использование IP адреса, тогда Zabbix выполнит:

`check_oracle.sh '-h' '192.168.1.4'`

Результат внешней проверки

Результирующим значением проверки является стандартный вывод вместе со стандартным выводом ошибок (возвращается полный вывод с обрезанными пробелами в конце начиная с Zabbix 2.0).



Текстовый (символ, журнал или текстовый типы данных) элемент данных не станет неподдерживаемым в случае вывода стандартной ошибки.

В случае, если выполняемый скрипт не найден, Zabbix сервер не имеет необходимых прав на его запуск, элемент данный становится неподдерживаемым и будет возвращено соответствующее сообщение об ошибке. В случае превышения времени ожидания, элемент данных также помечается как неподдерживаемый, соответствующее сообщение об ошибке будет отображено и отдельный процесс этого скрипта будет ликвидирован.

7.2.2.12 Агрегированные проверки

Обзор

В агрегированных проверках Zabbix сервер собирает совокупную информацию из элементов данных, выполняя прямые запросы в базу данных.

Агрегированные проверки не требуют на наблюдаемом узле сети наличия какого-либо агента.

Синтаксис

Синтаксис ключа агрегированного элемента данных:

```
функция_группировки["группа узлов сети", "ключ элемента данных", функция_элемента_данных, период_времени]
```

Поддерживаемые функции группировки (функция_группировки):

Функция группировки	Описание
<i>grpavg</i>	Среднее значение
<i>grpmax</i>	Максимальное значение
<i>grpmin</i>	Минимальное значение
<i>grpsum</i>	Сумма значений

Можно включить несколько групп узлов сети, добавив массив с элементами разделенными запятыми. Если указать родительскую группу узлов сети, будет включена родительская группа и все вложенные группы узлов сети вместе с их элементами данных.

Все элементы данных, на которые имеется ссылка из ключа агрегированного элемента данных, должны существовать и собирать данные. В вычисления добавляются только активированные элементы данных у активированных узлов сети.

Ключ агрегированного элемента данных необходимо обновлять вручную, если будет изменен ключ элемента данных на который агрегированный элемент данных ссылается.

Поддерживаемые функции элементов данных (функция_элемента_данных):

Функция элемента данных	Описание
<i>avg</i>	Среднее значение
<i>count</i>	Количество значений
<i>last</i>	Последнее значение
<i>max</i>	Максимальное значение
<i>min</i>	Минимальное значение
<i>sum</i>	Сумма значений

Параметр **период_времени** задает период времени последних собранных значений. Для удобства вы можете использовать [поддерживаемые символы единиц измерения](#), например '5m' (минут) вместо '300' (секунд) или '1d' (день) вместо '86400' (секунд).

Количество значений (с префиксом #) не поддерживается.

Параметр **период_времени** игнорируется сервером в случае, если третий параметр (функция элемента данных) - *last* и его можно не указывать:

```
groupfunc["Группа узлов сети", "Ключ элемента данных", last]
```

Если результатом агрегации будет число с плавающей точкой, оно будет обрезано до целого числа, в случае если типом информации агрегированного элемента данных указан Числовой (целое положительное).

Агрегированный элемент данных может стать неподдерживаемым, если:

- не найдено ни одного элемента данных, на которые он ссылается (что может произойти, если ключ элемента данных некорректный, элементы данных не существуют или все добавленные группы некорректны)
- отсутствуют данные для вычисления функции

Примеры использования

Примеры ключей агрегированных проверок:

Пример 1

Суммарный размер дисков в группе узлов сети 'MySQL Servers'.

```
grpsum["MySQL Servers", "vfs.fs.size[/,total]", last]
```

Пример 2

Средняя загрузка процессора в группе узлов сети 'MySQL Servers'.

```
grpavg["MySQL Servers", "system.cpu.load[,avg1]", last]
```

Пример 3

5-минутное усреднение количества запросов в секунду для группы узлов сети 'MySQL Servers'.

```
grpavg["MySQL Servers", mysql.qps, avg, 5m]
```

Пример 4

Средняя загрузка CPU на всех узлах сети в нескольких группах узлов сети.

```
grpavg[["Servers A", "Servers B", "Servers C"], system.cpu.load, last]
```

7.2.2.13 Траппер элементы данных

Обзор

Траппер элементы данных принимают входящие данные вместо запроса этих данных.

Это полезно для любых данных, которые вы возможно захотите “запихнуть” в Zabbix.

Для использования элемента данных траппера вы должны:

- иметь в Zabbix настроенный траппер элемент данных
- отправлять данные в Zabbix

Настройка

Настройка элемента данных

Для настройки элемента данных траппер:

- Перейдите в: *Настройка* → *Узлы сети*
- Нажмите на *Элементы данных* в строке узла сети
- Нажмите на *Создать элемент данных*
- Введите в диалоге параметры элемента данных

The screenshot shows the 'Create item' dialog box with the following fields filled in:

* Name	Trapper item
Type	Zabbix trapper
* Key	trap
Type of information	Text
* History storage period	3600
Allowed hosts	(empty)

Все обязательные поля ввода отмечены красной звёздочкой.

Поля, которые требуют указания специфичной информации для траппер элемента данных:

Тип Здесь выберите **Zabbix траппер**.

Ключ Ведите ключ, который будет использован для распознавания элемента данных при отправке данных.

Тип информации Выберите тип информации, который будет соответствовать формату данных,

Разрешенные хосты

которые будут отправлены.

Список разделенных запятыми IP адресов, опционально в CIDR представлении, или имена узлов сети.

Если задано, входящие подключения будут приниматься только с перечисленных здесь узлов сети.

Если включена поддержка IPv6, тогда '127.0.0.1', '::127.0.0.1', '::ffff:127.0.0.1' обрабатываются одинаково и '::/0' будет разрешать любые IPv4 или IPv6 адреса. '0.0.0.0/0' можно использовать, чтобы разрешить любые IPv4 адреса.

Обратите внимание, что "IPv4-совместимые IPv6 адреса" (0000::/96 префикс) поддерживаются, но являются устаревшими согласно [RFC4291](#).

Например: Server=127.0.0.1,192.168.1.0/24,::1,2001:db8::/32,zabbix.domain

Пробелы и [пользовательские макросы](#) разрешены в этом поле с Zabbix 2.2.0.

Возможно вам придется подождать до 60 секунд после сохранения элемента данных пока сервер подхватит изменения, до того как вы сможете отправлять значения.

Отправка данных

В самом простом случае, вы можете использовать утилиту [zabbix_sender](#) для отправки некоторого 'тестового значения' траппер элементу данных.

```
zabbix_sender -z <server IP address> -p 10051 -s "New host" -k trap -o "test value"
```

Для отправки значения мы используем следующие ключи:

-z - чтобы указать IP адрес Zabbix сервера

-p - чтобы указать номер порта Zabbix сервера (10051 по умолчанию)

-s - чтобы указать узел сети (убедитесь, что используете здесь 'техническое' [имя узла сети](#), вместо 'видимого' имени)

-k - чтобы указать ключ элемента данных, который мы только что добавили

-o - чтобы указать фактически отправляемое значение

Процесс Zabbix траппера не раскрывает макросы, которые используются в ключе элемента данных в попытке проверки наличия соответствующего ключа элемента данных на целевом узле сети.

Отображение

Результат в Мониторинг → Последние данные:

HOST	NAME	LAST CHECK	LAST VALUE	CHANGE
New host	- other - (2 items)			
	Trapper item	2015-08-11 18:50:53	test value	History

Штампы времени

Если значения отправляются с использованием утилиты [zabbix_sender](#) из файла со штампами времени, тогда эти штампы времени будут скорректированы со временем сервера. Например, если штамп времени элемента данных равен "10:30:50", текущее время на машине откуда запускается [zabbix_sender](#) - "10:40:03", и текущее время на Zabbix сервере "10:40:05", тогда значение элемента данных будет сохранено в базу данных со штампом времени равным "10:30:52".

Аналогично, если значение сначала отправляется на Zabbix прокси, который далее отправляет его Zabbix серверу, штамп времени будет скорректирован в соответствии со временем Zabbix прокси, и потом будет скорректирован со временем Zabbix сервера.

7.2.2.14 JMX мониторинг

Обзор

Мониторинг JMX можно использовать для наблюдения за счетчиками JMX в Java приложениях.

В Zabbix 2.0 добавлена встроенная поддержка мониторинга JMX, был выпущен новый Zabbix демон, так называемый “Zabbix Java gateway”.

Когда Zabbix сервер хочет узнать значение конкретного счетчика JMX у узла сети, он опрашивает Zabbix **Java gateway**, который в свою очередь используя [API управление JMX](#), удаленно опрашивает интересующее приложение.

Для получения более подробных сведений, включая где можно взять Zabbix Java gateway и как его настроить,смотрите [этот раздел](#) руководства.

Связь между Java gateway и наблюдаемым JMX приложением не должна быть закрыта брандмауэром.

Включение удаленного JMX мониторинга для Java приложений

Приложению Java не требуется какое-либо дополнительно установленное программное обеспечение, но для поддержки удаленного мониторинга JMX приложение должно быть запущено с указанными ниже параметрами командной строки.

Как минимум, если вы просто хотите начать наблюдение за простым приложением Java на локальном хосте без каких либо защиты, запустите его со следующими опциями:

```
java \
-Dcom.sun.management.jmxremote \
-Dcom.sun.management.jmxremote.port=12345 \
-Dcom.sun.management.jmxremote.authenticate=false \
-Dcom.sun.management.jmxremote.ssl=false \
-jar /usr/share/doc/openjdk-6-jre-headless/demo/jfc/Notepad/Notepad.jar
```

С этими аргументами Java будет слушать входящие соединения JMX на порту 12345, только с локальных хостов, без обязательных аутентификации или SSL.

Если вы хотите разрешить подключения с другого интерфейса, укажите параметр -Djava.rmi.server.hostname равным IP адресу этого интерфейса.

Если вы хотите иметь более строгую проверку в плане безопасности, есть много других опций в Java, которые вам доступны. Например, следующая иллюстрация запускает приложение с более универсальным набором опций и открывает это приложение для более широкой сети, не только для локального компьютера.

```
java \
-Djava.rmi.server.hostname=192.168.3.14 \
-Dcom.sun.management.jmxremote \
-Dcom.sun.management.jmxremote.port=12345 \
-Dcom.sun.management.jmxremote.authenticate=true \
-Dcom.sun.management.jmxremote.password.file=/etc/java-6-openjdk/management/
jmxremote.password \
-Dcom.sun.management.jmxremote.access.file=/etc/java-6-openjdk/management/
jmxremote.access \
-Dcom.sun.management.jmxremote.ssl=true \
-Djavax.net.ssl.keyStore=$ВАШ_ХРАНИЛИЩЕ_КЛЮЧЕЙ \
-Djavax.net.ssl.keyStorePassword=$ВАШ_ПАРОЛЬ_К_ХРАНИЛИЩУ_КЛЮЧЕЙ \
-Djavax.net.ssl.trustStore=$ВАШ_ДОВЕРЕННОЕ_ХРАНИЛИЩЕ \
-Djavax.net.ssl.trustStorePassword=$ВАШ_ПАРОЛЬ_К_ДОВЕРЕННОМУ_ХРАНИЛИЩУ \
-Dcom.sun.management.jmxremote.ssl.need.client.auth=true \
-jar /usr/share/doc/openjdk-6-jre-headless/demo/jfc/Notepad/Notepad.jar
```

Значительное количество (если не все) этих настроек можно указать в /etc/java-6-openjdk/management/management.properties (или там, где этот файл расположен на вашем компьютере).

Обратите внимание, если вы желаете использовать SSL, то вы должны изменить startup.sh скрипт Java gateway, добавив в него опции `-Djavax.net.ssl.*` так, чтобы он знал где искать хранилище ключей и доверенное хранилище.

Смотрите [Мониторинг и Управление с использованием JMX \[en\]](#) для получения более подробной информации.

Настройка JMX интерфейсов и элементов данных в веб-интерфейсе Zabbix

Когда Java Gateway запущен, сервер знает где его искать и Java приложение запущено с поддержкой 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.

The screenshot shows the 'Hosts' configuration page in Zabbix. A new host is being created with the name 'New host Java'. In the 'Groups' section, 'In groups' is selected, and 'Zabbix servers' is chosen. In the 'Other groups' list, several monitoring groups are listed: Anna group, Annas group, bypass, calendarian, data poolers, Discovered hosts, group 1, group 2, Hypervisors, and Linux servers. A 'New group' input field contains 'Java'. The 'Agent interfaces' section shows one interface with IP address 127.0.0.1, DNS name empty, and port 10050. The 'SNMP interfaces' and 'JMX interfaces' sections each show one interface with IP 127.0.0.1, DNS empty, and port 12345. A note at the bottom states: '* At least one interface must exist.'

Добавление элемента данных JMX агента

Для каждого интересующего вас счетчика JMX вам необходимо добавить элемент данных с типом **JMX агент** присоединенный к этому интерфейсу.

Ключ на снимке экрана ниже имеет следующий вид

`jmx["java.lang:type=Memory", "HeapMemoryUsage.used"]`.

Item Preprocessing

* Name	Used heap memory																
Type	JMX agent																
* Key	jmx["java.lang:type=Memory","HeapMemoryUsage.used"] <input type="button" value="Select"/>																
* Host interface	127.0.0.1 : 12345																
* JMX endpoint	service:jmx:rmi:///jndi/rmi://[HOST.CONN]:[HOST.PORT]/jmxrmi																
User name	{JMX_USERNAME}																
Password	{JMX_USERNAME}																
Type of information	Numeric (unsigned)																
Units																	
* Update interval	30s																
Custom intervals	<table border="1"> <thead> <tr> <th>Type</th> <th>Interval</th> <th>Period</th> <th>Action</th> </tr> </thead> <tbody> <tr> <td>Flexible</td> <td>Scheduling</td> <td>50s</td> <td>1-7.00:00-24:00</td> </tr> <tr> <td colspan="4">Remove</td> </tr> <tr> <td colspan="4">Add</td> </tr> </tbody> </table>	Type	Interval	Period	Action	Flexible	Scheduling	50s	1-7.00:00-24:00	Remove				Add			
Type	Interval	Period	Action														
Flexible	Scheduling	50s	1-7.00:00-24:00														
Remove																	
Add																	
* History storage period	90d																
* Trend storage period	365d																
Show value	As is <input type="button" value="show value mappings"/>																
New application																	
Applications	-None-																
Populates host inventory field	-None-																
Description																	
Enabled	<input checked="" type="checkbox"/>																
	<input type="button" value="Add"/> <input type="button" value="Cancel"/>																

Все обязательные поля ввода отмечены красной звёздочкой.

Поля, требующие специфичной информации для JMX элементов данных:

Тип Укажите здесь **JMX агент**.

Ключ элемента данных `jmx[]` состоит из двух параметров:

имя объекта - имя объекта MBean;

имя атрибута - имя атрибута MBean с опциональными составными именами полей, разделенных точками.

Ключ

Смотрите ниже для получения более подробных сведений о ключах элементов данных JMX.

Начиная с Zabbix 3.4, вы можете обнаруживать MBeans и MBean атрибуты, используя элемент данных `jmx.discovery[]` [низкоуровневого обнаружения](#).

JMX endpoint

Вы можете указать пользовательский JMX endpoint. Убедитесь, что параметры подключения JMX endpoint совпадают с JMX интерфейсом. Это можно сделать при помощи макросов `{HOST.*}`, как это сделано в JMX endpoint по умолчанию.

Это поле поддерживается начиная с 3.4.0. Поддерживаются [макросы](#) `{HOST.*}` и пользовательские макросы.

Имя пользователя	Укажите имя пользователя, если вы настроили аутентификацию у вашего Java приложения. Поддерживаются пользовательские макросы.
Пароль	Укажите пароль, если вы настроили аутентификацию у вашего Java приложения. Поддерживаются пользовательские макросы.
Если вы хотите наблюдать за Логическим счетчиком, который может быть “true” или “false”, вы должны указать тип информации “Числовой (целое положительное)” и “Логический” тип данных. Сервер будет записывать Логические значения как 1 или 0, соответственно.	

Детальная информация о ключах JMX элементов данных

Простые атрибуты

Имя объекта MBean неважно, кроме строки, которую вы определили в вашем Java приложении. Имя атрибута, с другой стороны, может быть более сложным. В случае, если атрибут возвращает простой тип данных (число, строку и т.п.), то не стоит волноваться об этом, ключ будет выглядеть примерно так:

```
jmx[com.example>Type=Hello,weight]
```

В этом примере именем объекта является “com.example>Type=Hello”, именем атрибута будет являться “weight” и, скорее всего, тип возвращаемого значения должен быть “Числовой (с плавающей точкой)”.

Атрибуты возвращающие составные данные

Ключ становится более сложным, когда ваш атрибут возвращает составные данные. Например: именем вашего атрибута является “apple” и он возвращает хэш 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 хэша. Точно также, если атрибут возвращает часть вложенных составных данных, их нужно снова разделить точкой:

```
jmx[com.example>Type=Hello,fruits.apple.weight]
```

Проблема с точками

Пока все 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 сказать Zabbix'у, что имя атрибута “all.fruits”, а не просто “all”? Как отличить точку, которая является частью имени, от точки которая разделяет имя атрибута и ключи хэшей?

До 2.0.4 Zabbix Java gateway был не способен справится с такими ситуациями и пользователи оставались с НЕПОДДЕРЖИВАЕМЫМИ элементами данных. Начиная с 2.0.4 проблема была исправлена, все что вам требуется сделать - экранировать точки, которые являются частью имени, обратной косой чертой:

```
jmx[com.example>Type=Hello,all\fruits.apple.weight]
```

Аналогично, если ваш ключ хэша содержит точку вам необходимо её экранировать:

```
jmx[com.example>Type=Hello,all\froits.apple.total\weight]
```

Другие проблемы

Символ обратной косой черты тоже должен быть экранирован:

```
jmx[com.example:type=Hello,c:\\\documents]
```

Для обработки любых других символов в ключе JMX элемента данных, пожалуйста, смотрите [раздел](#) формата ключа элементов данных.

На самом деле это все, что нужно сделать. Успешного мониторинга JMX!

Непримитивные типы данных

Начиная с Zabbix 4.0.0 имеется возможность работы с пользовательскими MBean, которые возвращают непримитивные типы данных, которые переопределяют метод **toString()**.

Пример пользовательского endpoint с JBoss EAP 6.4

Пользовательские endpoint позволяют работать с различными транспортными протоколами, которые отличаются от протокола по умолчанию RMI.

Для иллюстрации этой возможности в качестве примера давайте попытаемся настроить JBoss EAP 6.4. Во-первых, давайте сделаем некоторые предположения:

- У вас уже имеется установленный Zabbix Java gateway. Если нет, тогда вам нужно 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 запущен в автономном режиме.
- Мы будем считать, что все эти компоненты работают на одном и том же хосте.
- Брандмауэр и SELinux отключены (или настроены соответствующим образом).

Давайте выполним некоторые простые настройки в zabbix_server.conf:

```
JavaGateway=127.0.0.1
StartJavaPollers=5
```

И в файле конфигурации zabbix_java/settings.sh (или zabbix_java_gateway.conf):

```
START_POLLERS=5
```

Проверьте, что JBoss слушает свой стандартный порт управления:

```
$ netstat -natp | grep 9999
tcp        0      0 127.0.0.1:9999          0.0.0.0:*                  LISTEN
10148/java
```

Теперь давайте создадим в Zabbix узел сети с JMX интерфейсом 127.0.0.1:9999.

Hosts

All hosts / jboss Enabled ZBX SNMP JMX IPMI Applications 8 Items 55 Triggers 26 Graphs 11 Discovery rules Web scenarios

Host Templates IPMI Macros Host inventory Encryption

Host name: jboss
Visible name:

Groups: In groups: Linux servers; Other groups: Discovered hosts, Hypervisors, Templates, Templates/Applications, Templates/Databases, Templates/Modules, Templates/Network Devices, Templates/Operating Systems, Templates/Servers Hardware

New group:

Agent interfaces: IP address: 127.0.0.1, DNS name:, Connect to: IP, Port: 10050, Default, Remove

SNMP interfaces: Add

JMX interfaces: IP address: 127.0.0.1, DNS name:, Connect to: IP, Port: 9999, Default, Remove (highlighted with a red box and arrow)

Как мы знаем эта версия JBoss использует протокол JBoss Remoting вместо RMI, мы можем использовать массовое обновление параметра JMX endpoint в нашем шаблоне JMX в соответствии:
service:jmx:remoting-jmx://{{HOST.CONN}}:{HOST.PORT}

Items

All templates / Template App Generic Java JMX-remoting Applications 8 Items 55 Triggers 26

Type: Original

JMX endpoint: service:jmx:remoting-jmx://{{HOST.CONN}}:{HOST.PORT}

Давайте обновим кэш конфигурации:

```
$ /usr/local/sbin/zabbix_server -R config_cache_reload
```

Обратите внимание, что сначала может возникнуть ошибка.

```

3. mc [root@centos7-dev]:/home/vagrant/zabbix-3.2.6/src/zabbix_java (ssh)
com.zabbix.gateway.ZabbixException: java.net.MalformedURLException: Unsupported protocol: remoting-jmx
    at com.zabbix.gateway.JMXItemChecker.getValues(JMXItemChecker.java:97) ~[zabbix-java-gateway-3.4.2.jar:na]
    at com.zabbix.gateway.SocketProcessor.run(SocketProcessor.java:63) ~[zabbix-java-gateway-3.4.2.jar:na]
    at java.util.concurrent.ThreadPoolExecutor.runWorker(ThreadPoolExecutor.java:1149) [na:1.8.0_144]
    at java.util.concurrent.ThreadPoolExecutor$Worker.run(ThreadPoolExecutor.java:624) [na:1.8.0_144]
    at java.lang.Thread.run(Thread.java:748) [na:1.8.0_144]
Caused by: java.net.MalformedURLException: Unsupported protocol: remoting-jmx
    at javax.management.remote.JMXConnectorFactory.newJMXConnector(JMXConnectorFactory.java:359) ~[na:1.8.0_144]
    at javax.management.remote.JMXConnectorFactory.connect(JMXConnectorFactory.java:269) ~[na:1.8.0_144]
    at com.zabbix.gateway.ZabbixJMXConnectorFactory$1.run(ZabbixJMXConnectorFactory.java:76) ~[zabbix-java-gateway-3.4.2.jar:na]
    at java.util.concurrent.Executors$RunnableAdapter.call(Executors.java:511) ~[na:1.8.0_144]
    at java.util.concurrent.FutureTask.run(FutureTask.java:266) ~[na:1.8.0_144]
    ... 3 common frames omitted
2017-11-07 13:52:12.644 [pool-1-thread-1] WARN com.zabbix.gateway.SocketProcessor - error processing request
com.zabbix.gateway.ZabbixException: java.net.MalformedURLException: Unsupported protocol: remoting-jmx
    at com.zabbix.gateway.JMXItemChecker.getValues(JMXItemChecker.java:97) ~[zabbix-java-gateway-3.4.2.jar:na]
    at com.zabbix.gateway.SocketProcessor.run(SocketProcessor.java:63) ~[zabbix-java-gateway-3.4.2.jar:na]
    at java.util.concurrent.ThreadPoolExecutor.runWorker(ThreadPoolExecutor.java:1149) [na:1.8.0_144]
    at java.util.concurrent.ThreadPoolExecutor$Worker.run(ThreadPoolExecutor.java:624) [na:1.8.0_144]
    at java.lang.Thread.run(Thread.java:748) [na:1.8.0_144]
Caused by: java.net.MalformedURLException: Unsupported protocol: remoting-jmx
    at javax.management.remote.JMXConnectorFactory.newJMXConnector(JMXConnectorFactory.java:359) ~[na:1.8.0_144]
    at javax.management.remote.JMXConnectorFactory.connect(JMXConnectorFactory.java:269) ~[na:1.8.0_144]
    at com.zabbix.gateway.ZabbixJMXConnectorFactory$1.run(ZabbixJMXConnectorFactory.java:76) ~[zabbix-java-gateway-3.4.2.jar:na]
    at java.util.concurrent.Executors$RunnableAdapter.call(Executors.java:511) ~[na:1.8.0_144]
    at java.util.concurrent.FutureTask.run(FutureTask.java:266) ~[na:1.8.0_144]
    ... 3 common frames omitted
2017-11-07 13:52:14.889 [Thread-0] INFO com.zabbix.gateway.JavaGateway - Zabbix Java Gateway 3.4.2 (revision 72885) has stopped
2017-11-07 13:52:26.167 [main] INFO com.zabbix.gateway.JavaGateway - Zabbix Java Gateway 3.4.2 (revision 72885) has started

```

“Unsupported protocol: remoting-jmx” означает, что Java gateway не знает как работать с указанным протоколом. Эту ошибку 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</pre>

```

и затем выполнив эту команду:

```
$ for i in $(cat ~/needed_modules.txt); do find /opt/jboss-eap-6.4 -iname ${i}*.jar -exec cp {} /usr/local/sbin/zabbix_java/lib/ \; ; done
```

Таким образом, у Java gateway будут в наличии все необходимые модули для работы с jmx-remoting. Осталось только перезапустить Java gateway, немного подождать и, если вы все сделали правильно, вы увидите, что эти данные JMX мониторинга начинают поступать в Zabbix:

	Last check	Last value	Change
Name			
Classes (3 items)			
cl Loaded Class Count	2017-11-07 14:08:13	7868	+2
cl Total Loaded Class Count	2017-11-07 14:08:09	7865	+2
cl Unloaded Class Count	2017-11-07 14:08:13	0	
Compilation (2 items)			
comp Accumulated time spent in compilation	2017-11-07 14:08:13	49s 758ms	+1s 440ms
comp Name of the current JIT compiler	2017-11-07 14:00:39	HotSpot 64-Bit Tiered Compilers	
Garbage Collector (4 items)			
gc Copy accumulated time spent in collection	2017-11-07 14:08:09	0	
gc Copy number of collections per second	2017-11-07 14:08:09	0	
gc MarkSweepCompact accumulated time spent in collection	2017-11-07 14:08:13	372ms	
gc MarkSweepCompact number of collections per second	2017-11-07 14:08:13	0	
Memory (8 items)			
mem Heap Memory committed	2017-11-07 14:08:13	1.23 GB	
mem Heap Memory max	2017-11-07 14:00:39	1.23 GB	
mem Heap Memory used	2017-11-07 14:08:09	271.07 MB	+4.01 MB
mem Non-Heap Memory committed	2017-11-07 14:08:13	66.38 MB	+364 KB
mem Non-Heap Memory used	2017-11-07 14:08:13	59.5 MB	+128.1 KB
mem Object Pending Finalization Count	2017-11-07 14:08:13	0	
Memory Pool (6 items)			
mp Code Cache committed	2017-11-07 14:08:09	12.31 MB	+128 KB
mp Code Cache max	2017-11-07 14:00:40	240 MB	
mp Code Cache used	2017-11-07 14:08:09	12.23 MB	+145.64 KB
mp Tenured Gen committed	2017-11-07 14:08:13	869.38 MB	
mp Tenured Gen max	2017-11-07 14:00:40	869.38 MB	
mp Tenured Gen used	2017-11-07 14:08:09	32.26 MB	

7.2.2.15 ODBC мониторинг

Обзор

ODBC мониторинг соответствует типу элемента данных *Монитор баз данных* в веб-интерфейсе Zabbix.

ODBC - язык программирования на C, промежуточная прослойка API для доступа к системам управления баз данных (DBMS). Концепт ODBC был разработан Microsoft и в дальнейшем портирован на другие платформы.

Zabbix может выполнять запросы к любой базе данных, которая поддерживается ODBC. Чтобы это сделать, Zabbix не подключается напрямую к базам данных, он использует интерфейс ODBC и драйвера установленные в ODBC. Эта функция позволяет мониторить различные базы данных с различными целями с большей эффективностью - например, проверка специфичных запросов к базе данных, статистика использования и прочее. Zabbix поддерживает unixODBC, которая наиболее часто используется в реализациях ODBC API с открытым исходным кодом.

Установка unixODBC

Предлагаемый вариант установки unixODBC состоит из использования репозитариев пакетов по умолчанию в Linux операционной системы. В наиболее популярные дистрибутивы Linux unixODBC включен в репозитарии пакетов по умолчанию. Если он недоступен, вы можете обратиться к домашней странице UnixODBC: <http://www.unixodbc.org/download.html>.

Установка unixODBC на системы на базе RedHat/Fedora с использованием менеджера пакетов yum:

```
shell> yum -y install unixODBC unixODBC-devel
```

Установка unixODBC на системы на базе SUSE с использованием менеджера пакетов zypper:

```
# zypper in unixODBC-devel
```

Пакет unixODBC-devel требуется для компиляции Zabbix с поддержкой unixODBC.

Установка драйверов unixODBC

Драйвер unixODBC базы данных должен быть установлен для базы данных, которая будет наблюдаваться. unixODBC имеет список поддерживаемых баз данных и драйверов:

<http://www.unixodbc.org/drivers.html>. В некоторых дистрибутивах Linux драйвера баз данных

включены в репозитарии пакетов. Драйвера MySQL базы данных на системы на базе RedHat/Fedora можно установить с помощью менеджера пакетов *yum*:

```
shell> yum install mysql-connector-odbc
```

Установка MySQL драйвера на системы на базе SUSE с использованием менеджера пакетов *zypper*:

```
# zypper in MyODBC-unicODBC
```

Настройка unixODBC

Настройка ODBC выполняется редактированием файлов **odbcinst.ini** и **odbc.ini**. Для проверки размещения этих файлов введите:

```
shell> odbcinst -j
```

odbcinst.ini используется для перечисления установленных драйверов баз данных ODBC:

```
[mysql]
Description = ODBC for MySQL
Driver      = /usr/lib/libmyodbc5.so
```

Подробная информация:

Атрибут	Описание
<i>mysql</i>	Имя драйвера базы данных.
<i>Description</i>	Описание драйвера базы данных.
<i>Driver</i>	Размещение библиотеки драйвера базы данных.

odbc.ini используется для определения источников данных:

```
[test]
Description = MySQL test database
Driver      = mysql
Server      = 127.0.0.1
User        = root
Password    =
Port        = 3306
Database   = zabbix
```

Подробная информация:

Атрибут	Описание
<i>test</i>	Имя источника данных (DSN).
<i>Description</i>	Описание источника данных.
<i>Driver</i>	Имя драйвера базы данных - как указано в odbcinst.ini
<i>Server</i>	IP/DNS сервера базы данных.
<i>User</i>	Пользователь базы данных для подключения.
<i>Password</i>	Пароль к базе данных.
<i>Port</i>	Порт подключения к базе данных.
<i>Database</i>	Имя базы данных.

Для проверки работает ли соединение ODBC корректно, подключение к базе данных необходимо протестировать. Для этого можно воспользоваться утилитой **isql** (включена в пакет unixODBC):

```
shell> isql test
+-----+
| Connected!
|
| sql-statement
| help [tablename]
| quit
|
+-----+
```

SQL>

Компиляция Zabbix с поддержкой ODBC

Для включения поддержки ODBC, Zabbix должен быть скомпилирован со следующим флагом:

```
--with-unixodbc[=ARG]    use odbc driver against unixODBC package
```

Смотрите более подробную информацию о установке Zabbix из [исходных кодов](#).

Настройка элемента данных в веб-интерфейсе Zabbix

Настройка [элемента данных](#) для мониторинга базы данных.

The screenshot shows the configuration of a 'Database monitor' data item. The fields are as follows:

- Name: MySQL host count
- Type: Database monitor
- Key: db.odbc.select[mysql-simple-check,test]
- User name: zabbix
- Password: (empty)
- SQL query: select count(*) from hosts
- Type of information: Numeric (unsigned)

Все обязательные поля ввода отмечены красной звёздочкой.

Специально для элементов данных мониторинга баз данных вы должны указать:

Тип	Выберите здесь <i>Монитор баз данных</i> . Введите db.odbc.select[уникальное_описание,имя_источника_данных]
Ключ	Уникальное описание будет служить идентификатором элемента данных в триггерах и тому подобном.
Имя пользователя	Имя источника данных (DSN) должно быть указано как в настройках odbc.ini. Введите имя пользователя для доступа к базе данных (опционально, если пользователь указан в odbc.ini)
Пароль	Введите пароль пользователя для доступа к базе данных (опционально, если пароль указан в odbc.ini)
SQL запрос	Введите необходимый SQL запрос
Тип информации	Очень важно знать какой тип информации будет возвращаться указанным запросом, то есть выберите корректный тип информации здесь. С некорректным типом информации элемент данных станет неподдерживаемым.

Важные замечания

- Zabbix не ограничивает время выполнения запроса. Пользователь вправе выбирать запросы, которые могут быть выполнены в разумное время.
- Значение параметра [Timeout](#) с Zabbix сервера используется как время ожидания подключения ODBC (обратите внимание, в зависимости от драйвера ODBC время ожидания подключения может быть проигнорировано).
- Запрос должен возвращать только одно значение.
- Если запрос возвращает более чем одну колонку, будет прочитана только первая колонка.

- Если запрос возвращает более чем одну строку, будет прочитана только первая строка.
- Команда SQL должна начинаться с `select`.
- Команда SQL не должна включать в себя переводы строк.
- Смотрите также [известные проблемы](#) по ODBC проверкам

Сообщения об ошибках

Сообщения об ошибках ODBC скомпонованы в поля для предоставления подробной информации.

Например:

```
Cannot execute ODBC query: [SQL_ERROR]:[42601][7][ERROR: syntax error at or near ";"];
Error while executing the query]
```

Обратите внимание, что длина сообщения об ошибке ограничена 2048 байтами, поэтому сообщение может быть укорочено. Если есть более одной ODBC диагностической записи, Zabbix пытается их скомпоновать (разделяя символом |) на сколько позволяет максимальная длина сообщения.

7.2.2.15.1 Рекомендуемые настройки UnixODBC для MySQL

Установка

- Red Hat Enterprise Linux/CentOS:

```
# yum install mysql-connector-odbc
```

- Debian/Ubuntu:

Пожалуйста, обратитесь к [документации MySQL](#), чтобы загрузить необходимый драйвер базы данных для соответствующей платформы.

Для получения дополнительной информации, пожалуйста, обратитесь к: [Установка unixODBC](#).

Настройка

Настройка ODBC выполняется изменением `odbcinst.ini` и `odbc.ini` файлов. Эти файлы конфигурации можно найти в `/etc` папке. Файл `odbcinst.ini` может отсутствовать и в этом случае его необходимо создать вручную.

`odbcinst.ini`

```
[mysql]
Description = General ODBC for MySQL
Driver      = /usr/lib64/libmyodbc5.so
Setup       = /usr/lib64/libodbcmyS.so
FileUsage   = 1
```

Пожалуйста, обратите внимание на следующие примеры параметров конфигурации `odbc.ini`.

- Пример подключения по IP адресу:

```
[TEST_MYSQL]
Description = MySQL database 1
Driver      = mysql
Port        = 3306
Server      = 127.0.0.1
```

- Пример подключения по IP адресу и с использованием учетной записи. По умолчанию используется база данных zabbix:

```
[TEST_MYSQL_FILLED_CRED]
Description = MySQL database 2
Driver = mysql
User = root
Port = 3306
Password = zabbix
Database = zabbix
Server = 127.0.0.1
```

- Пример подключения через сокет и с использованием учетной записи. По умолчанию используется база данных zabbix:

```
[TEST_MYSQL_FILLED_CRED_SOCK]
Description = MySQL database 3
Driver = mysql
User = root
Password = zabbix
Socket = /var/run/mysqld/mysqld.sock
Database = zabbix
```

Все остальные возможные опции параметров конфигурации можно найти на странице [официальной документации MySQL \[en\]](#).

7.2.2.15.2 Рекомендуемые настройки UnixODBC для PostgreSQL

Установка

- Red Hat Enterprise Linux/CentOS:

```
# yum install postgresql-odbc
```

- Debian/Ubuntu:

Пожалуйста, обратитесь к [документации PostgreSQL](#), чтобы загрузить необходимый драйвер базы данных для соответствующей платформы.

Для получения дополнительной информации, пожалуйста, обратитесь к: [Установка unixODBC](#).

Настройка

Настройка ODBC выполняется изменением **odbcinst.ini** и **odbc.ini** файлов. Эти файлы конфигурации можно найти в **/etc** папке. Файл **odbcinst.ini** может отсутствовать и в этом случае его необходимо создать вручную.

Пожалуйста, обратите внимание на следующие примеры:

odbcinst.ini

```
[postgresql]
Description = General ODBC for PostgreSQL
Driver      = /usr/lib64/libodbcpsql.so
Setup       = /usr/lib64/libodbcpsqlS.so
FileUsage   = 1
# Since 1.6 if the driver manager was built with thread support you may add another
entry to each driver entry.
# This entry alters the default thread serialization level.
Threading   = 2
```

odbc.ini

```
[TEST_PSQL]
Description = PostgreSQL database 1
Driver = postgresql
#CommLog = /tmp/sql.log
Username = zbx_test
Password = zabbix
# Name of Server. IP or DNS
```

```
Servername = 127.0.0.1
# Database name
Database = zabbix
# Postmaster listening port
Port = 5432
# Database is read only
# Whether the datasource will allow updates.
ReadOnly = No
# PostgreSQL backend protocol
# Note that when using SSL connections this setting is ignored.
# 7.4+: Use the 7.4(V3) protocol. This is only compatible with 7.4 and higher backends.
Protocol = 7.4+
# Includes the OID in SQLColumns
ShowOidColumn = No
# Fakes a unique index on OID
FakeOidIndex = No
# Row Versioning
# Allows applications to detect whether data has been modified by other users
# while you are attempting to update a row.
# It also speeds the update process since every single column does not need to be
specified in the where clause to update a row.
RowVersioning = No
# Show SystemTables
# The driver will treat system tables as regular tables in SQLTables. This is good for
Access so you can see system tables.
ShowSystemTables = No
# If true, the driver automatically uses declare cursor/fetch to handle SELECT
statements and keeps 100 rows in a cache.
Fetch = Yes
# Bools as Char
# Bools are mapped to SQL_CHAR, otherwise to SQL_BIT.
BoolsAsChar = Yes
# SSL mode
SSLmode = Yes
# Send to backend on connection
ConnSettings =
```

7.2.2.15.3 Рекомендуемые настройки UnixODBC для Oracle

Установка

Пожалуйста, для получения всех требуемых инструкций обратитесь к [документации Oracle](#).

Для получения дополнительной информации, пожалуйста, обратитесь к: [Установка unixODBC](#).

7.2.2.15.4 Рекомендуемые настройки UnixODBC для MSSQL

Установка

- **Red Hat Enterprise Linux/CentOS:**

```
# yum -y install freetds unixODBC
```

- **Debian/Ubuntu:**

Пожалуйста, обратитесь к [документации пользователя FreeTDS](#), чтобы загрузить необходимый драйвер базы данных для соответствующей платформы.

Для получения дополнительной информации, пожалуйста, обратитесь к: [Установка unixODBC](#).

Настройка

Настройка ODBC выполняется изменением **odbcinst.ini** и **odbc.ini** файлов. Эти файлы конфигурации можно найти в */etc* папке. Файл **odbcinst.ini** может отсутствовать и в этом случае его необходимо создать вручную.

Пожалуйста, обратите внимание на следующие примеры:

odbcinst.ini

```
$ vi /etc/odbcinst.ini
[FreeTDS]
Driver = /usr/lib64/libtdsodbc.so.0
```

odbc.ini

```
$ vi /etc/odbc.ini
[sql1]
Driver = FreeTDS
Server = <SQL сервер 1 IP>
PORT = 1433
TDS_Version = 8.0
```

7.2.2.16 Зависимые элементы данных

Обзор

Имеются ситуации, когда один элемент данных собирает несколько метрик за раз или имеется больший смысл в одновременном сборе связанных метрик, например:

- Утилизация CPU отдельных ядер
- Входящий/исходящий/весь сетевой трафик

Для обеспечения массового сбора метрик и использования синхронности в нескольких связанных элементах данных, Zabbix поддерживает зависимые элементы данных. Зависимые элементы данных используют основной элемент данных, чтобы собрать свои данные одновременно, одним запросом. Новое значение у основного элемента данных автоматически заполняет значения и зависимых элементов данных.

Опции предобработки в Zabbix можно использовать для извлечения части, которая необходима зависимому элементу данных, из данных основного элемента данных.

Предварительная обработка управляется при помощи менеджера предобработки процесса, который добавлен в Zabbix 3.4, вместе с процессами, которые выполняют шаги предобработки. Все значения (с и без предварительной обработкой) от разных сборщиков данных проходят через менеджер предварительной обработки перед добавлением в кэш истории. Для связи между сборщиками данных (поллерами, трапперами и т.д.) и процессами предобработки используется межпроцессорное взаимодействие (IPC) на основе сокета.

Только Zabbix сервер выполняет шаги предварительной обработки и он же обрабатывает зависимые элементы данных.

Элемент данных любого типа, даже зависимый элемент данных, может быть основным элементом данных. Дополнительные уровни зависимых элементов данных можно использовать для извлечения меньших частей значения уже существующего зависимого элемента данных.

Ограничения

- Зависимости разрешены только в пределах одного узла сети (шаблона)
- Прототип элемента данных может зависеть от другого прототипа элемента данных или от обычного элемента данных с этого же узла сети
- Максимальное количество зависимых элементов данных от одного основного элемента данных ограничено 999 элементами данных
- Разрешено максимум 3 уровней зависимости
- Зависимый элемент данных вместе с основным элементом данных из шаблона не будет экспортироваться в XML

Настройка элемента данных

Зависимый элемент данных зависит от его основного элемента данных. Поэтому сначала необходимо настроить (или использовать существующий) **основной элемент данных**:

- Перейдите в: *Настройка* → *Узлы сети*
- Нажмите на *Элементы данных* в строке с узлом сети
- Нажмите на *Создать элемент данных*
- В диалоге укажите параметры элемента данных

The screenshot shows the 'Preprocessing' tab of a configuration dialog. The fields are as follows:

* Name	Apache server status
Type	Zabbix agent
* Key	web.page.get[127.0.0.1,/server-status]
* Host interface	127.0.0.1 : 10050
Type of information	Text
* Update interval	30s

Все обязательные поля ввода отмечены красной звёздочкой.

Нажмите на *Добавить* для сохранения основного элемента данных.

Теперь вы можете настроить **зависимый элемент данных**.

The screenshot shows the 'Preprocessing' tab of a configuration dialog for a dependent item. The fields are as follows:

* Name	Apache server uptime
Type	Dependent item
* Key	apache.server.uptime
* Master item	Apache server status: web.page.get[127.0.0.1,/server-status]
Type of information	Text

Все обязательные поля ввода отмечены красной звёздочкой.

Следующие поля требуют особые параметры по зависимым элементам данных:

Тип	Выберите здесь Зависимый элемент данных .
Ключ	Укажите ключ, который будет использоваться для распознавания этого элемента данных.
Основной элемент данных	Выберите основной элемент данных. Значение основного элемента данных будет использовано для заполнения значения зависимого элемента данных.
Тип информации	Выберите тип информации, который соответствует формату сохраняемых данных.

Вы можете использовать значение [предобработки](#), чтобы извлечь требуемую часть значения основного элемента данных.

Item Preprocessing

Preprocessing steps	Name	Parameters
	Regular expression	<dt>Server uptime: (.*)</dt>\1
Add		

Без предварительной обработки значением зависимого элемента данных будет тем же значением, что и значение основного элемента данных.

Нажмите на *Добавить*, чтобы сохранить зависимый элемент данных.

В списке элементов данных при помощи быстрого доступа создания зависимого элемента данных можно использовать помощника:

The screenshot shows a list of items under 'Apache server status'. One item is selected, and a context menu is open over it. The menu options are: 'Create trigger', 'Edit trigger', and 'Create dependent item'. The 'Create dependent item' option is highlighted with a mouse cursor.

Отображение

В списке элементов данных зависимые элементы данных отображают с префиксом имени основного элемента данных.

Wizard	Name	Triggers	Key
	Apache server status		web.page.get[192.168.3.31/server-status]
	Apache server status: Apache server uptime		apache.server.uptime

Если основной элемент данных удаляется, тогда удаляются и все его зависимые элементы данных.

7.2.2.17 HTTP агент

Обзор

Этот тип элемента данных позволяет получать данные с использованием HTTP/HTTPS протокола. Трапы также возможны с использованием Zabbix sender или протокола Zabbix sender.

Проверка HTTP элемента данных выполняется на стороне Zabbix сервера. Однако, когда узлы сети наблюдаются через Zabbix прокси, проверки HTTP элементов данных выполняются этими прокси.

Проверки HTTP элементов данных не требуют наличия какого-либо агента, запущенного на наблюдаемом узле сети.

HTTP агент поддерживает как HTTP, так и HTTPS. Zabbix будет опционально следовать перенаправлениям (смотрите ниже опцию *Следовать перенаправлениям*). Максимальное количество перенаправлений вшито в код и равно 10 (используется cURL опция CURLOPT_MAXREDIRS).

Смотрите также [известные проблемы](#) при использовании HTTPS протокола.

Zabbix сервер/прокси должны быть изначально сконфигурированы с поддержкой cURL (libcurl).

Настройка

Для настройки HTTP элемента данных:

- Перейдите в: *Настройка* → *Узлы сети*
- Нажмите на *Элементы данных* в строке с узлом сети
- Нажмите на *Создать элемент данных*
- Ведите параметры элемента данных в диалоге

Item Preprocessing

* Name: HTTP agent item

Type: HTTP agent

* Key: http_value_search

* URL: http://localhost:9200/_search

Query Fields

Name	Value
scroll	10s

Add

Request type: POST

Timeout: 3s

Request body type: Raw data, JSON data, XML data

Request body:

```
{"query": { "bool": { "must": [ { "match": { "itemid": 28275 } } ] } }}
```

Headers

Name	Value
name	value

Add

Required status codes: 200

Follow redirects:

Retrieve mode: Body, Headers, Body and headers

Convert to JSON:

HTTP proxy: http://[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

Type of information: Numeric (unsigned)

Units:

* Update interval: 30s

Custom intervals

Type	Interval	Period	Action
Flexible	50s	1-7.00:00-24:00	Remove

Add

* History storage period: 90d

* Trend storage period: 365d

Show value: As is, show value mappings

Enable trapping:

Allowed hosts: 104.24.103.152

New application:

Все обязательные поля ввода отмечены красной звёздочкой.

Поля, которые требуют специфичную информацию HTTP элементов данных:

Tip

Выберите здесь **HTTP агент**.

Ключ

Введите уникальный ключ элемента данных.

	<p>URL для подключения и получения данных. Например: https://www.google.com http://www.zabbix.com/download</p> <p>Имена доменов можно указывать Юникод символами. Они автоматически конвертируются методом ripuscode в ASCII при выполнении шага веб-сценария.</p> <p>Кнопку Анализ можно использовать чтобы отделить из URL optionalные поля запроса (например, ?name=Admin&password=mypassword), переместив атрибуты и значения в Поля запроса, чтобы URL кодировка выполнялась автоматически.</p> <p>Ограничено 2048 символами.</p> <p>Поддерживаемые макросы: {HOST.IP}, {HOST.CONN}, {HOST.DNS}, {HOST.HOST}, {HOST.NAME}, {ITEM.ID}, {ITEM.KEY}, пользовательские макросы, макросы низкоуровневого обнаружения.</p> <p>Это поле задает CURLOPT_URL [en] опцию cURL.</p> <p>Переменные для URL (см. выше).</p> <p>Задаются в виде пар атрибутов и значений.</p> <p>Значения URL кодируются автоматически. Значения с макросов раскрываются и затем URL кодируются автоматически.</p> <p>Поддерживаемые макросы: {HOST.IP}, {HOST.CONN}, {HOST.DNS}, {HOST.HOST}, {HOST.NAME}, {ITEM.ID}, {ITEM.KEY}, пользовательские макросы, макросы низкоуровневого обнаружения.</p> <p>Это поле задает CURLOPT_URL [en] опцию cURL.</p>
Поля запроса	<p>Переменные для URL (см. выше).</p> <p>Задаются в виде пар атрибутов и значений.</p> <p>Значения URL кодируются автоматически. Значения с макросов раскрываются и затем URL кодируются автоматически.</p> <p>Поддерживаемые макросы: {HOST.IP}, {HOST.CONN}, {HOST.DNS}, {HOST.HOST}, {HOST.NAME}, {ITEM.ID}, {ITEM.KEY}, пользовательские макросы, макросы низкоуровневого обнаружения.</p> <p>Это поле задает CURLOPT_URL [en] опцию cURL.</p>
Тип запроса	<p>Выберите тип метода запроса: <i>GET</i>, <i>POST</i>, <i>PUT</i> или <i>HEAD</i></p> <p>Zabbix не будет тратить более указанного времени при обработке URL (максимум 1 минута). В действительности же этот параметр определяет максимальное время создания подключения к URL и максимальное время для выполнения HTTP запроса. Следовательно, Zabbix не будет тратить более 2 x Время ожидания секунд за одну проверку.</p> <p>Поддерживаются суффиксы времени, например, 30s, 1m.</p> <p>Поддерживаемые макросы: пользовательские макросы, макросы низкоуровневого обнаружения.</p> <p>Это поле задает CURLOPT_TIMEOUT [en] опцию cURL.</p>
Время ожидания	<p>Выберите тип запроса тела:</p> <p>Сырые данные - пользовательское тело HTTP запроса, макросы заменяются, но кодирование не выполняется</p> <p>JSON данные - тело HTTP запроса в формате JSON. Макросы можно использовать в виде строки, числа, true или false; макросы, которые используются в виде строки должны быть заключены в двойные кавычки.</p> <p>Значения из макросов раскрываются и затем экранируются автоматически. Если в заголовках "Content-Type" не задан, тогда будет указано значение по умолчанию "Content-Type: application/json"</p> <p>XML данные - тело HTTP запроса в формате XML. Макросы можно использовать в виде нод текста, атрибутов или CDATA секции. Значения из макросов раскрываются и затем экранируются автоматически в нодах текста и в атрибутах. Если в заголовках "Content-Type" не задан, тогда будет указано значение по умолчанию "Content-Type: application/xml"</p> <p><i>Обратите внимание</i>, что выбор XML данные требует наличия поддержки libxml2.</p> <p>Введите тело запроса.</p> <p>Поддерживаемые макросы: {HOST.IP}, {HOST.CONN}, {HOST.DNS}, {HOST.HOST}, {HOST.NAME}, {ITEM.ID}, {ITEM.KEY}, пользовательские макросы, макросы низкоуровневого обнаружения.</p> <p>Пользовательские HTTP заголовки, которые будут отправлены при выполнении запроса.</p> <p>Задаются в виде пар атрибутов и значений.</p> <p>Поддерживаемые макросы: {HOST.IP}, {HOST.CONN}, {HOST.DNS}, {HOST.HOST}, {HOST.NAME}, {ITEM.ID}, {ITEM.KEY}, пользовательские</p>
Тип запроса тела	<p>Поддерживаемые макросы: {HOST.IP}, {HOST.CONN}, {HOST.DNS}, {HOST.HOST}, {HOST.NAME}, {ITEM.ID}, {ITEM.KEY}, пользовательские макросы, макросы низкоуровневого обнаружения.</p> <p>Пользовательские HTTP заголовки, которые будут отправлены при выполнении запроса.</p> <p>Задаются в виде пар атрибутов и значений.</p> <p>Поддерживаемые макросы: {HOST.IP}, {HOST.CONN}, {HOST.DNS}, {HOST.HOST}, {HOST.NAME}, {ITEM.ID}, {ITEM.KEY}, пользовательские</p>
Запрос тела	<p>Поддерживаемые макросы: {HOST.IP}, {HOST.CONN}, {HOST.DNS}, {HOST.HOST}, {HOST.NAME}, {ITEM.ID}, {ITEM.KEY}, пользовательские макросы, макросы низкоуровневого обнаружения.</p>
Заголовки	<p>Поддерживаемые макросы: {HOST.IP}, {HOST.CONN}, {HOST.DNS}, {HOST.HOST}, {HOST.NAME}, {ITEM.ID}, {ITEM.KEY}, пользовательские макросы, макросы низкоуровневого обнаружения.</p>

Требуемые коды состояния

макросы, макросы низкоуровневого обнаружения.

Это поле задает [CURLOPT_HTTPHEADER](#) [en] опцию cURL.

Список ожидаемых кодов состояния HTTP. Если Zabbix получает код не из списка, то элемент данных станет неподдерживаемым. Если поле не заполнено, то проверка не производится.

Например: 200,201,210-299

Поддерживаемые макросы в этом списке: пользовательские макросы, макросы низкоуровневого обнаружения.

Это поле задает [CURLINFO_RESPONSE_CODE](#) [en] опцию cURL.

Следовать перенаправлениям

Отметьте для следования по HTTP перенаправлениям.

Это поле задает [CURLOPT_FOLLOWLOCATION](#) [en] опцию cURL.

Выберите часть ответа, которую необходимо получать:

Тело - только тело

Заголовки - только заголовки

Тело и заголовки - тело и заголовки

Заголовки сохраняются в виде пар атрибутов и значений под ключем "header". Если встречается 'Content-Type: application/json', тогда тело сохраняется в виде объекта, в противном случае оно сохраняется в виде строки, например:

```
{  
    "header": {  
        "<key>": "<value>",  
        "<key2>": "<value>"  
    },  
    "body": <body>  
}
```

Конвертация в JSON

Вы можете указать необходимый HTTP прокси, следуя следующему формату:
`http://[имя пользователя[:пароль]@]прокси.тускомпани.ком[:порт]`

По умолчанию будет использоваться порт 1080.

Если указан, прокси заменит переменные окружения связанные с прокси такие как http_proxy, HTTPS_PROXY. Если не указан, переменные окружения не будут заменены. Введённое значение передается "как есть", проверка правильности не производится.

HTTP прокси

Вы также можете указать адрес SOCKS прокси. Если вы укажите ошибочный протокол, подключение провалится и элемент данных станет неподдерживаемым. Если протокол не указан, прокси будет считаться HTTP прокси.

Примечание: Для HTTP прокси поддерживается только простая аутентификация.

Поддерживаемые макросы: {HOST.IP}, {HOST.CONN}, {HOST.DNS}, {HOST.HOST}, {HOST.NAME}, {ITEM.ID}, {ITEM.KEY}, пользовательские макросы, макросы низкоуровневого обнаружения.

Это поле задает [CURLOPT_PROXY](#) [en] опцию cURL.

Тип аутентификации:

Пусто - без использования аутентификации.

Простая аутентификация - с использованием простой аутентификации.

NTLM аутентификация - с использованием NTLM ([Windows NT LAN Manager](#)) аутентификации.

HTTP Аутентификация

Выбрав какой-либо метод аутентификации, будут доступны доступны два дополнительных поля для ввода имени пользователя и пароля, в которых поддерживаются пользовательские макросы и макросы низкоуровневого обнаружения.

Это поле задает [CURLOPT_HTTPAUTH](#) [en] опцию cURL.

Отметьте для верификации SSL сертификата веб-сервера. Сертификат сервера будет автоматически взят из места центра сертификации (CA) всей системы. Вы можете перезаписать расположение CA файлов, используя параметр конфигурации SSLCAlocation в Zabbix сервере или прокси.

Это поле задает [CURLOPT_SSL_VERIFYPEER](#) [en] опцию cURL.

Проверка SSL узла

Проверка SSL хоста

Отметьте для верификации, что поле *Common Name* или поле *Subject Alternate Name* сертификата веб-сервера совпадают.
Это поле задает [CURLOPT_SSL_VERIFYHOST](#) [en] опцию cURL.

Файл SSL сертификата

Имя файла SSL сертификата для аутентификации клиента. Файл сертификата должен быть в формате PEM¹. Если файл сертификата также содержит и приватный ключ, оставьте поле Файл SSL ключа пустым. Если ключ зашифрован, укажите пароль в поле Пароль к SSL ключу. Папка, содержащая этот файл указывается в параметре конфигурации SSLCertLocation Zabbix сервера или прокси.

Поддерживаемые макросы: {HOST.IP}, {HOST.CONN}, {HOST.DNS}, {HOST.HOST}, {HOST.NAME}, {ITEM.ID}, {ITEM.KEY}, пользовательские макросы, макросы низкоуровневого обнаружения.

Это поле задает [CURLOPT_SSLCERT](#) [en] опцию cURL.

Файл SSL ключа

Имя файла приватного SSL ключа, который используется для аутентификации клиента. Файл приватного ключа должен быть в формате PEM¹. Папка, содержащая этот файл указывается в параметре конфигурации SSLKeyLocation Zabbix сервера или прокси.

Поддерживаемые макросы: {HOST.IP}, {HOST.CONN}, {HOST.DNS}, {HOST.HOST}, {HOST.NAME}, {ITEM.ID}, {ITEM.KEY}, пользовательские макросы, макросы низкоуровневого обнаружения.

Это поле задает [CURLOPT_SSLKEY](#) [en] опцию cURL.

Пароль к файлу приватного ключа.

Пароль к SSL ключу

Поддерживаемые макросы: пользовательские макросы, макросы низкоуровневого обнаружения.

Это поле задает [CURLOPT_KEYPASSWD](#) [en] опцию cURL.

Активировать трапы

При наличии этой опции, элемент данных будет работать как [траппер элемент данных](#) и будет принимать данные, которые отправляются на этот элемент данных при помощи Zabbix sender или с использованием протокола Zabbix sender.

Видимо только, если выбрана опция *Активировать трапы*.

Список разделенных запятыми IP адресов, дополнительно в CIDR представлении, или имена узлов сети.

Если задано, входящие подключения будут приниматься только с перечисленных здесь узлов сети.

Если включена поддержка IPv6, тогда '127.0.0.1', '::127.0.0.1', '::ffff:127.0.0.1' обрабатываются одинаково и '::/0' будет разрешать любые IPv4 или IPv6 адреса. '0.0.0.0/0' можно использовать, чтобы разрешить любые IPv4 адреса.

Обратите внимание, что “IPv4-совместимые IPv6 адреса” (0000::/96 префикс) поддерживаются, но являются устаревшими согласно [RFC4291](#).

Пример: Server=127.0.0.1,192.168.1.0/24,::1,2001:db8::/32,zabbix.domain
В этом поле разрешены пробелы и [пользовательские макросы](#).

Если поле *HTTP прокси* оставить пустым, можно воспользоваться другим способом указать HTTP прокси, для этого необходимо задать переменные окружения.

Для HTTP проверок - укажите переменную окружения **http_proxy** для пользователя Zabbix сервера. Например, `http_proxy=http://proxy_ip:proxy_port`.

Для HTTPS проверок - укажите переменную окружения **HTTPS_PROXY**. Например, `HTTPS_PROXY=http://proxy_ip:proxy_port`. Более подробную информацию можно получить, выполнив в shell команду `# man curl`.

[1] Zabbix поддерживает файлы сертификатов и приватных ключей только в PEM формате. В случае, если у вас имеются данные сертификата и приватного ключа в формате файла PKCS #12 (обычно используется с расширением *.p12 или *.pfx), вы можете сгенерировать из них PEM файл, используя следующие команды:

```
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

Отправка простых GET запросов на получение данных с 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, используя шаг предварительной обработки JSONPath:
`$.version.number`

Пример 2

Отправка простых POST запросов на получение данных с таких сервисов, как Elasticsearch:

- Создайте POST элемент данных с URL:
http://localhost:9200/_search?scroll=10s
- Настройте следующее тело POST, чтобы получать загрузку процессора (среднее значение за 1 по каждому ядру)

```
{
  "query": {
    "bool": {
      "must": [
        {
          "match": {
            "itemid": 28275
          }
        }
      ],
      "filter": [
        {
          "range": {
            "clock": {
              "gt": 1517565836,
              "lte": 1517566137
            }
          }
        }
      ]
    }
  }
}
```

- Получено:

```
{
  "_scroll_id": "DnF1ZXJ5VGhbkZldGNoBQAAAAAAAAAAkFllRMLZBWS1UU1pxTmdEeGVwQjRBTFEAAAAAAAJRZZUTJWQVktVF
NacU5nRHhlcEI0QUxRAAAAAAAAACYWWVEyVkfZLVRTWnF0Z0R4ZXBCNEFMUQAAAAAAAAnFllRMLZBWS1UU1pxT
mdEeGVwQjRBTFEAAAAAAAABZZUTJWQVktVFNacU5nRHhlcEI0QUxR",
  "took": 18,
```

```

"timed_out": false,
"_shards": {
    "total": 5,
    "successful": 5,
    "skipped": 0,
    "failed": 0
},
"hits": {
    "total": 1,
    "max_score": 1.0,
    "hits": [
        {
            "_index": "dbl",
            "_type": "values",
            "_id": "dqX9VWEBV6sEKSMyk6sw",
            "_score": 1.0,
            "_source": {
                "itemid": 28275,
                "value": "0.138750",
                "clock": 1517566136,
                "ns": 25388713,
                "ttl": 604800
            }
        }
    ]
}
}

```

- Теперь используйте шаг предварительной обработки JSONPath, чтобы получить значение элемента данных: `$.hits.hits[0]._source.value`

Пример 3

Проверка, доступен ли Zabbix API с использованием [apiinfo.version](#).

- Настройка элемента данных:

Item Preprocessing

* Name	Check Zabbix API version						
Type	HTTP agent ▾						
* Key	check_zabbix_api_info.version						
* URL	http://zabbix-web-apache-mysql/api_jsonrpc.php						
Query fields	<table border="1"> <thead> <tr> <th>Name</th> <th>Value</th> </tr> </thead> <tbody> <tr> <td>Add</td> <td></td> </tr> </tbody> </table>	Name	Value	Add			
Name	Value						
Add							
Request type	POST ▾						
Timeout	3s						
Request body type	Raw data JSON data XML data						
Request body	<pre>{ "jsonrpc": "2.0", "method": "apiinfo.version", "params": [], "id": 1 }</pre>						
Headers	<table border="1"> <thead> <tr> <th>Name</th> <th>Value</th> </tr> </thead> <tbody> <tr> <td>Content-Type</td> <td>application/json-rpc</td> </tr> <tr> <td>Add</td> <td></td> </tr> </tbody> </table>	Name	Value	Content-Type	application/json-rpc	Add	
Name	Value						
Content-Type	application/json-rpc						
Add							
Required status codes							
Follow redirects	<input type="checkbox"/>						
Retrieve mode	Body Headers Body and headers						

Обратите внимание, на использование POST метода с JSON данными, с настройкой заголовков в запросе и получением только заголовков:

- Предварительная обработка значений элемента данных при помощи регулярного выражения, чтобы получить HTTP код:

Item Preprocessing

Preprocessing steps	Name	Parameters
	Regular expression ▾	HTTPV1.1 ([0-9]+) \1
Add		

- Проверка результата в *Последние данные*:

Latest data

The screenshot shows the 'Latest data' search interface in Zabbix. The search criteria are set to 'Host groups' (empty), 'Hosts' (Zabbix server, nginx), and 'Name' (Check Zabbix API). The results table shows one item: 'Check Zabbix API version' from 'Zabbix server' last checked at 2018-05-16 23:50:34 with status OK (200). There is a 'Graph' link next to the item.

Пример 4

Получение информации о погоде при помощи подключения к публичному сервису Openweathermap.

- Настройка основного элемента данных для массового сбора данных в одном JSON:

The screenshot shows the 'Item' configuration screen in Zabbix. The item is named 'Get weather' and is of type 'HTTP agent'. It uses the 'Template Weather' key. The URL is set to <http://api.openweathermap.org/data/2.5/weather>. The 'Query fields' section contains five entries: 'units' (Value: metric), 'lat' (Value: {\$LAT}), 'lon' (Value: {\$LON}), 'APPID' (Value: {\$WEATHER_APIKEY}), and 'lang' (Value: {\$WEATHER_LANG}). The 'Request type' is set to 'GET', 'Timeout' is '3s', and 'Request body type' is 'Raw data'. The 'Request body' field is empty.

Обратите внимание на использование макросов в полях запроса. Обратитесь к [Openweathermap API](#) для получения подробных сведений о том как заполнять их.

Пример JSON, который вернулся в ответе 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 - настройка зависимых элементов данных, которые извлекут данные с JSON.

- Настройка примера зависимого элемента данных для влажности:

Item	Preprocessing
* Name	Humidity
Type	Dependent item
* Key	humidity
* Master item	Template Weather: Get weather
Type of information	Numeric (float)

Другие метрики погоды, такие как 'Температура', добавляются тем же способом.

- Пример предобработки значений зависимого элемента данных с использованием JSONPath:

Item	Preprocessing		
Preprocessing steps	Name	Parameters	Action
	JSON Path	\$.body.main.humidity	Remove
	Add		

- Проверка результата 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 в странице состояния Nginx и получения его метрик за один запрос.

- Настройте Nginx, следуя [официальному руководству](#).
- Настройте основной элемент данных для массового сбора данных:

Item Preprocessing

* Name	Get NGINX status page				
Type	HTTP agent				
* Key	get_nginx				
* URL	http://(HOST.CONN)/nginx_status				
Query fields	<table border="1"> <tr> <th>Name</th> <th>Value</th> </tr> <tr> <td>name</td> <td>value</td> </tr> </table> Add	Name	Value	name	value
Name	Value				
name	value				
Request type	GET				
Timeout	3s				
Request body type	Raw data				
Request body					

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 необходимые данные.

- Настройка примера зависимого элемента данных для запросов в секунду:

Item Preprocessing

* Name	Client requests per second
Type	Dependent item ▾
* Key	nginx_requests_rps
* Master item	Template App Nginx HTTP: Get Nginx stub status X
Type of information	Numeric (unsigned) ▾

- Настройка предобработки значений зависимого элемента данных при помощи регулярного выражения:

Item Preprocessing

Preprocessing steps	Name	Parameters
Regular expression	server accepts handled requests s+([0-9]+) ([0-9]+) ([0-9]+)	13
Change per second		

[Add](#)

- Проверьте полученный результат от модуля stub в *Последние данные*:

Host	Name	Last check	Last value
nginx	Nginx (8 items)		
	Accepted client connections	2018-05-18 17:54:53	568
	Active connections	2018-05-18 17:54:53	1
	Client requests per second	2018-05-18 17:54:53	0 rps
	Get Nginx stub status	2018-05-18 17:54:53	HTTP/1.1 200 OK Se...
	Handled connections per second	2018-05-18 17:54:53	0
	Reading	2018-05-18 17:54:53	0
	Waiting	2018-05-18 17:54:53	0
	Writing	2018-05-18 17:54:53	1

7.2.3 История и динамика изменений

Обзор

История и динамика изменения - два пути хранения в Zabbix собранных данных.

В то время как история хранит каждое собранное значение, динамика изменений хранит усредненную информацию за каждый час, что, следовательно, менее ресурсоемко.

Хранение истории

Вы можете указать как много дней история будет храниться:

- в [диалоге](#) свойств элемента данных
- при массовом обновлении элементов данных
- при [настройке](#) задач очистки истории

Любые более старые данные будут удалены с помощью автоматической очистки базы данных (Housekeeper).

Самый большой совет - хранить историю как можно меньшее количество дней и таким образом не перегружать базу данных большим количеством значений истории.

Вместо долговременного хранения истории, вы можете хранить больше данных динамики изменений. Например, вы можете хранить историю 14 дней и динамику изменений 5 лет.

Вы можете прикинуть сколько потребуется места для истории в сравнении с динамикой изменений, используя [страницу оценки размера базы данных](#).

При меньшем хранении истории, вы всё равно сможете просматривать старые данные на графиках, графики будут использовать значения динамики изменений для отображения старых данных.

Если история установлена в '0', элемент данных будет обновлять только поля инвентаря. Функции триггера не будут вычисляться.

В качестве альтернативного подхода к хранению истории обратитесь к функционалу [экспорта истории](#) в подгружаемых модулях.

Хранение динамики изменений

Динамика изменений является встроенным механизмом обработки данных истории, где для каждого часа хранятся значения минимума, максимума и усредненное значения, а также общее количество значений за этот час.

Вы можете указать как много дней динамика изменений будет храниться:

- в [диалоге](#) свойств элемента данных
- при массовом обновлении элементов данных
- при настройке задач очистки истории

Обычно динамика изменений может храниться намного дольше чем история. Любые более старые данные будут удалены с помощью автоматической очистки базы данных (Housekeeper).

Если динамика изменений установлена в '0', Zabbix сервер вообще не будет считать и хранить динамику изменений.

Динамика изменений вычисляется и сохраняется с использованием того же типа данных, как и оригинальные значения. В результате среднее значение вычисления значений целочисленных типов данных округляется и чем меньше интервал значений, тем меньше будет точность результата.

Например, если элемент данных имеет значения 0 и 1, средним значением будет 0, а не 0.5.

Перезапуск сервера также может привести к потере точности вычисления усредненных значений у целочисленных типов данных за текущий час.

7.2.4 Пользовательские параметры

Обзор

Иногда вы можете захотеть выполнять проверку через агент, которая не предопределена в Zabbix. В этом случае вам на помощь придут пользовательские параметры.

Вы можете написать команду, которая вернет необходимые вам данные и добавить эту команду как пользовательский параметр в [файл конфигурации агента](#) (параметр конфигурации 'UserParameter').

Пользовательский параметр имеет следующий синтаксис:

UserParameter=<ключ>, <команда>

Как вы можете видеть, пользовательский параметр также содержит ключ. Ключ потребуется при настройке элемента данных. Введите ключ на ваш выбор, на который будет легко ссылаться (он должен быть уникальным в пределах узла сети). Перезапустите агента.

Затем, когда [настроите элемент данных](#), введите ключ на который ссылается команда из пользовательского параметра, которую вы хотите выполнять.

Команды из пользовательских параметров выполняются Zabbix агентом. Может быть возвращено до 512КБ данных до шагов предварительной обработки элементов данных. Однако, обратите внимание на то, что такое текстовое значение, которое может быть сохранено в базе данных ограничено 64КБ в MySQL (смотрите информацию по остальным базам данных в [таблице](#)).

В операционных системах UNIX используется интерпретатор командной строки **/bin/sh**.

Пользовательские параметры подчиняются времени ожидания проверок на агента; если время ожидания будет превышено, созданный по пользовательским параметрам процесс будет завершен.

Смотрите также:

- [Пошаговую инструкцию](#) 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 ключем '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 допускают параметры с указанным ключом. В этом случае гибкие пользовательские параметры могут быть основой для создания нескольких элементов данных.

Гибкие пользовательские параметры имеют следующий синтаксис:

```
UserParameter=ключ [*], команда
```

Параметр	Описание
Ключ	Уникальный ключ элемента данных. [*] задает, что ключ может принимать параметры из скобок. Параметры указываются при настройке элемента данных. Команда, которая выполняется для получения значения ключа. <i>Только для гибких пользовательских параметров:</i> Вы можете использовать ссылки на позиции \$1...\$9 для того, чтобы обратиться к соответствующему параметру в ключе элемента данных. Zabbix разбирает параметры заключенные в [] ключа элемента данных и заменяет \$1,...,\$9 в команде соответственно.
Команда	\$0 будет заменена оригинальной командой (до раскрытия \$0,...,\$9) для выполнения. Ссылки на позиции интерпретируются Zabbix агентом независимо от того заключены ли они в двойные ("") или в одинарные ('') кавычки. Для использования ссылок на позиции без изменения, укажите двойной символ доллара - например, awk '{print \$\$2}'. В этом случае \$\$2 фактически превратится в \$2 при выполнении команды.

Ссылки на позиции, начинающиеся со знака \$, ищутся и заменяются Zabbix агентом только в случае гибких пользовательских параметров. В случае простых пользовательских параметров такие ссылки пропускаются и, следовательно, экранирование любых знаков \$ не требуется.

По умолчанию некоторые символы не разрешены в пользовательских параметрах. Смотрите документацию по [UnsafeUserParameters](#) для получения полного списка таких символов.

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. В качестве параметров 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

Результирующим значением команды является стандартный вывод вместе со стандартным выводом ошибок.

Текстовый элемент данных (символ, журнал или текст типами информации) не станет неподдерживаемым в случае вывода ошибки стандартным выводом.

Пользовательские параметры, которые возвращают текст (символ, журнал, текстовый типы информации) могут возвращать пробел. В случае ошибочного результата элемент данных станет неподдерживаемым.

7.2.4.1 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 в zabbix_agentd.conf команду:

```
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, однако, стоит отметить, что агент может иметь разные права при работе демоном):

```
zabbix_agentd -t mysql.questions
```

Шаг 3

Перезапустите Zabbix агента.

Агент заново загрузит файл конфигурации.

Протестируйте этот параметр, используя утилиту [zabbix_get](#).

Шаг 4

Добавьте новый элемент данных с Ключем=mysql.questions к наблюдаемому узлу сети. Тип этого элемента данных должен быть Zabbix агент или Zabbix агент (активный).

Помните, очень важно указать корректный тип возвращаемого значения в Zabbix сервере. Иначе Zabbix не будет принимать значения.

7.2.5 Подгружаемые модули

7.2.5.1 Обзор

Подгружаемые модули предлагают производительную опцию для расширения функциональности Zabbix.

Уже имеются возможности для расширения функциональности Zabbix при помощи:

- [пользовательских параметров](#) (метрики агента)
- [внешние проверки](#) (мониторинг без агента)
- `system.run[]` [элемент данных Zabbix агента](#).

Они работают очень хорошо, но имеют главный недостаток, называемый форком (`fork()`). Zabbix должен создавать новый ответвленный процесс каждый раз для сбора пользовательских проверок, что не очень хорошо оказывается на производительности. Обычно это самая большая проблема, но тем не менее это может быть серьезной проблемой для мониторинга встроенных систем, имеющих большое количество наблюдаемых параметров или тяжелых скриптов со сложной логикой или длительным временем запуска.

Поддержка подгружаемых модулей предлагает пути расширения Zabbix агента, сервера и прокси без ущерба производительности.

Подгружаемый модуль - в своей основе разделяемая библиотека используемая Zabbix демоном и загружаемая при старте демона. Библиотека должна содержать определенные функции такие, чтобы Zabbix процесс мог определить что файл на самом деле модуль и он может загрузить его и работать с ним.

Подгружаемые модули имеют много преимуществ. Отличная производительность и возможность внедрения в любую логику что очень важно, но возможно наиболее важное преимущество - возможность развития, использования и распространения Zabbix модулей. Это поспособствует безпроблемному обслуживанию и поможет вносить новую функциональность легче и независимо от кода ядра Zabbix.

Лицензирование и распространение модулей в бинарной форме регламентируется GPL лицензией (модули линкуются с Zabbix во время выполнения и используют заголовки Zabbix; в настоящее время

весь код Zabbix лицензируется под GPL лицензией). Бинарная совместимость не гарантируется Zabbix'ом.

Постоянство API модулей гарантируется в пределах одного цикла Zabbix LTS (Долгосрочная поддержка) [выпуска](#). Постоянство Zabbix API не гарантируется (технически имеется возможность вызова внутренних функций Zabbix из модуля, но гарантии, что такие модули будут работать нет).

7.2.5.2 API модулей

Для того чтобы разделяемая библиотека обрабатывалась как Zabbix модуль, она должна реализовывать и экспортить несколько функций. На данный момент имеется шесть функций в API модулей Zabbix, только одна из которых обязательны, а остальные пять - опциональны.

2.1 Обязательный интерфейс

Единственная обязательная функции - **zbx_module_api_version()**:

```
int      zbx_module_api_version(void);
```

Эта функция должна возвращать API версию реализованную в модуле и, чтобы модуль загрузился, версия должна совпадать с версией API модулей поддерживаемой Zabbix. Версией API модулей поддерживаемой Zabbix является ZBX_MODULE_API_VERSION. Таким образом эта функция должна возвращать эту константу. Старая константа ZBX_MODULE_API_VERSION_ONE, которая ранее использовалась для этих целей, теперь определена равной ZBX_MODULE_API_VERSION для сохранения совместимости исходного кода, но её использование не рекомендуется.

2.2 Опциональный интерфейс

Опциональными функциями являются следующие функции - **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 для модуля (если таковые имеются). В случае успеха, функция должна вернуть ZBX_MODULE_OK. В противном случае, она должна вернуть ZBX_MODULE_FAIL. В последнем случае Zabbix не запустится.

```
ZBX_METRIC      *zbx_module_item_list(void);
```

Эта функция должна возвращать список элементов данных, поддерживаемых модулем. Каждый элемент данных указывается в структуре ZBX_METRIC, смотрите раздел ниже для подробностей. Список завершается при помощи структуры ZBX_METRIC с полем "key" равным NULL.

```
void      zbx_module_item_timeout(int timeout);
```

Если модуль экспортирует **zbx_module_item_list()**, тогда эта функция используется Zabbix, чтобы задать опцию времени ожидания в файле конфигурации Zabbix, которой проверки элементов данных реализованных в модуле должны подчиняться. Здесь, параметр "время ожидания" задается в секундах.

```
ZBX_HISTORY_WRITE_CBS  zbx_module_history_write_cbs(void);
```

Эта функция должна возвращать функции обратного вызова (callback), которые будут использоваться Zabbix сервером для экспорта истории различных типов данных. Функции обратного вызова представляют собой поля ZBX_HISTORY_WRITE_CBS структуры, поля могут быть NULL, если модуль не заинтересован в истории некоторого типа.

```
int      zbx_module_uninit(void);
```

Эта функция должна выполнять необходимые деинициализации (если таковые имеются), такие как освобождение выделенных ресурсов, закрытие файловых дескрипторов и так далее.

Все функции вызываются один раз при запуске Zabbix, когда модуль загружен, за исключением zbx_module_uninit(), которая вызывается один раз при завершении работы Zabbix, когда модуль выгружен.

2.3 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** - ключ элемента данных (например, "dummy.random"), **flags** - либо CF_HAVEPARAMS, либо 0 (в зависимости от того, принимает ли элемент данных параметры или нет), **function** - C функция, которая обрабатывает элемент данных (например, "zbx_module_dummy_random"), и **test_param** - список параметров, которые используется когда Zabbix агент запускается с флагом "-р" (например, "1,1000", может быть NULL). Пример определения может выглядеть наподобие этого:

```
static ZBX_METRIC keys[] =
{
    { "dummy.random", CF_HAVEPARAMS, zbx_module_dummy_random, "1,1000" },
    { NULL }
}
```

Каждая функция, которая обрабатывает элемент данных должна принимать два указателя параметра, первый с типом AGENT_REQUEST и второй с типом 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 получено успешно. В противном случае, функции должны возвращать SYSINFO_RET_FAIL. Смотрите пример "dummy" модуля ниже для получения деталей как получать информацию от AGENT_REQUEST и как указывать информацию в AGENT_RESULT.

2.4 Предоставление обратного вызова экспорту истории

Экспорт истории через модули более не поддерживается Zabbix прокси начиная с Zabbix 4.0.0.

Модуль может задавать функции для экспорта данных истории по типам: Числовой (с плавающей точкой), Числовой (целое положительное), Символ, Текст и Журнал (лог):

```
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;
```

Каждый из них должен принимать массив “history” из элементов “history_num” в виде аргументов. В зависимости от экспортруемого типа данных истории, “history” является массивом из 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;
```

Обратные вызовы будут использоваться Zabbix сервером процессами синхронизации истории в конце процедуры синхронизации истории после того, как данные записаны в базу данных Zabbix и сохранены в кэш значений.

2.5 Сборка модулей

В настоящее время подразумевается, что модули должны быть собраны внутри дерева исходных кодов Zabbix, так как API модулей зависит от некоторых структур данных, которые определены в заголовках Zabbix.

Наиболее важный заголовок для подгружаемых модулей - **include/module.h**, который определяет перечисленные выше структуры данных. Другой полезный заголовок - **include/sysinc.h**, который

выполняет включение необходимых системных заголовков, который сам по себе помогает include/module.h работать должным образом.

Для того чтобы include/module.h и include/sysinc.h были включены, необходимо сначала выполнить команду **./configure** (без аргументов) из корня дерева исходных кодов Zabbix. Команда создаст **include/config.h** файл, который основывается на include/sysinc.h. (Если вы получили исходные коды Zabbix из хранилища Subversion, скрипт ./configure не будет существовать, сначала выполните команду **./bootstrap.sh**, чтобы скрипт сгенерировался.)

Зная эту информацию, имейте в виду, что все готово для сборки модуля. Модуль должен включать **sysinc.h** и **module.h**, и скрипт сборки должен понимать, что эти два файла указаны в **include**. Смотрите пример модуля “dummy” ниже для получения деталей.

Другим полезным заголовком является **include/log.h**, который определяет функцию **zabbix_log()**, которую можно использовать для журналирования и отладки.

7.2.5.3 Параметры конфигурации

Zabbix агент, сервер и прокси поддерживают два [параметра](#) для работы с модулями:

- LoadModulePath – полный путь до расположения подгружаемых модулей
- LoadModule – модуль(и) для загрузки при старте. Модули должны находиться в папке указанной в параметре LoadModulePath. Допускается добавлять несколько параметров LoadModule.

Например, для расширения возможностей Zabbix агента мы можем добавить следующие параметры:

```
LoadModulePath=/usr/local/lib/zabbix/agent/
LoadModule=mariadb.so
LoadModule=apache.so
LoadModule=kernel.so
LoadModule=dummy.so
```

После запуска агента будут загружены модули mariadb.so, apache.so, kernel.so и dummy.so из папки /usr/local/lib/zabbix/agent. Агент не запустится, если модуль отсутствует, либо в случае некорректных прав доступа или разделяемая библиотека не является модулем Zabbix.

7.2.5.4 Настройка веб-интерфейса

Подгружаемые модули поддерживаются Zabbix агентом, сервером и прокси. Следовательно, тип элемента данных в Zabbix веб-интерфейсе зависит от того где этот модуль загружен. Если модуль загружен на стороне агента, тогда тип элемента данных должен быть “Zabbix агент” или “Zabbix агент (активный)”. Если модуль загружен на стороне сервера или прокси, тогда тип элемента данных должен быть “Простая проверка”.

Экспорт истории через модули Zabbix не требует какой-либо настройки через веб-интерфейс. Если модуль успешно загружен сервером и предоставляет функцию **zbx_module_history_write_cbs()**, которая возвращает по крайней мере одну не-NUL функцию обратной связи, тогда экспорт истории будет включен автоматически.

7.2.5.5 Пустой модуль

Zabbix включает простой модуль, написанный на языке C. Модуль находится 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, он можно использовать как шаблон для ваших собственных модулей.

После выполнения ./configure в корне дерева исходных кодов Zabbix, как описано ранее, просто выполните **make** для сборки **dummy.so**.

```
/*
** Zabbix
** Copyright (C) 2001-2016 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 "sysinc.h"
#include "module.h"

/* the variable keeps timeout setting for item processing */
static int      item_timeout = 0;

/* module SHOULD define internal functions as static and use a naming pattern different
from Zabbix internal */
/* symbols (zbx_*) and loadable module API functions (zbx_module_*) to avoid conflicts
*/
static int      dummy_ping(AGENT_REQUEST *request, AGENT_RESULT *result);
static int      dummy_echo(AGENT_REQUEST *request, AGENT_RESULT *result);
static int      dummy_random(AGENT_REQUEST *request, AGENT_RESULT *result);

static ZBX_METRIC keys[] =
/*      KEY                      FLAG          FUNCTION        TEST PARAMETERS */
{
    {"dummy.ping",             0,            dummy_ping,      NULL},
    {"dummy.echo",              CF_HAVEPARAMS, dummy_echo,      "a message"},
    {"dummy.random",            CF_HAVEPARAMS, dummy_random,   "1,1000"},
    {NULL}
};

/*****
*
* Function: zbx_module_api_version
*
* Purpose: returns version number of the module interface
*
* Return value: ZBX_MODULE_API_VERSION - version of module.h module is
*               compiled with, in order to load module successfully Zabbix
*               MUST be compiled with the same version of this header file
*
*****/
int      zbx_module_api_version(void)
{
    return ZBX_MODULE_API_VERSION;
}

/*****
*
* Function: zbx_module_item_timeout
*
* Purpose: set timeout value for processing of items
*
* Parameters: timeout - timeout in seconds, 0 - no timeout set
*
*****/
void      zbx_module_item_timeout(int timeout)
```

```

{
    item_timeout = timeout;
}

/*********************************************
*
* Function: zbx_module_item_list
*
* Purpose: returns list of item keys supported by the module
*
* Return value: list of item keys
*
********************************************/
ZBX_METRIC      *zbx_module_item_list(void)
{
    return keys;
}

static int      dummy_ping(AGENT_REQUEST *request, AGENT_RESULT *result)
{
    SET_UI64_RESULT(result, 1);

    return SYSINFO_RET_OK;
}

static int      dummy_echo(AGENT_REQUEST *request, AGENT_RESULT *result)
{
    char      *param;

    if (1 != request->nparam)
    {
        /* set optional error message */
        SET_MSG_RESULT(result, strdup("Invalid number of parameters."));
        return SYSINFO_RET_FAIL;
    }

    param = get_rparam(request, 0);

    SET_STR_RESULT(result, strdup(param));

    return SYSINFO_RET_OK;
}

/*********************************************
*
* Function: dummy_random
*
* Purpose: a main entry point for processing of an item
*
* Parameters: request - structure that contains item key and parameters
*             request->key - item key without parameters
*             request->nparam - number of parameters
*             request->timeout - processing should not take longer than
*                                 this number of seconds
*             request->params[N-1] - pointers to item key parameters
*
*             result - structure that will contain result
*
* Return value: SYSINFO_RET_FAIL - function failed, item will be marked
*               as not supported by zabbix
*               SYSINFO_RET_OK - success
*
* Comment: get_rparam(request, N-1) can be used to get a pointer to the Nth
*          parameter starting from 0 (first parameter). Make sure it exists
*          by checking value of request->nparam.
*
********************************************/
static int      dummy_random(AGENT_REQUEST *request, AGENT_RESULT *result)
{
    char      *param1, *param2;
    int       from, to;
}

```

```

if (2 != request->nparam)
{
    /* set optional error message */
    SET_MSG_RESULT(result, strdup("Invalid number of parameters."));
    return SYSINFO_RET_FAIL;
}

param1 = get_rparam(request, 0);
param2 = get_rparam(request, 1);

/* there is no strict validation of parameters for simplicity sake */
from = atoi(param1);
to = atoi(param2);

if (from > to)
{
    SET_MSG_RESULT(result, strdup("Invalid range specified."));
    return SYSINFO_RET_FAIL;
}

SET_UI64_RESULT(result, from + rand() % (to - from + 1));

return SYSINFO_RET_OK;
}

/********************* Function: zbx_module_init ********************/
/*
 * Function: zbx_module_init
 *
 * Purpose: the function is called on agent startup
 *           It should be used to call any initialization routines
 *
 * Return value: ZBX_MODULE_OK - success
 *               ZBX_MODULE_FAIL - module initialization failed
 *
 * Comment: the module won't be loaded in case of ZBX_MODULE_FAIL
 *
*/
int      zbx_module_init(void)
{
    /* initialization for dummy.random */
    srand(time(NULL));

    return ZBX_MODULE_OK;
}

/********************* Function: zbx_module_uninit ********************/
/*
 * Function: zbx_module_uninit
 *
 * Purpose: the function is called on agent shutdown
 *           It should be used to cleanup used resources if there are any
 *
 * Return value: ZBX_MODULE_OK - success
 *               ZBX_MODULE_FAIL - function failed
 *
*/
int      zbx_module_uninit(void)
{
    return ZBX_MODULE_OK;
}

/********************* Functions: dummy_history_float_cb ********************/
/*
 * Functions: dummy_history_float_cb
 *            dummy_history_integer_cb
 *            dummy_history_string_cb
 *            dummy_history_text_cb
 *            dummy_history_log_cb
 *

```

```

* Purpose: callback functions for storing historical data of types float,      *
*           integer, string, text and log respectively in external storage      *
*
* Parameters: history      - array of historical data                      *
*              history_num - number of elements in history array            *
*
*****static void      dummy_history_float_cb(const ZBX_HISTORY_FLOAT *history, int      *
history_num)
{
    int      i;

    for (i = 0; i < history_num; i++)
    {
        /* do something with history[i].itemid, history[i].clock,
history[i].ns, history[i].value, ... */
    }
}

static void      dummy_history_integer_cb(const ZBX_HISTORY_INTEGER *history, int      *
history_num)
{
    int      i;

    for (i = 0; i < history_num; i++)
    {
        /* do something with history[i].itemid, history[i].clock,
history[i].ns, history[i].value, ... */
    }
}

static void      dummy_history_string_cb(const ZBX_HISTORY_STRING *history, int      *
history_num)
{
    int      i;

    for (i = 0; i < history_num; i++)
    {
        /* do something with history[i].itemid, history[i].clock,
history[i].ns, history[i].value, ... */
    }
}

static void      dummy_history_text_cb(const ZBX_HISTORY_TEXT *history, int history_num)
{
    int      i;

    for (i = 0; i < history_num; i++)
    {
        /* do something with history[i].itemid, history[i].clock,
history[i].ns, history[i].value, ... */
    }
}

static void      dummy_history_log_cb(const ZBX_HISTORY_LOG *history, int history_num)
{
    int      i;

    for (i = 0; i < history_num; i++)
    {
        /* do something with history[i].itemid, history[i].clock,
history[i].ns, history[i].value, ... */
    }
}

*****/* Function: zbx_module_history_write_cbs
*
* Purpose: returns a set of module functions Zabbix will call to export
*           different types of historical data
*
```

```

*
* Return value: structure with callback function pointers (can be NULL if
*               module is not interested in data of certain types)
*
***** */
ZBX_HISTORY_WRITE_CBS zbx_module_history_write_cbs(void)
{
    static ZBX_HISTORY_WRITE_CBS dummy_callbacks =
    {
        dummy_history_float_cb,
        dummy_history_integer_cb,
        dummy_history_string_cb,
        dummy_history_text_cb,
        dummy_history_log_cb,
    };

    return dummy_callbacks;
}

```

Модуль экспортирует 3 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 из диапазона param1-param2, например, `dummy.random[1,1000000]`

7.2.5.6 Ограничения

Поддержка подгружаемых модулей реализована только на платформах Unix. Это означает, что подгружаемый модуль не работает на Zabbix агентах под Windows.

В некоторых случаях модулю может потребоваться прочитать параметры имеющие отношения к модулю из `zabbix_agentd.conf`. В настоящее время этот функционал не поддерживается. Если вам необходимо в вашем модуле использовать некоторые параметры конфигурации вам, возможно, следует осуществлять анализ отдельного файла конфигурации модуля.

7.2.6 Счетчики 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, смотрите [специфичные ключи элементов данных для Windows](#).

Чтобы получить полный список счетчиков производительности для мониторинга, вы можете выполнить:

```
typerperf -qx
```

Числовое представление

В зависимости от настроек локализации, именования счетчиков производительности могут быть разными на разных серверах Windows. Такое поведение может внести определенные проблемы при создании шаблона для мониторинга нескольких Windows машин, использующих разные настройки локализации.

В то же время каждый счетчик производительности может быть переведен в цифровую форму, которая является уникальной и независимой от языковых настроек, так что вы можете использовать числовое представление, а не строковое.

Для того чтобы найти цифровые эквиваленты, выполните **regedit**, а затем найдите **HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows NT\CurrentVersion\Perflib\009**.

Запись в реестре содержит информацию наподобии этой:

```
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 вы можете найти соответствующие числа для каждой части строки счетчика производительности, такой как '\System\% Processor Time':

```
System -> 2  
% Processor Time -> 6
```

Затем вы можете использовать эти числа для преобразования пути в числа:

\2\6

Пользовательские параметры

Вы можете разворачивать некоторые PerfCounter параметры для мониторинга счетчиков производительности Windows.

Например, вы можете добавить следующее в файл конфигурации Zabbix агента:

```
PerfCounter=UserPerfCounter1, "\Memory\Page Reads/sec", 30  
или  
PerfCounter=UserPerfCounter2, "\4\24", 30
```

С такими параметрами, вы можете просто использовать *UserPerfCounter1* или *UserPerfCounter2* как ключи при создании соответствующих элементов данных.

Не забудьте перезапустить Zabbix агента после внесения изменений в файл конфигурации.

Решение проблем

Периодически Zabbix агент не может получить значения счётчиков производительности в системах на основе Windows 2000, потому что pdh.dll файл устарел. Такое поведение отображается

сообщениями об ошибках в файлах журналов Zabbix агента и сервера. В этом случае необходимо обновить pdh.dll на более новую 5.0.2195.2668 версию.

7.2.7 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.

Использование массового обновления

Для массового обновления каких-либо элементов данных, выполните следующее:

- Отметьте элементы данных из списка для обновления
- Выберите *Массовое обновление* из списка ниже
- Отметьте атрибуты для обновления
- Введите новые значения для этих атрибутов и нажмите на *Обновить*

7.2.8 Преобразование значений

Обзор

Для более “человеческого” представления полученных значений, вы можете использовать преобразования значений, которые содержат соответствия между числовыми значениями и строковыми представлениями.

Преобразования значений можно использовать как в веб-интерфейсе Zabbix, так и в оповещениях отправляемых по email/SMS/jabber и т.п.

Например, элемент данных, который имеет значения '0' или '1' может использовать преобразование значений для представления этих значений в более читаемой форме:

- '0' ⇒ 'Недоступен'
- '1' ⇒ 'Доступен'

Или преобразование значений для резервного копирования данных может быть такое:

- 'F' ⇒ 'Полный'
- 'D' ⇒ 'Дифференциальный'
- 'T' ⇒ 'Инкрементный'

Таким образом, при [настройке элементов данных](#) вы можете указать преобразование значений для более “человеческого” отображения этого элемента данных. Чтобы сделать это, вы должны выбрать имя ранее добавленного преобразования значения в поле *Отображение значения*.

Преобразование значений можно использовать с элементами данных, которые имеют *Числовой (с плавающей точкой)* и *Символ* типы информации.

Преобразования значений, начиная с Zabbix 3.0, можно экспортить/импортировать, как отдельно, так и с соответствующими шаблоном или узлом сети.

Настройка

Для того чтобы добавить преобразование значений:

- Перейдите в: *Администрирование* → *Общие*
- Выберите *Преобразование значений* из выпадающего меню
- Нажмите на *Создать преобразование значений* (или на имя уже существующего преобразования)

* Name	Windows service state	
* Mappings	Value	Mapped to
	0	⇒ Running
	1	⇒ Paused
	2	⇒ Start pending
	3	⇒ Pause pending
	4	⇒ Continue pending
	5	⇒ Stop pending
	6	⇒ Stopped
	7	⇒ Unknown
	255	⇒ No such service
	Add	
	Add	Cancel

Параметры преобразования значений:

Параметр

Описание

Имя Уникальное имя набора преобразования значений.

Соответствия Индивидуальные преобразования - пары числовых (буквенных) значений и их строковых представлений.

Все обязательные поля ввода отмечены красной звёздочкой.

Для добавления отдельного соответствия, нажмите на *Добавить*.

Как это работает

Например, один из предустановленных агентских элементов данных 'Ping to the server (TCP)' использует существующее преобразование значение именуемое 'Service state' для отображения своих значений.

* Name	Service state	
* Mappings	Value	Mapped to
	0	⇒ Down
	1	⇒ Up
	Add	
	Add	Cancel

В [диалоге настройки](#) элемента данных вы можете увидеть ссылку на это преобразование значений в поле *Отображение значения*:



Таким образом в *Мониторинг → Последние данные* преобразование подставляется для отображения 'Up' (вместе с исходным значением в скобках).

NAME	LAST CHECK	LAST VALUE
Zabbix agent (1 item)		
Agent ping	2015-08-11 22:01:07	Up (1)

В разделе *Последние данные* отображаемые значения сокращаются до 20 символов. Если используется преобразование значений, то усечение не применяется к значению соответствия, оно применяется только к сырому значению отдельно (отображается в круглых скобках).

Значение, которое отображается в более читаемой форме, также проще для понимания при получении оповещений.

Без предустановленного преобразования значений вы могли бы получить только следующее:

NAME	LAST CHECK	LAST VALUE
Zabbix agent (1 item)		
Agent ping	2015-08-11 22:09:21	1

Поэтому, в этом случае придется либо догадываться, что означает '1' или искать в документации, чтобы узнать значение этой цифры.

Type	<input type="checkbox"/>	Original
Host interface	<input type="checkbox"/>	Original
JMX endpoint	<input type="checkbox"/>	Original
URL	<input type="checkbox"/>	Original
Request body type	<input type="checkbox"/>	Original
Request body	<input type="checkbox"/>	Original
Headers	<input type="checkbox"/>	Original
SNMP community	<input type="checkbox"/>	Original
Context name	<input type="checkbox"/>	Original
Security name	<input type="checkbox"/>	Original
Security level	<input type="checkbox"/>	Original
Authentication protocol	<input type="checkbox"/>	Original
Authentication passphrase	<input type="checkbox"/>	Original
Privacy protocol	<input type="checkbox"/>	Original
Privacy passphrase	<input type="checkbox"/>	Original
Port	<input type="checkbox"/>	Original
Type of information	<input type="checkbox"/>	Original
Units	<input type="checkbox"/>	Original
Authentication method	<input type="checkbox"/>	Original
User name	<input type="checkbox"/>	Original
Public key file	<input type="checkbox"/>	Original
Private key file	<input type="checkbox"/>	Original
Password	<input type="checkbox"/>	Original
Preprocessing steps	<input type="checkbox"/>	Original
Update interval	<input type="checkbox"/>	Original
History storage period	<input checked="" type="checkbox"/>	<input type="text" value="7d"/>
Trend storage period	<input type="checkbox"/>	Original
Status	<input type="checkbox"/>	Original
Log time format	<input type="checkbox"/>	Original
Show value	<input type="checkbox"/>	Original
Enable trapping	<input type="checkbox"/>	Original
Allowed hosts	<input type="checkbox"/>	Original
Replace applications	<input type="checkbox"/>	Original
Add new or existing applications	<input type="checkbox"/>	Original
Master item	<input type="checkbox"/>	Original
Description	<input type="checkbox"/>	Original

Заменить группы элементов данных удалит элемент данных из существующих групп элементов данных и заменит на те, что перечислены в этом поле.

Добавить новую или существующую группу элементов данных позволяет указать дополнительные группы элементов данных из существующих или создать совершенно новые группы элементов данных для элементов данных.

Оба эти поля имеют функцию автодополнения - начните вводить и в них автоматически появится выпадающий список совпадающих групп элементов данных. Если группа элементов данных новая, то она также появится в выпадающем списке и будет обозначена (*новый*) в конце строки. Просто прокрутите список и выберете нужное.

7.2.9 Группы элементов данных

Обзор

Группы элементов данных используются для группировки элементов данных в логические группы.

Например, группа элементов данных *MySQL Server* может содержать все элементы данных связанные с MySQL сервером: доступность MySQL, дисковое пространство, загрузка процессора, транзакции в секунду, количество медленных запросов и прочее.

Группы элементов данных также используются для группировки веб-сценариев.

Если вы используете группы элементов данных, то в *Мониторинг → Последние данные* вы увидите элементы данных и веб-сценарии сгруппированные в соответствии со своими группами элементов данных.

Настройка

Для работы с группами элементов данных вы должны сначала создать их и затем присоединить к ним элементы данных или веб-сценарии.

Для создания группы элементов данных, сделайте следующее:

- Перейдите в *Настройка → Узлы сети или Шаблоны*
- Далее нажмите на *Группы элементов данных* у требуемого узла сети или шаблона
- Нажмите на *Создать группу элементов данных*
- Ведите имя группы элементов данных и нажмите на *Добавить*, чтобы сохраните её

The screenshot shows the Zabbix interface under 'Applications'. At the top, there's a navigation bar with 'All hosts / Zabbix server' and 'Enabled' status. Below it are tabs for 'ZBX', 'SNMP', 'JMX', and 'IPMI', with 'Applications 3' and 'Items 33' also visible. A modal dialog box is open in the center, prompting for a 'Name' (marked with a red asterisk) in a text input field. Below the input field are two buttons: 'Add' (in blue) and 'Cancel' (in grey).

Вы можете также создать новую группу элементов данных напрямую из диалога свойств элемента данных.

Элементы данных соединяются с группами элементов данных в диалоге свойств элементов данных. Выберите одну или несколько групп элементов данных, которым элемент данных будет принадлежать.

Веб-сценарии соединяются с группами элементов данных в диалоге добавления веб-сценария. Выберите группу элементов данных, которой сценарий будет принадлежать.

7.2.10 Очередь

Обзор

Очередь отображает элементы данных, которые ожидают обновления. Очередь просто **логическое** представление данных. Это не очередь IPC и не какой-то другой механизм очереди в Zabbix.

Элементы данных наблюдаемые через прокси также включены в очередь - они будут считаться задерживающимися за период обновления данных истории с прокси.

В очереди отображаются только те элементы данных, которые имеют запланированное время обновления. Это означает, что следующие типы элементов данных исключены из очереди:

- активные элементы данных Zabbix агента log, logrt и журнал событий
- элементы данных SNMP трапов
- траппер элементы данных
- элементы данных веб-мониторинга

Статистика, представленная в очереди, является хорошим показателем производительности сервера Zabbix.

Очередь запрашивается напрямую с Zabbix сервера с использованием JSON протокола. Информация доступна только, если Zabbix сервер запущен.

Чтение очереди

Для чтения очереди, перейдите в *Администрирование → Очередь*. В правом верхнем углу в выпадающем списке должен быть выбран *Обзор*.

ITEMS	5 SECONDS	10 SECONDS	30 SECONDS	1 MINUTE	5 MINUTES	MORE THAN 10 MINUTES
Zabbix agent	0	0	0	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	1	0	5	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
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

В целом это изображение “зеленое”. Поэтому мы можем считать, что сервер работает хорошо.

Очередь показывает один элемент данных ожидающий около 10 секунд и один ожидающий около 5 минут. Приятно, было бы здорово знать, что это за элементы данных.

Чтобы сделать это, выберите *Детали* в верхнем правом углу из выпадающего меню. Теперь вы можете видеть список этих задерживающихся элементов данных.

SCHEDULED CHECK	DELAYED BY	HOST	NAME
2015-08-11 22:43:51	12s	Remote proxy: Zabbix server	Zabbix history write cache, % free
2015-08-11 22:43:52	11s	Remote proxy: Zabbix server	Zabbix text write cache, % free
2015-08-11 22:43:53	10s	Remote proxy: Zabbix server	Zabbix trend write cache, % free
2015-08-11 22:43:54	9s	Remote proxy: Zabbix server	Values processed by Zabbix server per second

При наличии таких деталей имеется возможность выяснить почему эти элементы данных могут задерживаться.

При задержке одного или двух элементов данных, возможно, нет причин для беспокойства. Они могут быть обновлены с секунды на секунду. Однако, если вы видите большое количество элементов данных ожидающих обновления длительное время, это может стать серьезной проблемой.

ITEMS	5 SECONDS	10 SECONDS	30 SECONDS	1 MINUTE	5 MINUTES	MORE THAN 10 MINUTES
Zabbix agent	0	13	7	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	5	1	9	0	0	0

Агент недоступен?

Элемент данных очереди

Можно использовать специальный внутренний элемент данных **zabbix[очередь,<от>,<до>]** для наблюдения за состоянием очереди Zabbix. Он будет возвращать количество элементов данных задерживающихся за указанный период времени. Для более подробной информации смотрите [Внутренние элементы данных](#).

7.2.11 Кэш значений

Обзор

Чтобы сделать вычисления выражений триггеров, вычисляемых/аггрегированных элементов данных и некоторых макросов, начиная с Zabbix 2.2 в Zabbix сервере поддерживается опция кэша значений.

Этот кэш находящийся в оперативной памяти используется для доступа к данным истории, вместо выполнения прямых запросов SQL в базу данных. Если данные истории не присутствуют в кэше, недостающие значения запрашиваются из базы данных и, соответственно, кэш обновляется.

Для включения функционала кэша значений, поддерживается дополнительный параметр **ValueCacheSize** в [файле конфигурации](#) Zabbix server.

Два внутренних элемента данных поддерживаются для наблюдения за кэшем значений **zabbix[vcache,buffer,<режим>]** и **zabbix[vcache,cache,<параметр>]**. Для получения более детальных сведений смотрите [внутренние элементы данных](#).

7.2.12 Проверить сейчас

Обзор

Проверка нового значения элемента данных в Zabbix циклический процесс, который основывается на заданных интервалах обновления. Хотя у многих элементов данных интервалы обновления достаточно маленькие, имеются и другие (включая правила низкоуровневого обнаружения) у которых интервалы обновления весьма большие, таким образом в реальных ситуациях в жизни может

потребоваться более быстрая проверка нового значения - чтобы подхватить изменения в обнаруживаемых ресурсах, например. Чтобы покрыть подобную необходимость, имеется возможность изменения расписания пассивной проверки и получения нового значения немедленно.

Эта функция поддерживается только для **пассивных** проверок. Поддерживаются следующие типы элементов данных:

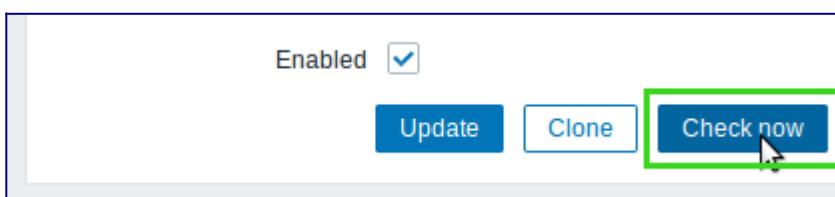
- Zabbix агент (пассивный)
- SNMPv1/v2/v3 агент
- IPMI агент
- Простая проверка
- Zabbix внутренний
- Zabbix агрегированный
- Внешняя проверка
- Монитор баз данных
- JMX агент
- SSH агент
- Telnet
- Вычисляемый
- HTTP агент

Проверка должна существовать в кэше конфигурации, чтобы выполнение было успешным; для получения более подробных сведений смотрите [CacheUpdateFrequency](#). При проверке нового значения кэш конфигурации **не** обновляется, поэтому значения не будут отражать совсем недавние изменения конфигураций элементов данных/правил обнаружения. Таким образом, также отсутствует возможность проверки нового значения по элементу данных/правилу обнаружения, которые созданы только что.

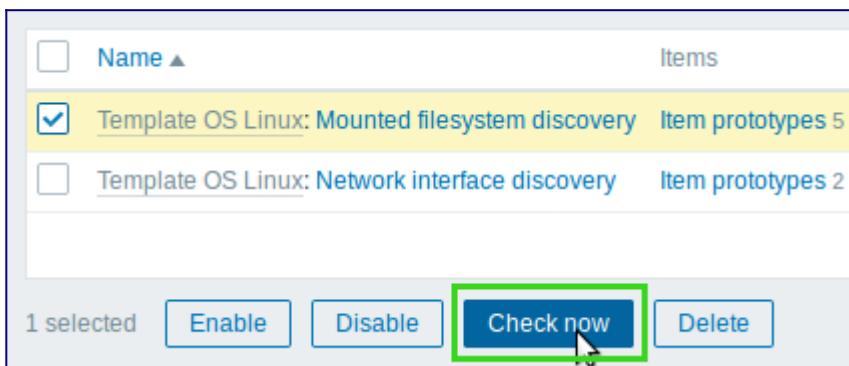
Настройка

Для выполнения немедленной пассивной проверки:

- нажмите на *Проверить сейчас* в диалоге настройки существующего элемента данных (или правила обнаружения):



- нажмите на *Проверить сейчас* для выбранных элементов данных/правил в списке элементов данных/правил обнаружения:



В последнем случае можно выбрать несколько элементов данных/правил и выполнить “проверку сейчас” сразу по всем выбранным объектам.

7.3 Триггеры

Обзор

Триггеры это логические выражения, которые “оценивают” данные собранные элементами данных и отражают текущее состояние системы.

Тогда как элементы данных используются для сбора системных данных, крайне непрактично наблюдать за этими данными всё время в ожидании выполнения условия, которое приведет к оповещению или которое заслуживает внимания. Работа по “оценке” данных может быть оставлена на выражения триггеров.

Выражения триггеров позволяют задать порог при котором состояние данных “приемлемое”. Таким образом, если входящие данные превышают приемлемое состояние, триггер “поджигается” - или другими словами меняет состояние на ПРОБЛЕМА.

Триггер может принимать следующие состояния:

ЗНАЧЕНИЕ	ОПИСАНИЕ
OK	Нормальное состояние триггера. В более ранних версиях Zabbix оно называлось ЛОЖЬ.
ПРОБЛЕМА	Обычно означает, что что-то случилось. Например, загрузка процессора слишком высокая. В более ранних версиях Zabbix оно называлось ИСТИНА.
Состояние триггера (выражение)	пересчитывается каждый раз, когда Zabbix сервер получает новое значение, которое является частью выражения.

Если в выражении триггера используются функции, связанные со временем (**nodata()**, **date()**, **dayofmonth()**, **dayofweek()**, **time()**, **now()**), то тогда триггер будет пересчитываться каждые 30 секунд процессом Zabbix *синхронизации истории*. Если используются оба типа функций связанные со временем и не связанные со временем в одном выражении, то пересчёт выполняется при поступлении нового значения **и же** каждые 30 секунд.

Вы можете [создавать выражения триггеров](#) с различной степенью сложности.

7.3.1 Настройка триггера

Обзор

Для настройки триггера, сделайте следующее:

- Перейдите в: *Настройка* → *Узлы сети*
- Нажмите на *Триггеры* в строке с узлом сети
- Нажмите на *Создать триггер* сверху справа (или на имя триггера для изменения уже существующего триггера)
- Введите в диалоге параметры триггера

Настройка

Вкладка **Триггер** содержит все существенные атрибуты триггера.

Trigger Dependencies

* Name	Disk I/O is overloaded on {HOST.NAME}				
Severity	Not classified	Information	Warning	Average	High
* Expression	{Zabbix server:system.cpu.util[iowait].avg(5m)}>20 and {Zabbix server:system.uname.str(Linux)}=1				
Expression constructor					
OK event generation	<input checked="" type="radio"/> Expression <input type="radio"/> Recovery expression <input type="radio"/> None				
PROBLEM event generation mode	<input checked="" type="radio"/> Single <input type="radio"/> Multiple				
OK event closes	<input checked="" type="radio"/> All problems <input type="radio"/> All problems if tag values match				
Tags	<input type="text"/> tag <input type="text"/> value				
Add					
Allow manual close	<input type="checkbox"/>				
URL					
Description	OS spends significant time waiting for I/O (input/output) operations. It could be indicator of performance issues with storage system.				
Enabled	<input checked="" type="checkbox"/>				
Add Cancel					

Все обязательные поля ввода отмечены красной звёздочкой.

Параметр	Описание
Имя	<p>Имя триггера. Имя может содержать поддерживаемые макросы: {HOST.HOST}, {HOST.NAME}, {HOST.CONN}, {HOST.DNS}, {HOST.IP}, {ITEM.VALUE}, {ITEM.LASTVALUE} и {\$MACRO}.</p> <p>Можно использовать макросы \$1, \$2...\$9 как ссылки на первую, вторую... девятую константу из выражения.</p> <p><i>Обратите внимание:</i> Макросы \$1-\$9 будут раскрыты корректно, если константы на которые ссылаются макросы, в относительно простых, понятных выражениях. Например, имя “Загрузка процессора превышает \$1 на {HOST.NAME}” будет автоматически изменено на “Загрузка процессора превышает 5 на New host”, если выражение - {New host:system.cpu.load[percprc,avg1].last()}>5</p>
Важность	Определение требуемой важности триггера нажатием кнопок.
Выражение проблемы	Логическое выражение , используемое для определения условий проблемы.
Формирование OK	Опции формирования OK событий:

Параметр	Описание
событий	<p>Выражение - ОК события формируются на основе того же выражения что и события о проблемах;</p> <p>Выражение восстановления - ОК события формируются, если выражение проблемы вычисляется как ЛОЖЬ и выражение восстановления вычисляется как ПРАВДА;</p> <p>Нет - в этом случае триггер никогда самостоятельно не вернётся в состояние ОК.</p> <p>Поддерживается начиная с Zabbix 3.2.0.</p> <p>Логическое выражение, используемое для определения условий, когда проблема решена.</p> <p>Выражение восстановления вычисляется только после того, как выражение проблемы будет вычислено как ЛОЖЬ. Невозможно решить проблему с помощью выражения восстановления, если условие проблемы всё ещё присутствует.</p> <p>Это поле опционально и доступно только, если в <i>Формирование ОК событий</i> выбрано 'Выражение восстановления'.</p> <p>Поддерживается начиная с Zabbix 3.2.0.</p>
Выражение восстановления	<p>Режим формирования события о проблемах:</p> <p>Одиночный - формируется одно событие, когда триггер переходит в состояние 'Проблема' в первый раз;</p> <p>Множественный - событие формируется на <i>каждое</i> 'Проблема' вычисление триггера.</p> <p>Выберите, если ОК событие закрывает:</p> <p>Все проблемы - все проблемы этого триггера</p> <p>Все проблемы, если значение тега совпадает - только те проблемы триггера, значения тегов событий которых совпадают</p> <p>Поддерживается начиная с Zabbix 3.2.0.</p> <p>Укажите имя тега события, которое будет использоваться для корреляции событий.</p> <p>Это поле отображается, если в свойстве <i>OK событие закрывает</i> выбрано 'Все проблемы, если значения тегов совпадают' и в этом случае это поле обязательно.</p> <p>Поддерживается начиная с Zabbix 3.2.0.</p> <p>Укажите пользовательские теги для маркировки событий триггера.</p> <p>Теги событий могут использоваться для корреляции событий, в условиях действий и будут также видны в <i>Мониторинг → Проблемы</i> и в <i>Проблемы</i> виджете.</p> <p>Тегами является пара имя тега и значение. Вы можете использовать только имя или пару имя со значением.</p> <p>Пользовательские макросы, контекст пользовательских макросов, макросы низкоуровневого обнаружения и макросы функций <code>{{ITEM.VALUE}}</code>, <code>{{ITEM.LASTVALUE}}</code>, а также макросы низкоуровневого обнаружения, поддерживаются в тегах событий. Макросы низкоуровневого обнаружения можно использовать в контексте макросов.</p> <p>Если суммарная длина раскрытоого значения превышает, оно будет обрезано до 255 символов.</p> <p>Смотрите все поддерживаемые макросы тегами событий.</p> <p>Поддерживается начиная с Zabbix 3.2.0.</p> <p>Выберите, чтобы разрешить закрывать вручную события о проблемах сформированные этим триггером. Закрытие вручную возможно при подтверждении событий о проблемах.</p> <p>Поддерживаются начиная с Zabbix 3.2.0.</p>
Разрешить закрывать вручную	<p>Если не пустой, то URL, указанный здесь, доступен в виде ссылки в нескольких местах веб-интерфейса, например при нажатии на имя проблемы в <i>Мониторинг → Проблемы</i> (опция <i>URL</i> в меню <i>Триггер</i>) и в виджете <i>Проблемы</i></p>
URL	

Параметр	Описание
на панели.	Поддерживаемые макросы: {ITEM.VALUE}, {ITEM.LASTVALUE}, {TRIGGER.ID}, несколько {HOST.*} макросов, пользовательские макросы.
Описание	Текстовое поле используется, для того чтобы сообщить больше информации об этом триггере. Может содержать инструкции по устранению специфичной проблемы, контактные данные ответственного сотрудника и т.п. Начиная с Zabbix 2.2, описание может содержать тот же набор макросов, что и имя триггера.
Активировано	Уберите отметку деактивирует триггер, если это требуется.
Вкладка Зависимости содержит все зависимости триггера.	
Нажмите на <i>Добавить</i> для добавления новой зависимости.	
Вы также можете настроить триггер, открыв уже существующий триггер, затем нажать на кнопку <i>Клонировать</i> и сохранить его под другим именем.	

Тестирование выражений

Имеется возможность проверки настроенного выражения триггера относительного того, как результат выражения будет меняться в зависимости от полученного значения.

Следующее выражение с официального шаблона взято в качестве примера:

```
{Template Net Cisco IOS SNMPv2:sensor.temp.value[ciscoEnvMonTemperatureValue.
{#SNMPINDEX}].avg(5m)}>{$TEMP_WARN}
or
{Template Net Cisco IOS SNMPv2:sensor.temp.status[ciscoEnvMonTemperatureState.
{#SNMPINDEX}].last(0)}={$TEMP_WARN_STATUS}
```

Для тестирования выражения, нажмите на *Конструктор выражения* под полем выражения.

The screenshot shows the 'Trigger' tab selected in the navigation bar. Below it, the 'Dependencies' tab is also present. The main configuration area includes fields for 'Name' (set to 'Template OS CISCO: Temperature too high'), 'Severity' (set to 'Warning'), and an 'Expression' field containing the provided template code. At the bottom of this section is a green-bordered button labeled 'Expression constructor'.

В Конструкторе выражения перечислены все отдельные выражения. Чтобы открыть окно тестирования, нажмите на *Тест* ниже списка выражений.

This screenshot shows the 'Test' dialog box. It contains a 'Target Expression' section with a tree view. The root node 'Or' has two children: 'A' and 'B'. Node 'A' corresponds to the first part of the expression, and node 'B' corresponds to the second. A green-bordered button labeled 'Test' is located at the bottom left of the dialog.

В окне тестирования вы можете задать пробные значения ("80, 70, 0, 1" в этом примере) и затем увидеть результат выражения, нажав на кнопку *Тест*.

Test

Test data	Expression Variable Elements	Result type	Value
	{Template Net Cisco IOS SNMPv2:sensor.temp.value[ciscoEnvMonTemperatureValue.(#SNMPINDEX)].avg(5m)}	Numeric (float)	80
	{\$TEMP_WARN}	Numeric (float)	70
	{Template Net Cisco IOS SNMPv2:sensor.temp.status[ciscoEnvMonTemperatureState.(#SNMPINDEX)].last(0)}	Numeric (integer)	0
	{\$TEMP_WARN_STATUS}	Numeric (float)	1

Result	Expression	Result
	Or	
	└ A {Template Net Cisco IOS SNMPv2:sensor.temp.value[ciscoEnvMonTemperatureValue.(#SNMPINDEX)].avg(5m)]>{\$TEMP_WARN}}	TRUE
	└ B {Template Net Cisco IOS SNMPv2:sensor.temp.status[ciscoEnvMonTemperatureState.(#SNMPINDEX)].last(0)}=={\$TEMP_WARN_STATUS}	FALSE
	A or B	TRUE

Test Cancel

Можно увидеть результат отдельных выражений, наряду со полным выражением.

Результат “ПРАВДА” означает, что введённое выражение корректно. В этом конкретном выражении А, “80” больше чем указанное значение {\$TEMP_WARN}, “70” в этом примере. Соответственно, появится результат “ПРАВДА”.

Результат “ЛОЖЬ” означает, что введёное выражение некорректно. В этом конкретном выражении В, {\$TEMP_WARN_STATUS}, равно “1” в этом примере, должно быть равно указанному значению 0, и это неправда. Соответственно, появится результат “ЛОЖЬ”.

Выбранным типом выражения является “ИЛИ”/“ПРАВДА”. Если по крайней мере одно из заданных выражений (А или В в этом случае) будет ПРАВДА, общий результат будет также ПРАВДА. Что означает, что текущее значение превысило пороговое значение и возникнет Проблема.

7.3.2 Выражение триггера

Обзор

Используемые в триггерах выражения являются очень гибкими. Вы можете использовать их для создания сложных логических тестов, учитывая статистику по мониторингу.

Простое полезное выражение может выглядеть примерно так:

{<сервер>:<ключ>.<функция>(<параметр>)}<оператор><константа>

Функции

Функции триггеров позволяют ссылаться на собранные значения, текущее время и другие факторы.

Имеется полный список [поддерживаемых функций](#).

Параметры функций

Большинство числовых функций принимают количество секунд в качестве параметра.

Вы можете использовать префикс #, чтобы указать что этот параметр должен иметь другой смысл:

ВЫЗОВ ФУНКЦИИ

sum(600)

Сумма всех значений за 600 секунд

sum(#5)

Сумма последних 5 значений

СМЫСЛ

Функция **last** использует другой смысл для значений, когда начинается с решетки - она дает выбрать п-ое предыдущее значение, так что с учетом значений 3, 7, 2, 6, 5 (от наиболее нового до наиболее старого), при **last(#2)** вернется 7 и при **last(#5)** вернется 5.

Несколько функций поддерживают дополнительный, второй параметр `сдвиг_времени`. Этот параметр позволяет ссылаться на данные из периода времени в прошлом. Например, для `avg(1h,1d)` будет возвращено среднее значение за час днем ранее.

Вы можете использовать поддерживаемые [суффиксы преобразований](#) в выражениях триггеров, например, '5m' (минут) вместо '300' секунд или '1d' (день) вместо '86400' секунд. '1K' будет состоять из '1024' байт.

Операторы

Следующие операторы поддерживаются для триггеров (**представлены по убыванию приоритета выполнения**):

ПРИОРИТЕТ	ОПЕРАТОР	ОПРЕДЕЛЕНИЕ	Заметки по <u>неизвестным значениям</u>
1	-	Унарный минус	-Неизвестно → Неизвестно
2	not	Логическое НЕ	not Неизвестно → Неизвестно
3	*	Умножение	0 * Неизвестно → Неизвестно (да, Неизвестно, не 0 - чтобы не потерять Неизвестно в арифметических операциях)
	/	Деление	1.2 * Неизвестно → Неизвестно Неизвестно / 0 → ошибка Неизвестно / 1.2 → Неизвестно 0.0 / Неизвестно → Неизвестно
4	+	Арифметический плюс	1.2 + Неизвестно → Неизвестно
	-	Арифметический минус	1.2 - Неизвестно → Неизвестно
5	<	Менее чем. Этот оператор может быть представлен в виде: $A < B \Leftrightarrow (A < B - 0.000001)$	1.2 < Неизвестно → Неизвестно
	\leq	Менее чем или равно. Этот оператор может быть представлен в виде: $A \leq B \Leftrightarrow (A \leq B + 0.000001)$	Неизвестно \leq Неизвестно → Неизвестно
	>	Более чем. Этот оператор может быть представлен в виде: $A > B \Leftrightarrow (A > B + 0.000001)$	
	\geq	Более чем или равно. Этот оператор может быть представлен в виде: $A \geq B \Leftrightarrow (A \geq B - 0.000001)$	
6	=	Равенство. Этот оператор может быть представлен в виде: $A = B \Leftrightarrow (A \geq B - 0.000001) \text{ и } (A \leq B + 0.000001)$	

Не равно. Этот оператор может быть представлен в виде:

<>

$A <> B \Leftrightarrow (A < B - 0.000001) \text{ или } (A > B + 0.000001)$

7 **and** Логическое И

0 **and** Неизвестно → 0

1 **and** Неизвестно →

Неизвестно

Неизвестно **and** Неизвестно → Неизвестно

1 **or** Неизвестно → 1

0 **or** Неизвестно →

Неизвестно

Неизвестно **or** Неизвестно → Неизвестно

8 **or** Логическое ИЛИ

Операторы **not**, **and** and **or** регистрозависимы и должны быть в нижнем регистре. Они также должны быть окружены символами пробелов или круглыми скобками.

Все операторы, кроме унарных - и **not**, имеют ассоциативность слева на право. Унарные - и **not** не ассоциативны (имеется в виду необходимо использовать **-(-1)** и **not (not 1)** вместо **--1** и **not not 1**).

Результат вычисления:

- Операторы <, <=, >, >=, =, <> должны давать '1' в выражении триггера, если указанное соотношение правдиво и '0', если оно ложно. Если по крайней мере один операнд Неизвестен, то и результат будет Неизвестно;
- **and** по известным operandам должно давать '1', если оба из этих operandов сравнения не равны '0'; в противном случае, будет давать '0'; для неизвестных operandов **and** даст '0' только, если один из operandов сравнения равен '0'; в противном случае, он даст 'Неизвестно';
- **or** по известным operandам должно давать '1', если какой-либо из этих operandов сравнения не равен '0'; в противном случае, будет давать '0'; для неизвестных operandов **or** даст '1' только, если один из operandов сравнения не равен '0'; в противном случае, он даст 'Неизвестно';
- Результат логического operandя отрицания **not** для известного operandя равен '0', если значение этого operandя сравнения не равно '0'; '1', если значение его operandя сравнения равно '0'. Для неизвестных operandов **not** даст 'Неизвестно'.

Кэширование значений

Значения, которые требуются для вычисления триггеров, кэшируются Zabbix сервером. По этой причине такое вычисление триггеров на некоторое время приводит к более высокой загрузке базы данных после перезапуска сервера. Кэш значений не очищается, когда значения истории элементов данных удаляются (либо вручную, либо при помощи автоматической очистки истории), поэтому сервер будет использовать кэшированные значения пока они не станут старше, чем периоды времени, которые заданы в функциях триггеров, либо пока сервер не будет перезапущен.

Примеры триггеров

Пример 1

Высокая загрузка процессора на www.zabbix.com.

```
{www.zabbix.com:system.cpu.load[all,avg1].last()}>5
```

'www.zabbix.com:system.cpu.load[all,avg1]' представляет короткое имя наблюдаемого параметра. Эта строка указывает, что сервер - 'www.zabbix.com' и наблюдаемый ключ - 'system.cpu.load[all,avg1]'. Используя функцию 'last()', мы ссылаемся на самое последнее значение. И наконец '>5' означает, что

триггер перейдет в состояние ПРОБЛЕМА всякий раз, когда самое новое измерение загрузки процессора на сервере www.zabbix.com будет превышать 5.

Пример 2

www.zabbix.com перегружен

```
{www.zabbix.com:system.cpu.load[all,avg1].last()}>5 or  
{www.zabbix.com:system.cpu.load[all,avg1].min(10m)}>2
```

Это выражение будет истинным, когда либо текущая загрузка процессора станет более 5, либо загрузка процессора больше значения 2 за последние 10 минут.

Пример 3

/etc/passwd был изменен

Используется функция diff:

```
{www.zabbix.com:vfs.file.cksum[/etc/passwd].diff()}=1
```

Это выражение будет истинным, когда предыдущее значение контрольной суммы файла /etc/passwd отличается от самого нового значения.

Аналогичные выражения могут быть полезны для мониторинга изменений в важных файлах, таких как /etc/passwd, /etc/inetd.conf, /kernel и других.

Пример 4

Кто-то скачивает большой файл из Интернет

Используется функция min:

```
{www.zabbix.com:net.if.in[eth0,bytes].min(5m)}>100K
```

Это выражение будет истинным, когда количество полученных байт на eth0 превышает 100 КБ за последних 5 минут.

Пример 5

Оба узла кластера SMTP серверов недоступны

Примечание, в выражении используются два разных узла сети:

```
{smtp1.zabbix.com:net.tcp.service[smtp].last()}=0 and  
{smtp2.zabbix.com:net.tcp.service[smtp].last()}=0
```

Это выражение будет истинным, когда оба SMTP сервера недоступны на обоих smtp1.zabbix.com и smtp2.zabbix.com.

Пример 6

Zabbix агент нуждается в обновлении

Используется функция str():

```
{zabbix.zabbix.com:agent.version.str("beta8")}=1
```

Это выражение будет истинным, когда версия Zabbix агента содержит в себе 'beta8' (возможно 1.0beta8).

Пример 7

Сервер недоступен

```
{zabbix.zabbix.com:icmpping.count(30m,0)}>5
```

Это выражение будет истинным, если узел сети “zabbix.zabbix.com“ недоступен более 5 раз за последние 30 минут.

Пример 8

Нет данных за последние 3 минуты

Используется функцию nodata():

```
{zabbix.zabbix.com:tick.nodata(3m)}=1
```

Для того, чтобы этот триггер заработал, элемент данных ‘tick’ должен быть задан как элемент данных типа Zabbix [траппер](#). Узел сети должен периодически отправлять данные этому элементу данных, используя zabbix_sender. Если не было получено данных за последние 180 секунд, значением триггера станет ПРОБЛЕМА.

Обратите внимание, что 'nodata' можно использовать с любым типом элементов данных.

Пример 9

Активность CPU в ночное время

Используется функция time():

```
{zabbix:system.cpu.load[all,avg1].min(5m)}>2 and  
{zabbix:system.cpu.load[all,avg1].time()}>000000 and  
{zabbix:system.cpu.load[all,avg1].time()}<060000
```

Триггер может изменить свое состояние в истинное только в ночное время (00:00-06:00).

Пример 10

Проверка синхронизации времени на клиенте со временем на Zabbix сервере

Используется функция fuzzytime():

```
{MySQL_DB:system.localtime.fuzzytime(10)}=0
```

Триггер изменит состояние на проблему тогда, когда локальное время на сервере MySQL_DB и Zabbix сервере различаются более чем на 10 секунд.

Пример 11

Сравнение средней загрузки сегодня со средним значением загрузки за это же время вчера (использование второго параметра Сдвиг_времени).

```
{server:system.cpu.load.avg(1h)}/{server:system.cpu.load.avg(1h,1d)}>2
```

Триггер изменит свое состояние на проблему, если средняя загрузка за последний час будет в два раза больше чем за аналогичный период времени вчера.

Пример 12

Использование значение другого элемента данных в качестве порогового значения триггера:

```
{Template PfSense:hrStorageFree[#SNMPVALUE].last()}<{Template  
PfSense:hrStorageSize[#SNMPVALUE].last()}*0.1
```

Триггер изменит свое состояние на проблему, если свободное пространство на диски упадет ниже 10 процентов.

Пример 13

Использование [результата вычисления](#) для получения количества триггеров больше порога:

```
({server1:system.cpu.load[all,avg1].last()}>5) +  
({server2:system.cpu.load[all,avg1].last()}>5) +  
({server3:system.cpu.load[all,avg1].last()}>5)>=2
```

Триггер изменит свое состояние на проблему, если по крайней мере два триггера из выражения будут больше 5.

Гистерезис

Порой нам необходим интервал между состояниями OK и Проблема, а не просто порог. Например, мы бы хотели задать триггер, который переходит в состояние Проблема, когда температура в серверной комнате становится больше 20С и мы бы хотели чтобы он оставался в этом состоянии пока температура не опустится ниже 15С.

Чтобы это сделать, сначала мы зададим выражение триггера для события о проблеме. Затем выберем 'Выражение восстановления' в *Формирование OK события* и укажем выражение восстановления для OK события.

Обратите внимание, что выражение восстановления будет вычислено только при первом решении события о проблеме. Невозможно решить проблему при помощи выражения восстановления, если условие проблемы всё еще присутствует.

Пример 1

Температура в серверной комнате слишком высокая.

Выражение проблемы:

```
{server:temp.last()}>20
```

Выражение восстановления:

```
{server:temp.last()}<=15
```

Пример 2

Очень мало свободного места на диске

Выражение проблемы: если меньше 10ГБ за последние 5 минут

```
{server:vfs.fs.size[/,free].max(5m)}<10G
```

Выражение восстановления: если больше 40ГБ за последние 10 минут

```
{server:vfs.fs.size[/,free].min(10m)}>40G
```

Выражения с неподдерживаемыми элементами данных и неизвестными значениями

Версии до Zabbix 3.2 очень строго относились к неподдерживаемым элементам данных в выражениях триггеров. Любой неподдерживаемый элемент данных в выражении незамедлительно менял значение триггера на Неизвестно.

Начиная с Zabbix 3.2 существует более гибкий подход к неподдерживаемым элементам данных, допуская неизвестные значения при вычислении выражений:

- У некоторых функций их значения не зависят от того поддерживается ли элемент данных или нет. Такие функции теперь вычисляются даже, если ссылаются на неподдерживаемые элементы данных. Смотрите список в разделе [функции и неподдерживаемые элементы данных](#).
- Логические выражения с ИЛИ и И могут быть вычислены для известных значений в двух случаях независимо от неизвестных операндов:
 - “1 or Неподдерживаемый_элемент_данных1.некая_функция() or Неподдерживаемый_элемент_данных2. некая_функция() or ...” может быть вычислена как '1' (Правда),

- “0 and Неподдерживаемый_элемент_данных1. некая_функция() and Неподдерживаемый_элемент_данных2. некая_функция() and ...” может быть вычислена как '0' (Ложь).
Zabbix пытается вычислить логические выражения принимая неподдерживаемые элементы данных как Неизвестные значения. В двух случаях, упомянутых выше, будет приниматься известное значение; в остальных случаях значением триггера будет Неизвестно.
- Если вычисление триггера по поддерживаемому элементу данных приведет к ошибке, значением функции будет Неизвестно и оно будет частью дальнейшего вычисления выражения.

Обратите внимание на то, что неизвестные значения могут “исчезать” только в логических выражениях описанных выше. В арифметических выражениях неизвестные Неизвестному результату (за исключением деления на 0).

Если выражение триггера с несколькими неподдерживаемыми элементами данных вычисляется как Неизвестное, сообщение об ошибке в веб-интерфейсе ссылается на последний вычисленный неподдерживаемый элемент данных.

7.3.3 Зависимости триггеров

Обзор

Временами доступность одного узла сети зависит от другого. Сервер, который находится за каким-нибудь роутером будет недоступным, если сам роутер станет недоступным. При наличии настроенных триггеров на обоих устройствах, вы можете получить оповещения сразу для обоих недоступных узлов сети - в то время как виноват только роутер.

В этом случае некоторая зависимость между узлами сети может оказаться очень полезной. С наличием зависимости, набор оповещений о зависимых проблемах, может быть не отправлен и будет отправлено только одно оповещение о корне проблемы.

Хотя Zabbix не поддерживает зависимости между узлами сети напрямую, с другой стороны они могут быть заданы с помощью другого более гибкого метода - зависимостями между триггерами. Триггер может иметь один или более триггеров зависимых от него.

Таким образом в нашем простом примере мы откроем форму настройки триггера и укажем этому триггеру зависимость от триггера роутера. При такой зависимости зависящий триггер сервера от триггера роутера не изменит своё состояние пока триггер роутера находится в состоянии 'ПРОБЛЕМА' - и следовательно зависимые действия не будут предприняты и оповещения не будут отправлены.

Если и сервер, и роутер станут недоступными и зависимость будет настроена, Zabbix не выполнит действия для зависимого триггера.

Действия на зависимые триггера не будут выполняться, если триггер от которого они зависят:

- меняет своё состояние с 'ПРОБЛЕМА' на 'НЕИЗВЕСТНО'
- закрыт вручную, корреляцией или при помощи функций связанных со временем
- решен значением элемента данных, не участвующего в зависимом триггере
- деактивирован, имеет деактивированный элемент данных или узел элемента данных деактивирован

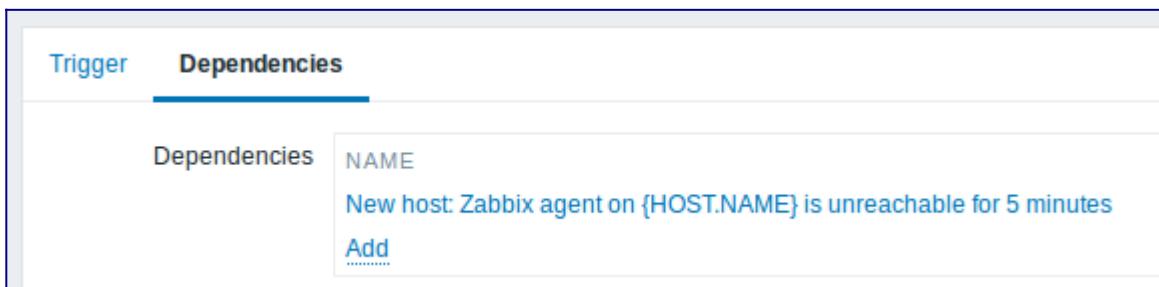
Обратите внимание, что “второй” (зависимый) триггер во всех вышеупомянутых случаях не будет обновлен незамедлительно.

Также:

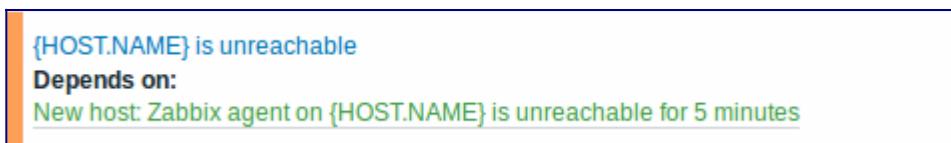
- Зависимость триггера может быть добавлена от любого триггера узла сети к триггеру любого другого узла сети, пока это не приведет к циклической зависимости.
- Зависимость триггера может быть добавлена от шаблона к шаблону. Если триггер из шаблона А зависит от триггера из шаблона В, то шаблон А может быть соединен с узлом сети (или с другим шаблоном) только вместе с шаблоном В, но шаблон В может быть соединен с узлом сети (или с другим шаблоном) в одиночку.
- Зависимость триггера может быть добавлена от шаблонного триггера к триггеру узла сети. В этом случае, соединение шаблона с узлом сети создаст триггер у узла сети, который будет зависеть от такого же триггера, что и шаблонный триггер. Это позволяет, например, иметь шаблон, в котором некоторые триггеры зависят от триггеров роутера (узла сети). Все узлы сети соединенные с этим шаблоном будут зависеть от этого конкретного роутера.
- Зависимость триггера узла сети от шаблонного триггера не может быть добавлена.
- Зависимость триггера может быть добавлена от прототипа триггера к другому прототипу триггера (для одного и того же правила низкоуровневого обнаружения) либо к реальному триггеру. Прототип триггера не может быть зависим от прототипа триггера другого правила обнаружения или от триггера созданного из прототипа триггера. Прототип триггера узла сети не может быть зависим от шаблонного триггера.

Настройка

Для того чтобы задать зависимость, откройте [диалог настройки](#) триггера. Далее нажмите на **Добавить** в 'Новая зависимость' и выберите один или несколько триггеров от которых будет зависеть наш триггер.



Нажмите **Обновить**. Теперь триггер имеет в списке индикацию на свою зависимость.



Примеры нескольких зависимостей

Например, узел сети Хост находится за маршрутизатором Роутер2, а Роутер2 находится за Роутер1.

Zabbix - Роутер1 - Роутер2 - Хост

Если Роутер1 недоступен, то очевидно, что и Хост и Роутер2 недоступны. И мы не хотим получать три оповещения с информацией о недоступности Хост, Роутер1 и Роутер2.

Для этого случая, мы зададим эти зависимости:

триггер 'Хост недоступен' зависит от триггера 'Роутер2 недоступен'
триггер 'Роутер2 недоступен' зависит от триггера 'Роутер1 недоступен'

Перед изменением состояния триггера 'Хост недоступен', Zabbix проверит зависимости у соответствующих триггеров. Если они будут найдены, и один из триггеров в находится в состоянии ПРОБЛЕМА, то состояние триггера не будет изменено и, следовательно, действия не будут выполнены и оповещения не будут отправлены.

Zabbix выполняет эту проверку рекурсивно. Если Роутер1 или Роутер2 недоступен, то триггер у Хост не будет изменен.

7.3.4 Важность триггеров

Важность триггера определяет насколько триггер важен. Zabbix поддерживает следующие важности триггеров:

ВАЖНОСТЬ	ОПРЕДЕЛЕНИЕ	ЦВЕТ
Не классифицировано	Неизвестная важность.	Серый
Информация	В информационных целях.	Светло-синий
Предупреждение	Предупреждающий.	Желтый
Средняя	Средняя проблема.	Оранжевый
Высокая	Произошло что-то важное.	Светло-красный
Чрезвычайная	Чрезвычайный. Финансовые потери и т.п.	Красный

Важности используются для:

- визуального представления триггеров. Различные цвета для различных уровней важности.
- аудио в глобальных оповещениях. Различные звуковые сигналы для различных важностей.
- оповещений пользователей. Различные способы оповещений (каналы оповещения) для различных важностей. Например, SMS – для высокой важности, email – для остального.
- ограничения действий по условиям важностей триггеров

Имеется возможность [настроить имена важностей триггеров и их цвета](#).

7.3.5 Пользовательские важности триггеров

Имена и цвета важностей триггеров связанные с важностью элементов веб-интерфейса можно настроить в *Администрирование* → *Общие* → *Важности триггеров*. Цвета являются общими для всех тем веб-интерфейса.

Перевод пользовательских имен важностей

Если используются переводы веб-интерфейса Zabbix, по умолчанию пользовательские имена важностей будут заменены переведенными именами.

Имена важностей триггеров по умолчанию доступны для перевода для всех локализаций. Если имя важности изменено, пользовательское имя будет использовано во всех локализациях и дополнительного потребуется ручной перевод.

Процедура перевода имени пользовательской важности:

- задайте требуемое имя пользовательской важности, например 'Важное'
- измените <папка_с_веб-интерфейсом>/locale/<требуемая_локаль>/LC_MESSAGES/frontend.po
- добавьте 2 строки:

```
msgid "Important"
msgstr "<переведенная строка>"
```

и сохраните файл

- создайте .mo файлы, как описано в <папка_с_веб-интерфейсом>/locale/README

Здесь **msgid** должно совпадать с новым именем пользовательской важности и **msgstr** должно являться переводом этой важности для указанного языка.

Эта процедура должна быть выполнена после каждого изменения имени важности.

7.3.6 Массовое обновление

Обзор

Используя массовое обновление, вы можете изменить любой параметр для нескольких триггеров за один раз, что спасет вас от необходимости открывать каждый отдельный триггер для изменения.

Использование массового обновления

Для массового обновления каких-либо триггеров, выполните следующее:

- Отметьте триггеры из списка для обновления
- Выберите *Массовое изменение* ниже списка
- Отметьте атрибуты для обновления
- Введите новые значения для этих атрибутов и нажмите на *Обновить*.

The screenshot shows the 'Mass Update' dialog box. At the top, there is a 'Severity' dropdown with options: Not classified, Information, Warning, Average (which is selected and highlighted in orange), High, and Disaster. Below this are two sections: 'Replace dependencies' and 'Replace tags'. Under 'Replace dependencies', there is a 'Name' field containing the text 'Zabbix server 1: Disk I/O is overloaded on {HOST.NAME}' and an 'Add' button. Under 'Replace tags', there is a 'tag' input field, a 'value' input field, and a 'Remove' button. Below these sections are 'Allow manual close' checkboxes (one checked, one unchecked) and 'No' and 'Yes' buttons. At the bottom are 'Update' and 'Cancel' buttons.

Замена зависимостей заменит существующие зависимости (если они есть) на новые, указанные в массовом обновлении.

7.3.7 Прогнозирующие функции триггеров

Обзор

Иногда имеются признаки надвигающейся проблемы. Эти признаки можно заметить, поэтому можно заблаговременно предпринять меры, чтобы предотвратить или хотя бы уменьшить влияние проблемы.

В Zabbix имеются средства прогнозирования будущего поведения наблюдаемой системы на основе исторических данных. Эти средства реализованы в виде прогнозирующих функций триггеров.

7.3.7.1 Функции

Необходимо знать две вещи, а именно то, как описать состояние проблемы и сколько времени нужно, чтобы предпринять меры. Далее есть два способа создать триггер, сигнализирующий о возможной нежелательной ситуации. Первый: триггер должен “загореться”, когда система после “пора действовать”, как ожидается, будет в состоянии проблемы. Второй: триггер должен “загореться”, если система перейдет в состояние проблемы за время меньше, чем “пора действовать”. Необходимо воспользоваться функциями триггеров, а именно **forecast** и **timeleft** соответственно. Надо заметить, что лежащий в основе статистический анализ, в основном идентичен у этих двух функций. Вы можете настроить триггер в зависимости от того, какой путь более предпочтителен для вас, и результаты будут идентичны.

7.3.7.2 Параметры

Обе функции используют почти одинаковый набор параметров. Обратитесь к списку [поддерживаемых функций](#) за справкой.

2.1 Интервал времени

Прежде всего вам необходимо указать период истории, который Zabbix должен проанализировать для составления прогноза. Сделайте это привычным способом при помощи параметров `sec` или `#num` и необязательного `сдвиг_времени` по аналогии с функциями `avg`, `count`, `delta`, `max`, `min` и `sum`.

2.2 Горизонт предсказания

(Только `forecast`)

Параметр `время` определяет, насколько далеко в будущее Zabbix будет экстраполировать зависимости, которые ему удастся найти в исторических данных. Независимо о того, используете ли вы `сдвиг_времени` или нет, время всегда отсчитывается от текущего момента.

2.3 Достигаемый порог

(Только `timelimit`)

Параметр `порог` определяет значение, которого должен достичь анализируемый элемент данных, нет никакой разницы, сверху или снизу. После того, как мы определили $f(t)$ (смотри ниже), мы должны решить уравнение $f(t) = \text{порог}$ и вернуть ближайший к текущему моменту корень, который находится справа от текущего момента, или вернуть 999999999999.9999, если таковых корней нет.

Когда значения элемента данных приближаются к порогу, а затем пересекают его, `timelimit` делает вывод, что пересечение уже находится в прошлом, и поэтому переключается на следующее пересечение с уровнем порога, если таковое имеется. Наилучшим решением является использование прогнозирований наряду с обычными диагностиками проблем, а не заменой одним на другое.¹¹

2.4 Функции аппроксимации

По умолчанию аппроксимация является линейной (*linear*) функцией. Если наблюдаемая система более сложная, вы можете выбрать один из следующих вариантов.

аппроксимация

$$x = f(t)$$

линейная (*linear*)

$$x = a + b*t$$

полином (*polynomial*)²

$$x = a_0 + a_1*t + a_2*t^2 + \dots + a_n*t^n$$

экспоненциальная (*exponential*)

$$x = a * \exp(b*t)$$

логарифмическая (*logarithmic*)

$$x = a + b * \log(t)$$

степенная (*power*)

$$x = a*t^b$$

2.5 Режимы

(Только `forecast`)

Каждый раз, когда вычисляется функция триггера, данные запрашиваются из указанного периода истории и по полученным данным строится указанная аппроксимация. Поэтому, если данные немного изменятся, то и построенная аппроксимация немного изменится. Если мы будем просто рассчитывать значение аппроксимирующей функции в заданный момент времени в будущем, то вы ничего не будете знать о том, как согласно прогнозу будет меняться анализируемый элемент данных между текущим моментом и этим моментом в будущем. При некоторых параметрах аппроксимации (вроде *polynomial*) просто лишь одно значение из будущего может ввести в заблуждение.

режим

результат `forecast`

значение (*value*) $f(\text{сейчас} + \text{время})$

максимум (*max*) $\max_{\text{сейчас} \leq t \leq \text{сейчас} + \text{время}} f(t)$

режим	результат forecast
минимум (min)	$\min_{\text{сейчас} \leq t \leq \text{сейчас} + \text{время}} f(t)$
дельта (delta)	$\max - \min$
среднее (avg)	среднее значение $f(t)$ ($\text{сейчас} \leq t \leq \text{сейчас} + \text{время}$) в соответствии с определением

7.3.7.3 Подробности

Для того, чтобы избежать вычислений с большими числами, мы рассматриваем штамп времени первого значения в указанном периоде плюс 1 наносекунда как новую точку отчёта времени (текущие штампы времени порядка 10^9 , в квадрате уже 10^{18} , а точность дробных значений около 10^{-16}). 1 нс прибавляется для того, чтобы все значения времени были положительными, поскольку построение логарифмической и степенной аппроксимаций подразумевает вычисление $\log(t)$. Этот сдвиг времени не влияет на линейную функцию, полином и экспоненциальную функции (за исключением более легких и более точных вычислений), но изменяет форму логарифмической и степенной функций.

7.3.7.4 Возможные ошибки

Функции возвращают -1 в следующих ситуациях:

- указанных период не содержит данных;
- математическая операция не задана^{[3\)](#)};
- сложности вычислений (к сожалению, для некоторых наборов входных данных диапазона и точности формата чисел с плавающей точкой двойной точности становится недостаточно)^{[4\)](#)}.

Никаких предупреждений или ошибок не выдаётся, если выбранная аппроксимирующая функция плохо описывает предоставленные данные или просто данных недостаточно для точного прогноза.

7.3.7.5 Примеры и обработка ошибок

Для получения предупреждения о том, что у вашего узла сети скоро закончится свободное дисковое пространство, вы можете использовать следующее выражение триггера:

```
{host:vfs.fs.size[/,free].timelimit(1h,,0)}<1h
```

Однако, может появиться код ошибки -1 и перевести ваш триггер в состояние проблемы. Вообще говоря, это не плохо, потому что вы получите предупреждение о том, что ваши прогнозирования не работает должным образом, и вам стоит обратить на них внимание, чтобы разобраться почему. Но иногда это плохо, потому что -1 может просто означать, что за последний час не было получено никаких данных о свободном дисковом пространстве данного узла сети. Если вы получаете много сообщений о ложных тревогах, подумайте об использовании более сложного выражения^{[5\)](#)} триггера:

```
{host:vfs.fs.size[/,free].timelimit(1h,,0)}<1h and
{host:vfs.fs.size[/,free].timelimit(1h,,0)}<>-1
```

Ситуация немного сложнее с **forecast**. Начнём с того, что -1 может перевести, а может и не перевести триггер в состояние проблемы в зависимости от вашего выражения триггера, которое может быть вроде

```
{host:item.forecast(...)}<...
```

или наподобие

```
{host:item.forecast(...)}>...
```

Более того, -1 может быть вполне корректным результатом прогнозирования, если прогнозируемый элемент данных может принимать отрицательные значения. Но вероятность возникновения такой ситуации в реальных условиях пренебрежительно мала (смотрите, [как](#) работает оператор =). Поэтому добавьте

```
... or {host:item.forecast(...)}=-1
```

или

```
... and {host:item.forecast(...)}<>-1
```

если вы соответственно хотите или, наоборот, не хотите рассматривать -1 как проблему.

Смотрите также

1. [Прогнозирующие функции триггеров \(pdf\)](#) on zabbix.org [en]

[1\)](#)

Например, простой триггер наподобие

```
{host:item.timeleft(1h,,X)} < 1h
```

может перейти в состояние проблемы, когда элемент данных приближается к X, и неожиданно “самоустраниться”, как только значение X достигнуто. Если проблема выражается в том, что значение элемента данных меньше X, используйте:

```
{host:item.last()} < X or {host:item.timeleft(1h,,X)} < 1h
```

Если проблема заключается в том, что значение элемента данных больше X, используйте:

```
{host:item.last()} > X or {host:item.timeleft(1h,,X)} < 1h
```

[2\)](#)

Степень многочлена может быть от 1 до 6, при этом функция *polynomial1* равнозначна *linear*. Однако, использовать полиномы высоких степеней следует [с осторожностью](#). Если период вычисления содержит меньше точек, чем требуется для определения коэффициентов полинома, степень полинома понизится (например, запрашивается *polynomial5*, но есть только 4 точки, поэтому для аппроксимации будет использована *polynomial3*).

[3\)](#)

Например, построение экспоненциальной и степенной функций требует логарифмических вычислений значений элемента данных. Если данные содержат нулевые или отрицательные числа, вы получите сообщение об ошибке, поскольку вычисление логарифма возможно только при положительных значениях.

[4\)](#)

При аппроксимации *linear*, *exponential*, *logarithmic* и *power* выражений все необходимые вычисления можно написать в явном виде. При *polynomial* можно вычислить только *value* без дополнительных шагов. Вычисление *avg* включает в себя вычисление первообразной полинома (аналитически). Вычисление *max*, *min* и *delta* включает в себя вычисление производной полинома (аналитически) и поиск его корней (численно). Решение $f(t) = 0$ требует нахождения корней полинома (численно).

[5\)](#)

Но в этом случае -1 может восстановить триггер из состояния проблемы. Для полной защищённости используйте:

```
{host:vfs.fs.size[/,free].timeleft(1h,,0)}<1h and ({TRIGGER.VALUE}=0 and {host:vfs.fs.size[/,free].timeleft(1h,,0)}<>-1 or {TRIGGER.VALUE}=1)
```

7.4 События

Обзор

События в Zabbix генерируются несколькими источниками:

- события на триггеры - всякий раз, когда триггер меняет свое состояние(OK → ПРОБЛЕМА → OK)
- события на обнаружение - при обнаружении узлов сети или сервисов
- события на авторегистрацию - когда активные агенты автоматически регистрируются сервером
- внутренние события - когда элементы данных/правила низкоуровневого обнаружения становятся не поддерживаемыми или триггер переходит в состояние неизвестно

Внутренние события поддерживаются начиная с Zabbix 2.2.

События имеют штамп времени и могут быть основой для действий, таких как отправка оповещений по почте и т.п.

Для просмотра деталей событий в веб-интерфейсе, перейдите в *Мониторинг* → *События*. Здесь вы можете нажать на дату и время события и просмотреть его детали.

Более подробная информация доступна в:

- [события на триггеры](#)
- [другие источники событий](#)

7.4.1 Генерация событий на триггеры

Обзор

Изменение состояния триггера является наиболее частым и наиболее важным источником событий. Каждый раз, когда триггер меняет свое состояние, генерируется событие. Событие содержит подробную информацию о изменении состояния триггера - когда это случилось, и какое сейчас новое состояние.

Триггерами создаются два типа событий - Проблема и OK.

События проблемы

Событие проблемы создается:

- когда выражение триггера вычисляется как ПРАВДА, если триггер в состоянии OK;
- каждый раз выражение триггера вычисляется как ПРАВДА, если у триггера активировано множественная генерация событий проблема.

События OK

Событие OK закрывает связанные событие(ия) о проблеме и может быть создано тремя компонентами:

- триггеры - основываясь на настройках 'Генерация OK событий' и 'OK событие закрывает';
- корреляция событий
- менеджер задач – когда событие [закрывается вручную](#)

Триггеры

У триггеров имеется настройка 'Формирование OK событий', которая определяет, как генерируются OK события:

- *Выражение* - событие OK генерируется триггеру в состоянии проблемы, когда выражение этого триггера вычисляется как ЛОЖЬ. Это самая простая настройка, активированная по умолчанию.
- *Выражение восстановления* - событие OK генерируется триггеру в состоянии проблемы, когда выражение этого триггера вычисляется как ЛОЖЬ и выражение восстановления вычисляется как ПРАВДА. Этую опцию можно использовать, если критерий восстановления триггера отличается от критерия проблемы.
- *Нет* - событие OK никогда не генерируется. Этую опцию можно использовать в сочетании с множественной генерацией проблем для простой отправки оповещений, когда что-то происходит.

Дополнительно, у триггеров имеется настройка 'OK событие закрывает', которая определяет какие события о проблемах закрываются:

- *Все проблемы* - событие OK закрывает 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. Этую опцию часто называют корреляция событий на уровне триггеров.

Корреляция событий

Корреляция событий (также называемая глобальной корреляцией событий) является способом настройки пользовательских правил закрытия (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 добавить условие на основе старого события, если используется операция 'закрыть старые события', или все существующие проблемы могут быть закрыты
3. избегайте использования распространенных имен тегов, которые используются различными настройками корреляции

Менеджер задач

Если у триггера активирована настройка 'Разрешить закрывать вручную', тогда появляется возможность закрывать события о проблемах, которые созданы этим триггером, вручную. Это можно сделать в веб-интерфейсе при [обновлении проблемы](#). Событие не закрывается напрямую – вместо 'закрытия события' создается задача, которая будет обработана менеджером задач в самое короткое время. Диспетчер задач создаст соответствующее OK событие и событие о проблеме закроется.

7.4.2 Другие источники событий

События на обнаружения

Zabbix периодически сканирует диапазоны IP адресов заданные в правилах сетевого обнаружения. Частота этой проверки настраивается индивидуально для каждого правила. 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 на авторегистрацию активных агентов

Авторегистрация активных агентов создает события в Zabbix.

Если настроено, событие на авторегистрацию активного агента может создаваться, если ранее неизвестный активный агент запрашивает свои проверки или, если изменились метаданные узла сети. Сервер добавляет новый автоматически зарегистрированный узел сети, используя полученные IP адрес и порт от агента.

Для получения более подробной информации, смотрите страницу об [автоматической регистрации активных агентов](#).

Внутренние события

Внутренние события возникают, когда:

- элемент данных меняет свое состояние с “нормального” на “неподдерживается”
- элемент данных меняет свое состояние с “неподдерживается” на “нормальное”
- правило низкоуровневого обнаружения меняет свое состояние с “нормального” на “неподдерживается”
- правило низкоуровневого обнаружения меняет свое состояние с “неподдерживается” на “нормальный”
- триггер меняет свое состояние с “нормального” на “неизвестное”
- триггер меняет свое состояние с “неизвестного” на “нормальное”

Внутренние события поддерживаются начиная с Zabbix 2.2. Целью введения внутренних событий - дать знать пользователям, когда происходят некоторые внутренние события, например, элемент данных становится неподдерживаемым и перестает собирать данные.

7.4.3 Закрытие проблем вручную

Обзор

Хотя в целом события о проблемах решаются автоматически, когда состояние триггера переходит из 'Проблема' в 'OK', могут быть случаи когда трудно определить при помощи выражения триггера что проблема решена. В таких случаях проблема должна быть решена вручную.

Например, *syslog* может сообщать о том, что для оптимальной производительности необходимо отрегулировать некоторые параметры ядра. В это случае информация о проблеме сообщается Linux администраторам, они исправляют её и затем вручную закрывают эту проблему.

Проблемы можно закрывать вручную только у триггеров, у которых включена опция *Разрешить закрывать вручную*.

Когда проблема “закрыта вручную”, Zabbix генерирует новую внутреннюю задачу для Zabbix сервера. Затем процесс *task manager* выполняет эту задачу и генерирует OK событие, следовательно закрывает событие о проблеме.

Вручную закрытая проблема не означает, что основной триггер никогда больше не перейдет в состояние 'Проблема'. При поступлении новых данных от любого элемента данных, которые включены в выражение триггера, произойдет повторное вычисление всего выражения, что может привести снова к проблеме. Также триггер вычисляется повторно, когда в его выражении используются функции связанные со временем. Полный список функций связанных со временем можно найти на [странице с Триггерами](#).

Настройка

Необходимо выполнить два шага для закрытия проблем вручную.

Настройка триггера

В настройке триггера включите опцию *Разрешить закрывать вручную*.



Экран обновления проблемы

Если возникает проблема у триггера с флагом *Закрыть вручную*, вы можете перейти на экран [обновления проблемы](#) этого триггера и закрыть проблему вручную.

Чтобы закрыть проблему, проверьте наличие опции *Закрыть проблему* в диалоге и нажмите на *Обновить*.

Update problem

Message	Fixed, closing.
History	Time User User action Message
Scope	<input checked="" type="radio"/> Only selected problem <input type="radio"/> Selected and all other problems of related triggers 1 event
Change severity	<input type="checkbox"/> Not classified Information Warning Average High
Acknowledge	<input type="checkbox"/>
Close problem	<input checked="" type="checkbox"/>
* At least one update operation or message must exist.	
<input type="button" value="Update"/> <input type="button" value="Cancel"/>	

Все обязательные поля ввода отмечены красной звёздочкой.

Запрос выполняется Zabbix сервером. Обычно такая процедура занимает несколько секунд до закрытия проблемы. В процессе этой процедуры в *Мониторинг → Проблемы* будет отображаться **ЗАКРЫТИЕ** как состояние проблемы.

Проверка

Можно проверить что проблема была закрыты вручную:

- в деталях события, доступных через *Мониторинг → Проблемы*;
- используя макрос {EVENT.UPDATE.HISTORY} в сообщениях оповещения, который предоставит такую информацию.

7.5 Корреляция событий

Обзор

Корреляция событий позволяет сопоставлять события о проблемах с их решениями очень точным и гибким способом.

Корреляцию событий можно определять:

- на уровне триггера - один триггер можно использовать для связки отдельных проблем с их решением
- глобально - проблемы можно соотнести с их решением из другого триггера/метода опроса, используя глобальные правила корреляции

7.5.1 Корреляция событий на основе триггеров

Обзор

Корреляция событий на основе триггеров позволяет сопоставлять отдельные проблемы, о которых сообщает один триггер.

В то время как ОК события закрывают в Zabbix все события о проблеме, бывают случаи, когда необходим более обстоятельный подход. Например, при мониторинге файлов журналов вы можете захотеть обнаруживать некоторые проблемы в файле журнала и закрывать их по отдельности, а не все разом.

Это тот случай, когда триггеры имеют активированную опцию *Формирование множественных Проблема событий*. Такие триггеры обычно используются для мониторинга журналов, обработки трапов и т.п.

В Zabbix имеется возможность сопоставить события о проблемах, основываясь на [тегах событий](#). Теги используются для извлечения значений и создания метки по событиям о проблеме. Используя преимущества такого подхода, проблемы могут быть закрыты отдельно на основании совпадения тега.

Другими словами, один триггер может создавать отдельные события, которые идентифицируются при помощи тега событий. Следовательно, события о проблемах могут быть выявлены одно за другим и закрыты отдельно на основании метки при помощи тега событий.

Как это работает

В мониторинге журналов вы можете встретиться со строками как эти:

Строка1: Приложение 1 остановлено
Строка2: Приложение 2 остановлено
Строка3: Приложение 1 перезапущено
Строка4: Приложение 2 перезапущено

Идея корреляции событий состоит в том, чтобы была возможность сопоставить событие о проблеме со Строки1 с решением со Строки3 и событие о проблеме со Строки2 с решением со Строки4 и закрыть эти проблемы одну за другой:

Строка1: Приложение 1 остановлено
Строка3: Приложение 1 перезапущено #проблема со Строки 1 закрыта

Строка2: Приложение 2 остановлено
Строка4: Приложение 2 перезапущено #проблема со Строки 2 закрыта

Чтобы сделать подобное вам необходимо отметить эти связанные события как, например, "Приложение 1" и "Приложение 2". Это можно сделать применив регулярное выражение к строке из файла журнала, чтобы извлечь значение тега. Затем, когда события будут созданы, они будут промаркованы как "Приложение 1" и "Приложение 2" соответственно и проблема может быть сопоставлена решению.

Настройка

Для настройки корреляции на уровне триггера:

- перейдите в [диалог настройки](#) триггера

Trigger **Dependencies**

* Name	Service {{ITEM.VALUE}}.regsub("^.* service ([a-zA-Z]*).*\$", "1") stopped							
Severity	Not classified	Information	Warning	Average	High	Disaster		
* Expression	{{Template Services:log[/var/log/messages].regexp("stopped")}}=1							
Expression constructor								
OK event generation	Expression	Recovery expression	None					
Recovery expression	{{Template Services:log[/var/log/messages].regexp("started")}}=1							
Expression constructor								
PROBLEM event generation mode	Single	Multiple						
OK event closes	All problems	All problems if tag values match						
Tag for matching	Service							
Tags	Service		{{ITEM.VALUE}}.regsub("^.* service ([a-zA-Z]*).*\$", "1")		Remove			
	Datacenter		value		Remove			
	Add							

Все обязательные поля ввода отмечены красной звёздочкой.

- выберите 'Режим формирования Проблема событий' значение **Множественный**
- в 'OK событие закрывает' выберите *Все проблемы, если значения тегов совпадают*
- введите имя тега для сопоставления событий
- настройте [теги](#) для извлечения значений тега со строк журнала

При успешной настройке у вас будет возможность видеть события о проблемах промаркированные приложением и сопоставленные с их решением в *Мониторинг → Проблемы*.

Problems										Export to CSV	Print
Time	Severity	Recovery time	Status	Info	Host	Problem	Duration	Ack	Actions	Tags	
08:38:18	High	08:38:18	RESOLVED	Zabbix server	Service Apache	Apache stopped	0	No		Service: Apache Webserver	Edit

Посколько возможна некорректная настройка, когда похожие теги событий могут быть созданы по **нерешаемым** проблемам, пожалуйста, рассмотрите указанные ниже подобные случаи!

- При наличии двух приложений, которые записывают сообщениях об ошибках и восстановлениях в один и тот же файл журнала, пользователь может решить использование двух *Приложение* тегов в одном триггере с различными значениями тегов, используя разные регулярные выражения в значениях тегов для извлечения имен, так сказать приложение А и приложение В из {{ITEM.VALUE}} макроса (например, если форматы сообщений различаются). Однако, такой подход может не сработать как задумано, если регулярным выражениям будут отсутствовать соответствия. Несоответствующие регулярные выражения могут выдать пустые значения тегов и одного пустого значения тега как по событиям о проблеме, так и по OK событиям, будет достаточно чтобы их соотнести. Таким образом сообщение о восстановлении с приложения А может случайно закрыть сообщение об ошибке с приложения В.
- Реальные теги и значения тегов становятся видимы только когда срабатывает триггер. Если используемое регулярное выражение ошибочно, оно автоматически заменяется на *НЕИЗВЕСТНО* строку. Если изначальное событие о проблеме со значением тега

НЕИЗВЕСТНО отсутствует, тогда могут появиться последующие OK события с тем же значением тега *НЕИЗВЕСТНО*, которые могут закрыть события о проблемах, которые они не должны закрывать.

- Если пользователь использует {ITEM.VALUE} макрос без функций макроса в качестве значения тега, тогда будет применяться ограничение по длине строки в 255-символ. Когда в журнале имеются длинные сообщения и первые 255 символов не конкретизируют проблему, может привести к идентичным тегам событий по нерешаемым проблемам.

Теги событий

Обзор

Существует возможность задать в Zabbix пользовательские теги событий. Теги событий задаются на уровне триггера. После того как теги добавлены, соответствующие новые события маркируются данными тегами.

Наличие пользовательских тегов событий даёт большую гибкость. Наиболее важно, что события можно [скоррелировать](#), основываясь на тегах событий. В качестве другого применения, действия можно назначать на основе тегов событий.

Теги событий реализованы парой *имя тега и значение*. Вы можете использовать только имя или пару имени со значением:

MySQL, Сервис:MySQL, Сервисы, Сервисы:Клиент, Приложения, Приложение: Java,
Приоритет:Высокий

Сценарии применения

Некоторые сценарии использования этой функции заключаются в следующем:

1. Распознавание проблем в файле журнала и раздельное их закрытие
 - Добавьте теги в триггер к журналу, которые будут распознавать события, используя значение извлеченное при помощи макроса {{ITEM.VALUE<N>}.regsub()}; * В настройках триггера укажите режим формирования множественных проблем события; * В настройках триггера используйте [[:ru/manual/config/event_correlation|корреляцию событий]]: выберите опцию, которая закроет OK события только в случае совпадения событий и укажите тег поиска совпадений; * Просматривайте события о проблемах созданные с тегом и закрытые по отдельности. - Использование тегов для фильтрации оповещений * Добавьте теги на уровне триггера, чтобы промаркировать события при помощи различных тегов; * Используйте фильтрацию по тегу в условиях действия чтобы получать оповещения только по тем событиям, которые совпадают с данными тега. - Просмотр информации по тегу в веб-интерфейсе * Добавьте теги на уровне триггера, чтобы промаркировать события при помощи различных тегов; * Просматривайте эту информацию в //Мониторинг// -> //Проблемы//. - Использование информации извлеченной из значения элемента данных значением тега * Используйте макрос {{ITEM.VALUE<N>}.regsub()} в значении тега; * Просматривайте значения тегов в //Мониторинг// -> //Проблемы// как извлеченные данные из значения элемента данных. - Лучшее распознавание проблем в оповещениях * Задайте теги на уровне триггера; * Используйте макрос {EVENT.TAGS} в оповещениях на проблемы; * В итоге более простое распознавание о каких приложениях/сервисах идет речь в оповещении. - Упрощение задач настройки при использовании тегов на уровне шаблона * Задайте теги на уровне шаблонного триггера; * Просматривайте эти теги у всех триггеров созданных из шаблонных триггеров. - Создание триггеров с тегами из низкоуровневого обнаружения (LLD) * Задайте теги прототипам триггеров; * Используйте LLD макросы в имени тега или значении; * Просматривайте эти теги у всех триггеров созданных из

прототипов триггеров. === Настройка === Теги событий задаются при настройке триггера. Теги событий можно задать для триггеров, шаблонных триггеров и прототипов триггеров. {{:manual:config:triggers:event_tags.png}} === Поддержка макросов === В тегах событий можно использовать следующие макросы: * Макросы {ITEM.VALUE}, {ITEM.LASTVALUE}, {HOST.HOST}, {HOST.NAME}, {HOST.CONN}, {HOST.DNS}, {HOST.IP}, {HOST.PORT} и {HOST.ID} можно использовать, чтобы заполнять имя тега или значение тега. * [[:ru/manual/appendix/macros/supported_by_location|Макросы]] {INVENTORY.*} можно использовать в качестве ссылки на значения инвентарных данных с одного или нескольких узлов сети в выражении триггера (поддерживается начиная с 4.0.0). * В именах тегов и значениях тегов поддерживаются [[:ru/manual/config/macros/usermacros|пользовательские макросы]], а также контекст пользовательских макросов. Контекст пользовательских макросов может включать в себя макросы низкоуровневого обнаружения. * Можно использовать макросы низкоуровневого обнаружения в имени тега и значении тега в прототипах триггеров. Можно использовать макросы {EVENT.TAGS} и {EVENT.RECOVERY.TAGS} в оповещениях на основе триггеров и они будут раскрыты в разделенный запятыми список тегов события или тегов события восстановления. == Извлечение подстроки == Поддерживается извлечение подстроки для заполнения имени тега или значения тега при использовании новой [[:ru/manual/config/macros/macro_functions|функции макросов]] - применение регулярного выражения к значению полученному при помощи {ITEM.VALUE}, {ITEM.LASTVALUE} макросов, а также макросов низкоуровневого обнаружения. Например: {{ITEM.VALUE}.regsub(шаблон, вывод)} {{ITEM.VALUE}.iregsub(шаблон, вывод)} {{#LLDMAKPOS}.regsub(шаблон, вывод)} {{#LLDMAKPOS}.iregsub(шаблон, вывод)} Имя тега и значение будут обрезаны до 255 символов, если их длина превысит 255 символов после раскрытия макросов. Смотрите также: Использование функций макросов в [[:ru/manual/config/macros/lld_macros#использование_функций_макросов|макросах низкоуровневого обнаружения]] для тегирования событий. === Просмотр тегов события === Теги событий, если заданы, можно просмотреть с новыми событиями в: * //Мониторинг// -> //Проблемы// * //Мониторинг// -> //Проблемы// -> //Детали события// * //Мониторинг// -> //Панель// -> виджет //Проблемы// (во всплывающем окне, которое открывается при наведении курсора мыши на имя проблемы) {{:manual:config:triggers:event_tags_view.png?600}} Отображаются только первые три записи тегов. Если имеется более трех записей тегов, такой факт будет обозначен тремя точками. Если вы проведете курсором мыши над этими тремя точками, во всплывающем окне отобразятся все записи тегов. Обратите внимание, что на порядок в котором отображаются теги влияет фильтрация тегов и опция //Приоритет отображения тегов// в фильтре //Мониторинг// -> //Проблемы// или в виджете панели //Проблемы/. === Влияние на производительность и хранение === Ожидается, что использование тегов событий может иметь следующие последствия: * Обработка событий будет происходить медленнее по причине создания новых записей в таблице тегов событий. * Операции связанные с манипуляциями с триггерами унаследованными из шаблонов будут медленнее так как имеется необходимость создания по одной записи на каждый тег в таблице тегов триггеров. По этой причине создание, обновление и удаление триггеров будет более медленным. * Синхронизация кэша конфигурации будет медленнее по причине наличия тегов триггеров. * Хранение тегов событий потребует дополнительное пространство на диске, которое может быть сравнимо с размером существующих таблиц триггеров и событий. Конкретные требования к хранению очень сильно зависят от количества тегов созданных по каждому триggerу и событию.

7.5.2 Глобальная корреляция событий

Обзор

Глобальная корреляция событий позволяют охватить все метрики наблюдаемые Zabbix и создавать корреляции.

Имеется возможность сопоставления событий, которые были созданы совершенно разными триггерами, и применения одних и тех же операций к ним ко всем. Созданием интеллектуальные правила корреляции можно обезопасить себя от тысяч повторяющихся оповещений и сфокусироваться на реальных причинах проблемы!

Глобальная корреляция событий - мощный механизм, который позволяет вам освободить себя от логики проблемы и решения основанных на одном триггере. До сих пор одно событие о проблеме создавалось одним триггером мы зависели от этого же триггера в плане решения проблемы. Мы не могли решить проблему, которая была создана одним триггером, при помощи другого триггера. Но с корреляцией событий на основе меток событий, мы можем.

Например, триггер на журнал может сообщать о проблемах с приложением в то время как, триггер на основе опроса может сообщать, что приложение запущено и работает. Используя теги событий вы можете пометить триггер на журнал как *Состояние: Недоступен*, тогда как меткой триггера на основе опроса будет *Состояние: Доступен*. Затем, в глобальном правиле корреляции вы можете соотнести эти триггеры и назначить соответствующую операцию на такую корреляцию, такую как закрытие старых событий.

В другом случае глобальная корреляция может распознавать похожие триггеры и применять к ним ту же операцию. Что если мы можем получить только одно сообщение о проблемах по каждому сетевому порту? Не требуется сообщать по ним всем. Это также возможно при помощи глобальной корреляции событий.

Глобальная корреляция событий настраивается в **правилах корреляции**. Правило корреляции определяет каким образом новые события о проблемах соотносятся с существующими событиями о проблемах и что делать в случае совпадения (закрыть новое событие, закрыть старые события, сформировав соответствующие OK события). Если проблема закрыта при помощи глобальной корреляции, об этом сообщается в *Инфо* колонке в *Мониторинг → Проблемы*.

Настройка глобальных правил корреляции доступна пользователям только с Zabbix Супер Администратор уровнем.

Корреляцию событий необходимо настраивать очень осторожно, так как некорректная настройка может негативно сказаться на производительности обработки событий или может привести к закрытию большего количества событий чем предполагается (в худшем случае все события о проблемах могут быть закрыты).

Для безопасной настройки глобальной корреляции обратите внимание на следующие важные советы:

- Уменьшайте масштаб корреляции. Всегда указывайте уникальный тег для нового события, который соотносится со старыми событиями и используйте *Тег нового события* условие корреляции;
- Добавляйте условие, которое основывается на старом событии, при использовании *Закрыть старое событие* операции (в противном случае все существующие проблемы могут быть закрыты);
- Избегайте использования общих имен тегов, которые в итоге могут быть использованы различными конфигурациями корреляции;
- Сведите количество правил корреляции к действительно необходимым вам.

Смотрите также: [известные проблемы](#).

Настройка

Для настройки глобальных правил корреляции событий:

- Перейдите в *Настройка* → *Корреляция событий*
- Нажмите на *Создать корреляцию* справа (или на имя корреляции, чтобы изменить уже существующее правило)
- В диалоге введите параметры правила корреляции

The screenshot shows the 'Correlation' configuration dialog. At the top, there are tabs for 'Correlation' and 'Operations'. Below them, the 'Name' is set to 'Close old event'. The 'Type of calculation' is set to 'And' (with the expression 'A and (B and C) and D'). There is a section for 'Conditions' with four entries: A (Old event tag Application equals new event tag Application), B (Old event tag Application equals ABC), C (Old event tag State equals Down), and D (New event tag State equals Up). Each entry has a 'Remove' link next to it. Below this, there is a 'New condition' input field with dropdowns for 'New event tag value' (set to 'tag'), 'equals', and 'value'. An 'Add' button is available. A 'Description' text area contains the text: 'Close old events for Application ABC if an event with State=Up happens.'. At the bottom, there is an 'Enabled' checkbox which is checked, and two buttons: 'Add' and 'Cancel'.

Все обязательные поля ввода отмечены красной звёздочкой.

Параметр	Описание
Имя	Уникальное имя правила корреляции. Для вычисления условий доступны следующие опции: И - должны быть выполнены все условия Или - достаточно выполнения одного условия
Тип вычисления	И/Или - И с разными типами условий и ИЛИ с одинаковым типом условий Пользовательское выражение - формула вычисления, введенная пользователем, для оценки условий действия. Она должна включать в себя все условия (представленные в виде прописных букв А, В, С, ...) и может включать пробелы, символы табуляции, скобки (), and (с учетом регистра), or (с учетом регистра), not (регистрозависимое).
Условия	Список условий, выбранных из <i>Новое условие</i> поля. Выберите условия при которых события будут соотноситься и нажмите на <i>Добавить</i> . Доступны следующие условия: Тег старого события - совпадение нового события со старым событием(ями), которые соответствуют тегу старого события Тег нового события - совпадение нового события, которое соответствует тегу события, старому событию(ям) Группа узлов сети нового события - совпадение нового события, которое принадлежит соответствующей группе узлов сети, старому событию(ям) Пара тегов событий - совпадение нового события старому событию(ям), если значения указанных тегов совпадают в обоих событиях. <i>Имена</i> тегов не должны совпадать. Эта опция может быть полезна для сопоставления значений в реальном времени,
Новое условие	

Параметр

Описание

которые могут быть неизвестны в процессе настройки (смотрите также [Пример 1](#))
Значение тега старого события - совпадение нового события старому событию(ям), которое:

- = - имеет соответствующее значение тега старого события
- <> - не имеет соответствующее значение тега старого события

содержит - имеет соответствующую строку в значении тега старого события
не содержит - не имеет соответствующую строку в значении тега старого события
Значение тега нового события - совпадение нового события старому событию(ям), если новое событие:

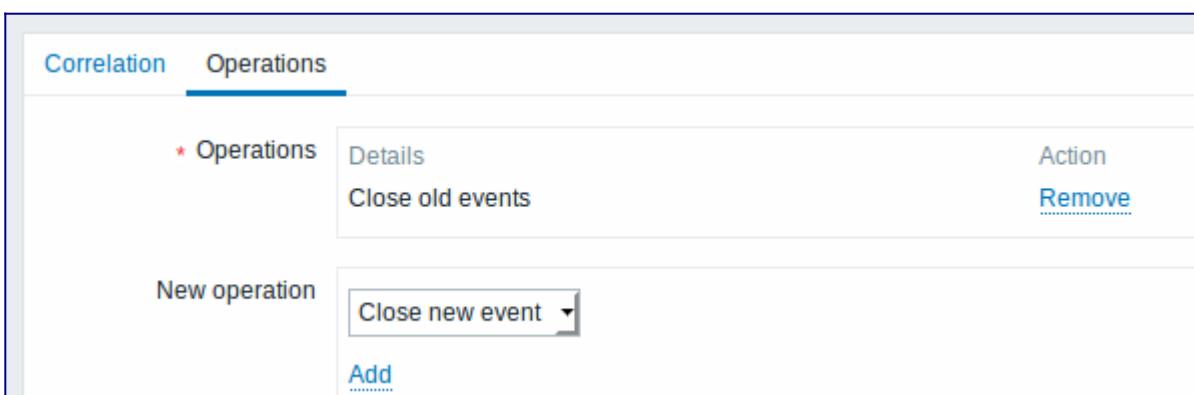
- = - имеет соответствующее значение тега нового события
- <> - не имеет соответствующее значение тега нового события

содержит - имеет соответствующую строку в значении тега нового события
не содержит - не имеет соответствующую строку в значении тега нового события

Описание Описание правила корреляции.

Активировано Если вы выберите этот параметр, правило корреляции будет активировано.

- В диалоге выберите операцию правила корреляции в диалоге



Параметр

Описание

Операции Список операций, выбранных из *Новая операция* поля.

Выберите выполняемую операцию при соответствии события и нажмите на *Добавить*. Доступны следующие операции:

Новая операция **Закрыть старые события** - закрыть старые события при появлении нового события. Всегда добавляйте условие на основе старого события при использовании **Закрыть старые события** операции или будут закрыты все существующие проблемы.
Закрыть новое событие - закрыть новое событие при его появлении

Посколько возможна некорректная настройка, когда похожие теги событий могут быть созданы по **нерешаемым** проблемам, пожалуйста, рассмотрите указанные ниже подобные случаи!

- Реальные теги и значения тегов становятся видимы только когда срабатывает триггер. Если используемое регулярное выражение ошибочно, оно автоматически заменяется на ***НЕИЗВЕСТНО*** строку. Если изначальное событие о проблеме со значением тега ***НЕИЗВЕСТНО*** отсутствует, тогда могут появиться последующие OK события с тем же значением тега ***НЕИЗВЕСТНО***, которые могут закрыть события о проблемах, которые они не должны закрывать.
- Если пользователь использует **{ITEM.VALUE}** макрос без функций макроса в качестве значения тега, тогда будет применяться ограничение по длине строки в 255-символ. Когда в журнале имеются длинные сообщения и первые 255 символов не конкретизируют проблему, может привести к идентичным тегам событий по нерешаемым проблемам.

Примеры

Пример 1

Остановка повторяющихся событий о проблемах с одного и того же сетевого порта.

Correlation **Operations**

* Name	Correlate network port problems			
Type of calculation	And	A and B		
* Conditions	Label	Name	Action	
	A	Old event tag Port = new event tag Port	Remove	
	B	Old event tag Host = new event tag Host	Remove	
New condition	Event tag pair	old event tag	=	new event tag
	Add			
Description	Keep only one problem per port. No need to report all of them.			
Enabled	<input checked="" type="checkbox"/>			
	Add	Cancel		

Такое глобальное правило корреляции сопоставит проблемы, если у триггера существуют Узел сети и Порт значения тегов и они такие же как у оригинального события, так и у нового события.

Correlation **Operations**

* Operations	Details	Action
	Close new event	Remove

Эта операция закроет все события о проблеме по тому же самому сетевому порту, оставив открытой только оригинальную проблему.

7.6 Визуализация

7.6.1 Графики

Обзор

При наличии большого потока данных поступающего в Zabbix данных, намного проще для пользователей, если они смогут видеть визуальное представление о происходящем, а не только числа.

Именно в этом случае приходят графики. Графики позволяют понять поток данных просто взглянув на него, соотнести проблемы, обнаружить когда что-то началось или выступить с докладом о том, что что-то может стать проблемой.

Zabbix предоставляет пользователям:

- встроенные [простые графики](#) по данным одного элемента данных
- возможность создавать более сложные [пользовательские графики](#)
- доступ к быстрому сравнению нескольких элементов данных, используя [ситуационные графики](#)

7.6.1.1 Простые графики

Обзор

Простые графики служат для визуализации данных собранных элементами данных.

Со стороны пользователя не требуются никакие усилия по настройке простых графиков для их просмотра. Они свободно доступны через Zabbix.

Просто перейдите в *Мониторинг → Последние данные* и нажмите на ссылку Графика напротив соответствующего элемента данных и график отобразится.



Простые графики доступны для всех числовых элементов данных. Для текстовых элементов данных доступна ссылка на Историю в *Мониторинг → Последние данные*.

Выбор периода времени

Обратите внимание на выбор периода времени над графиком. Он позволяет выбрать часто используемые периоды одним щелчком мыши.

Обратите внимание, что такие опции как *Сегодня*, *Эта неделя*, *Этот месяц*, *Этот год* отображают весь период целиком, включая часы/дни которые ещё не наступили (из будущего). *Сегодня до сих пор*, напротив, отображает только прошедшие часы.

Когда период выбран, его можно перемещать назад и вперед во времени нажимая на кнопки стрелок . Кнопка *Отдалить* позволяет увеличить период в два раза или на 50% с каждой стороны.

Выполнить увеличение также возможно, если дважды нажать в графиках. Можно свернуть весь селектор выбора времени, нажав на подпись к вкладке, которая содержит строку с выбранным периодом.

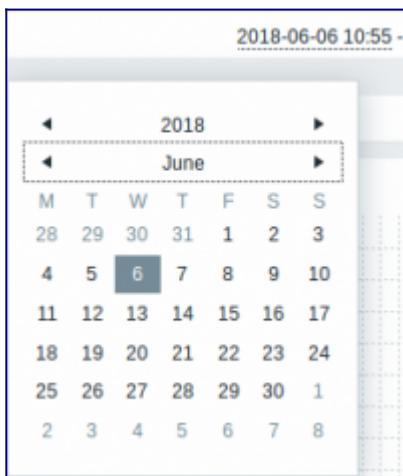
Поля *От/До* отображают выбранный период в виде:

- абсолютного времени в формате Г-М-Д Ч:М:С
- синтаксиса относительного времени, например: now - 1d

Дата в относительном формате может содержать только одну математическую операцию (- или +). В относительном времени поддерживаются следующие аббревиатуры:

- now
- s (секунды)
- m (минуты)
- h (часы)
- d (дни)
- w (недели)
- M (месяцы)
- y (года)

Имеется возможность выбрать конкретную дату начала/конца нажатием на иконке календаря после полей *От/До*. В этом случае откроется всплывающее окно выбора даты.



Внутри выбора даты имеется возможность перемещения между блоками года/месяца/даты используя Tab и Shift+Tab. Стрелки клавиатуры или кнопки стрелок позволяют выбрать желаемое значение. Нажатие Enter (или нажатие на желаемом значении) активирует выбор.

Другой путь контроля отображаемого времени - выделение области на графике левой кнопкой мыши. График будет приближен в выбранную область сразу как вы отпустите левую кнопку мыши.

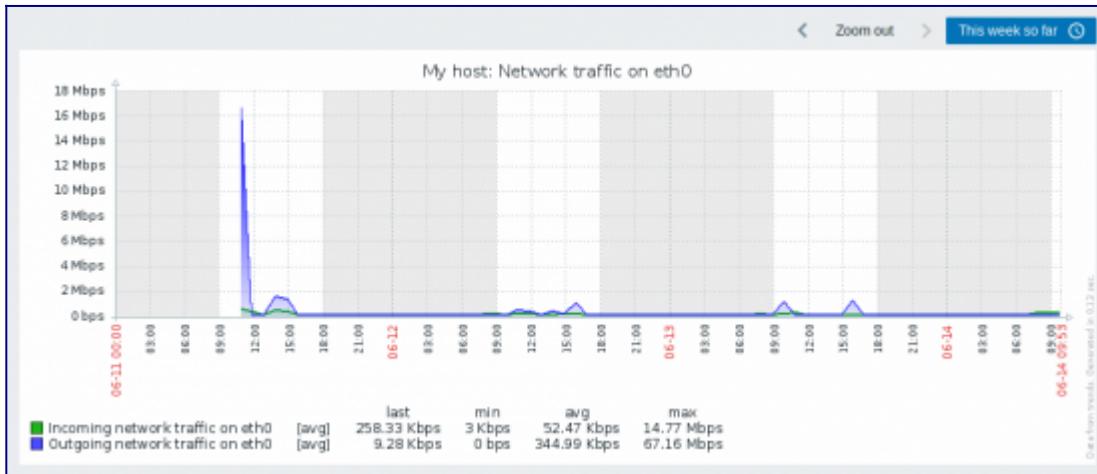
В случае, если значение времени не указано или поле пустое, тогда значение времени будет задано значением равным “00:00:00”. Такое поведение не применяется к выбору сегодняшней даты: в этом случае время будет задано текущим значением времени.

Недавние данные vs длительные периоды

Для самых последних данных **одиночная** линия рисуется с учетом каждого полученного значения. Одиночная линия рисуется до тех пор, пока доступен хотя бы один горизонтальный пиксель для одного значения.

Для данных показывающих длительный период рисуется **три линии** - темно зеленая линия отображает среднее значение, тогда как светло розовая и светло зеленая линии отображают максимальное и минимальное значения в данный момент времени. Пространство между максимумами и минимумами заполнено желтым цветом.

Рабочее время (рабочие дни) отображаются на графиках белым фоном, а нерабочее время отображается серым (при теме *Оригинальная синяя*, по умолчанию в веб-интерфейсе).



Рабочее время всегда отображается на простых графиках, тогда как отображение рабочего времени на [пользовательских графиках](#) является предпочтением пользователя.

Рабочее время не отображается, если график отображает период более 3 месяцев.

Формирование из истории/динамики изменений

Графики могут быть отрисованы на основе [истории или динамики изменений](#) элемента данных.

У пользователей, у которых в веб-интерфейсе активирован [режим отладки](#), снизу справа от графика отображается вертикальная подпись серым цветом, которая указывает откуда пришли данные.

Два фактора, которые влияют что будет использовано история или динамика изменений:

- длительность истории элемента данных. Например, история элемента данных хранится 14 дней. В этом случае, любые данные старше четырнадцати дней будут взяты из динамики изменений.
 - перегрузка данных в графике. Если отображаемое количество секунд по горизонтальным пикселям графика превышает $3600/16$, то тогда будут отображены данные из динамики изменений (даже если история элемента данных еще имеется за этот период).
 - если динамика изменений отключена, то для построения графика будет использоваться история элемента данных - если история доступна за требуемый период. Такое поведение поддерживается начиная с Zabbix 2.2.1 (до этого, отключенная динамика изменений означала пустой график за выбранный период, даже если история была доступна).

Переключение к сырым данным

Выпадающее меню в верхнем правом углу выше графика позволяет переключаться с простого графика на список *Значения/500 последних значений*. Эта функция может быть полезна для просмотра числовых значений, создающих график.

Значения представленные здесь “сырые”, то есть без единиц измерения и без постобработки значений. Однако, преобразования значений здесь также применяются.

Известные проблемы

Смотрите известные проблемы по графикам.

7.6.1.2 Пользовательские графики

Обзор

Пользовательские графики, как следует из названия, предлагают возможность индивидуальной настройки.

В то время как простые графики хороши для просмотра данных по одному элементу данных, они не дают возможности индивидуальной настройки.

Таким образом, если вы захотите изменить стиль графика или способ отрисовки линий, или сравнить несколько элементов данных, например, входящий и исходящий трафик на одном графике, то вам необходим пользовательский график.

Пользовательские графики настраиваются вручную.

Они могут быть созданы для одного узла сети, для нескольких узлов или в пределах одного шаблона.

Настройка пользовательских графиков

Для создания пользовательского графика, сделайте следующее:

- Перейдите в *Настройка* → *Узлы сети* (или *Шаблоны*)
- Нажмите на *Графики* в строке с желаемым узлом сети или шаблоном
- На странице графиков нажмите на *Создать график*
- Измените атрибуты графика

NAME	FUNCTION	DRAW STYLE	Y AXIS SIDE	COLOUR	ACTION
1: New host: Outgoing network traffic on eth0	avg	Filled region	Right	00C800	Remove
2: New host: Incoming network traffic on eth0	avg	Bold line	Right	C80000	Remove

Все обязательные поля ввода отмечены красной звёздочкой.

Атрибуты графика:

Параметр Описание

Имя

7.6.1.3 Ситуационные графики

Обзор

В то время как [простой график](#) отличное средство доступа к данным одного элемента данных, а [пользовательские графики](#) предлагают варианты настройки, ни один из них не позволяет быстро создавать графики для сравнения нескольких элементов данных с минимальными усилиями и без последующей поддержки этого графика.

С целью исправить эту проблему, начиная с версии Zabbix 2.4 имеется возможность создать ситуационные графики по некоторым элементам данных на лету.

Настройка

Для создания ситуационного графика, выполните следующее:

- Перейдите в *Мониторинг* → *Последние данные*
- Используйте фильтр, чтобы отображались те элементы данных, которые вам необходимы
- Выберите те элементы данных, которые желаете увидеть на графике
- Нажмите на кнопки *Показать стекируемыхый график* или *Показать график*

Latest data

Host groups: type here to search Select Name: Download speed for scenario "Zabbix frontend"

Hosts: My host Zabbix server Select Show items without data: Show details:

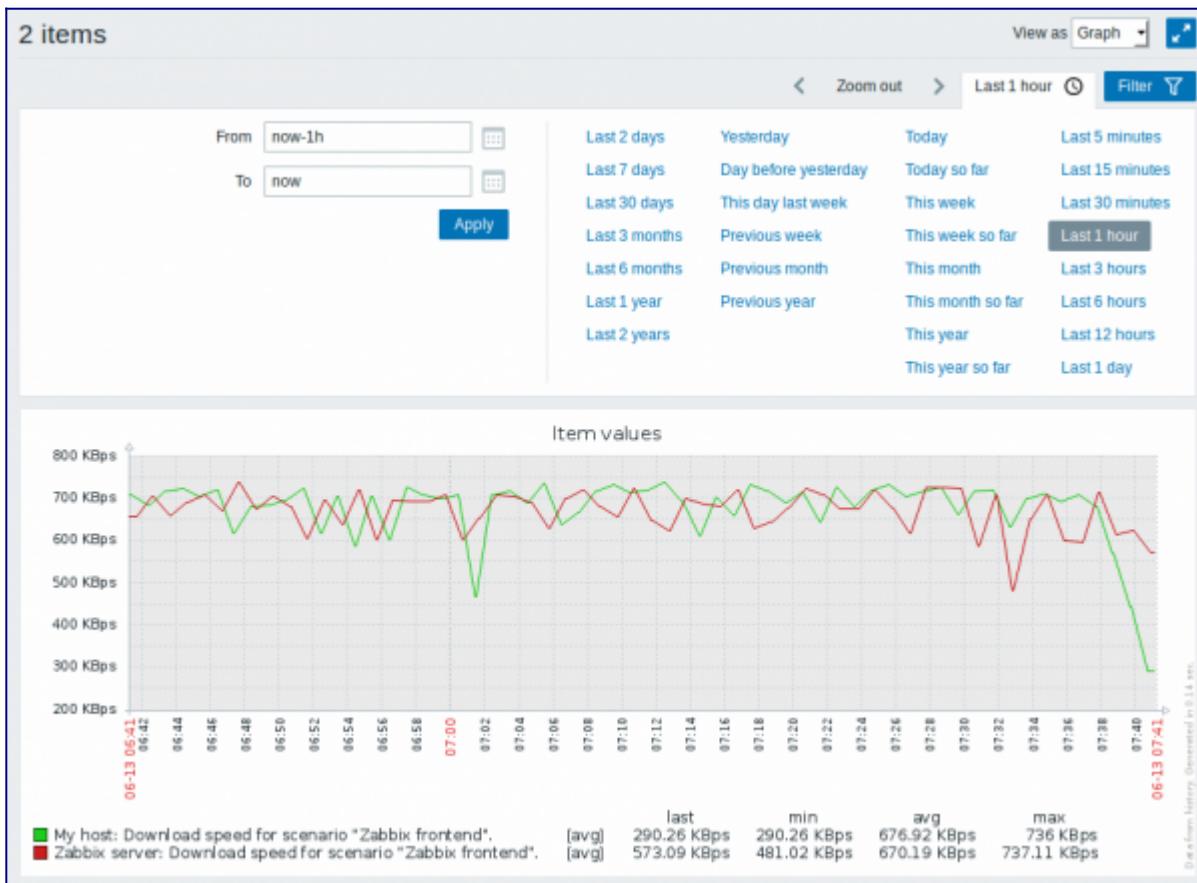
Application: Zabbix frontend Select

Apply Reset

Host	Name	Last check	Last value	Change	Graph
My host	Zabbix frontend (1 item)				
<input checked="" type="checkbox"/>	Download speed for scenario "Zabbix frontend".	2018-06-13 07:42:52	502.11 Kbps	-58.14 Kbps	Graph
Zabbix server	Zabbix frontend (1 item)				
<input checked="" type="checkbox"/>	Download speed for scenario "Zabbix frontend".	2018-06-13 07:43:01	593.04 Kbps	-54.14 Kbps	Graph

2 selected [Display stacked graph](#) [Display graph](#)

График будет создан мгновенно:



Обратите внимание, что во избежании отображения слишком большого количества линий на графике, показываются только средние значения по каждому элемента данных (линии минимального и максимального значения не отображаются). Также на графике не отображаются триггеры и информация с триггеров.

В созданном окне графика у вас имеется [выбор периода времени](#) и возможность переключения из “нормального” линейного графика на стекируемый (и обратно).

2 items

View as Graph

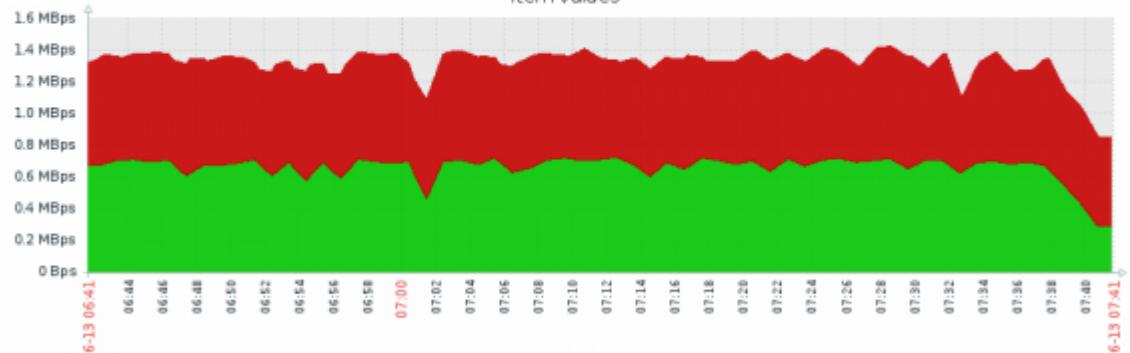
Zoom out

Last 1 hour

Filter

Graph type Normal Stacked

Item values



© Zabbix history generated in 0.48 sec.

7.6.2 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](#).

7.6.2.1 Настройка карты сети

Обзор

Настройка карты сети в Zabbix требует, что вы сначала создадите карту определив её общие параметры и затем начнёте заполнять карту элементами и связями между этими элементами.

Вы можете заполнять карту элементами: узлами сети, группами узлов сети, триггерами, изображениями или другими картами сети.

Для отображения элементов карты используются иконки. Вы можете указать информацию, которая будет отображена с иконками и установить, какие недавние недавние проблемы будут отображаться особым образом. Вы можете связать иконки и задать информацию, которая будет отображаться у связей.

Вы можете добавить пользовательские URL'ы, которые будут доступны при нажатии на иконках. Таким образом вы можете связать иконку узла сети со свойствами узла сети или иконку карты сети с другой картой.

Карты сети, которые уже готовы, можно просмотреть в Мониторинг → [Карты сети](#). На странице мониторинга вы можете нажать на иконки и воспользоваться ссылками на какие-нибудь скрипты или URL'ы.

Начиная с Zabbix 3.4, карты сети основаны на векторной графике (SVG).

Публичные и приватные карты

Все пользователи Zabbix (включая пользователей не являющихся администраторами) могут создавать карты сети. У карт сетей имеется владелец - пользователь, который создал их.

- *Публичные* карты сети видны всем пользователям, однако, эти пользователи должны иметь как минимум права на чтение по крайней мере одного элемента карты сети, чтобы её увидеть. Публичные карты можно редактировать в случае, если пользователь/группа пользователей имеет права чтения-записи к этой карте и по крайней мере права чтения всех элементов соответствующей карты, включая триггеры в связях.
- *Приватные* карты сети видны только своим владельцам и пользователям/группам пользователей с [общим доступом](#) к этой карте сети, которым поделился владелец. Обычные пользователи (не Супер Администраторы) могут предоставлять общий доступ только тем группам и пользователям, которым они принадлежат сами. Пользователи уровня Администратора могут видеть приватные карты сети независимо от того, являются ли они владельцами или принадлежат списку пользователей с общим доступом. Приватные карты может редактировать владелец карты и пользователь/группа пользователей с правами чтения-записи этой карты сети и по крайней мере с правами чтения всех элементов соответствующей карты, включая триггеры в связях.

Элементы карты, к которым у пользователя нет прав чтения, отображаются серой иконкой, а вся текстовая информация по элементу скрывается. Однако, подпись к триггеру видна даже, если у пользователя отсутствуют права к этому триггеру.

Чтобы добавить элемент на карту, у пользователя должны быть как минимум права на чтение этого элемента.

Создание карты сети

Для создания карты, сделайте следующее:

- Перейдите в *Мониторинг* → *Карты сети*
- Перейдите к просмотру всех карт
- Нажмите на *Создать карту сети*

Чтобы создать новую карту сети, вы также можете использовать кнопки *Клонировать* и *Полное клонирование* в диалоге конфигурации существующей карты. При нажатии на *Клонировать* будут сохранены все атрибуты макета оригинальной карты, но без элементов. При *Полном клонировании* сохранятся как общие атрибуты макета, так и все элементы оригинальной карты сети.

Вкладка **Карта сети** содержит общие атрибуты карт сетей:

The screenshot shows the 'Map' configuration page in Zabbix. At the top, there are tabs for 'Map' and 'Sharing'. Below the tabs, there are several input fields and dropdown menus:

- Owner:** Admin (Zabbix Administrator) (marked with a red asterisk)
- Name:** Local network
- Width:** 680
- Height:** 600
- Background image:** No image (dropdown menu)
- Automatic icon mapping:** <manual> (dropdown menu) with a link to 'show icon mappings'
- Icon highlight:** checked (checkbox)
- Mark elements on trigger status change:** checked (checkbox)
- Display problems:** Expand single problem (selected), Number of problems, Number of problems and expand most critical one
- Advanced labels:** checked (checkbox)
- Host group label type:** Label (dropdown menu)
- Host label type:** Label (dropdown menu)
- Trigger label type:** Status only (dropdown menu)
- Map label type:** Label (dropdown menu)
- Image label type:** Nothing (dropdown menu)
- Map element label location:** Bottom (dropdown menu)
- Problem display:** All (dropdown menu)
- Minimum severity:** Not classified (selected), Information, Warning, Average, High, Disaster
- Show suppressed problems:** unchecked (checkbox)
- URLs:** A table with one row:

Name	URL	Element
Latest data	https://localhost/zabbix/latest.php	Host

 Buttons at the bottom: Add (highlighted in blue), Cancel.

Все обязательные поля ввода отмечены красной звёздочкой.

Общие атрибуты карты сети:

Параметр

Владелец

Имя

Ширина

Высота

Фоновое изображение

Автоматическое соответствие иконок

Подсветка иконок

Описание

Имя владельца карты.

Уникальное имя карты.

Ширина карты в пикселях.

Высота карты в пикселях.

Использование фонового изображения:

Нет изображения - без фонового изображения (белый фон)

Изображение - выбранное изображение, которое используется как фоновое изображение. Масштабирование не производится. Вы можете использовать географическую карту или любое другое изображение для улучшения вашей карты.

Вы можете указать использование автоматического соответствия иконок, заданные в Администрирование → Общие → Соответствие иконок. Соответствие иконок позволяет отображать некоторые иконки в соответствии с полями инвентарных данных узлов сети.

Если вы отметите эту опцию, элементы карты получат подсветку.

Элементы с активными триггерами будут иметь фон в виде круга, такого же цвета, что и триггер с наивысшей важностью. Кроме того, будет отображаться зеленая толстая линия вокруг круга, если все проблемы подтверждены.

Если элемент в состоянии “деактивирован” или “в обслуживании”, будет использоваться фон в виде квадрата, серый и оранжевый соответственно.

Параметр	Описание
Помечать элементы при изменении состояния триггера	<p>Смотрите также: Просмотр карт сети</p> <p>Недавнее изменение состояния триггера (недавняя проблема или её решение) будет подсвечиваться маркерами (указывающие внутрь красные треугольники) по трем сторонам иконки элемента, которые не заняты подписью. Маркеры отображаются 30 минут.</p>
Отображение проблем	<p>Выбор способа отображения проблемы с элементом карты:</p> <p>Развертывание одиночной проблемы - если у элемента карты имеется одна проблема, отображать имя проблемы (триггера)</p> <p>Количество проблем - отображается общее количество проблем</p> <p>Количество проблем и раскрывать наиболее критичную - отображается имя проблемы (триггера) по самому критичному триггеру, а также общее количество проблем. Определение 'самого критичного' основывается на:</p> <p><i>Для элемента триггера на карте</i> - важность триггера и его позиция в списке триггеров;</p> <p><i>Для остальным элементов карты</i> - важность триггера и ID триггера.</p> <p>Если эта опция выбрана, у вас будет возможность задать тип подписи для разных типов элементов.</p> <p>Тип подписи, который будет использоваться для элементов карты:</p> <p>Подпись - подпись к элементу карты</p> <p>IP адрес - IP адрес</p> <p>Имя элемента - имя элемента (например, имя узла сети)</p> <p>Только состояние - только состояние (OK или ПРОБЛЕМА)</p> <p>Ничего - подписи не будут отображаться</p> <p>Расположение подписи по отношению к элементу карты:</p> <p>По нижнему краю - ниже иконки</p> <p>По левой стороне - слева</p> <p>По правой стороне - справа</p> <p>По верхнему краю - выше элемента карты</p> <p>Отображение количества проблем как:</p> <p>Все - будет отображено полное количество проблем</p> <p>Неподтвержденные отдельно - количество неподтвержденных проблем будет отображено отдельным числом из общего количества проблем</p> <p>Только неподтвержденные - будет отображено только количество неподтвержденных проблем</p> <p>Проблемы с важностью ниже заданной в данной опции не будут отображены на карте. Например, с выбранной важностью <i>Предупреждение</i>, изменения состояний триггеров с <i>Информация</i> и <i>Не классифицировано</i> не будут отражены на карте.</p> <p>Параметр поддерживается начиная с Zabbix 2.2.</p> <p>Отметьте опцию отображения проблем, которые в противном случае были бы подавлены (не отображены) по причине обслуживания узла сети.</p> <p>Можно указать URL'ы для каждого типа элемента (с подписью). Они будут отображаться как ссылки, если пользователь нажмет на элементе в режиме просмотра карты.</p> <p>В URL'ах карт можно использовать макросы: {MAP.ID}, {HOSTGROUP.ID}, {HOST.ID}, {TRIGGER.ID}</p>
URL'ы	
Общий доступ	

Вкладка **Общий доступ** содержит как тип карты сети, так и опции общего доступа (группы пользователей, пользователи) для приватных карт сети:

Параметр

Описание

Tip

Список групп пользователей участников

Список пользователей участников

Когда вы нажмете *Добавить* для сохранения карты сети, вы создадите пустую карту сети с именем, размерами и определенными настройками. Теперь вам необходимо добавить какие-нибудь элементы. Чтобы это сделать, нажмите на *Конструктор* в списке карт сетей, чтобы открыть область редактирования.

Добавление элементов

Для добавления элемента, нажмите на ссылку *Добавить* следующую за подписью 'Элемент карты сети'. Новый элемент появится в левом верхнем углу карты. Возьмите и переместите куда бы вам ни захотелось.

Обратите внимание с "Вкл" опцией Сетки, элементы будут всегда выравниваться по сетке (вы можете выбрать различные размеры сетки из выпадающего меню, а также скрыть/показать сетку). Если вы хотите разместить элементы где бы то ни было без выравнивания, переключите опцию в "Выкл". (Элементы, расположенные случайным образом, можно в дальнейшем снова выровнять по сетке с помощью кнопки *Выровнять элементы карты*.)

Теперь у вас есть несколько размещенных элементов, вы можете захотеть начать различать их, дав имена и т.п. Нажав на элемент, отобразится диалог и вы сможете задать тип элемента, дать имя, выбрать другую иконку и так далее.

Map element: Add / Remove Shape: Add / Remove Link: Add / Remove Expand macros: Off Grid: Shown / On 50x50 Align map elements Update

Атрибуты элементов карт сетей:

Параметр

Описание

Тип

Тип элемента:

Узел сети - иконка, отображающая состояние всех триггеров у выбранного узла сети

Подпись

Карта сети - иконка, отображающая состояние всех элементов на карте сети

Триггер - иконка, отображающая состояние одного и более триггеров

Группа узлов сети - иконка, отображающая состояние всех триггеров у на всех узлах сети в выбранной группе узлов сети

Изображение - иконка, не связанная с каким-то ресурсом

Подпись к иконке, любая строка.

В подписях можно использовать [макросы](#) и многострочные строки.

Расположение подписи по отношению к иконке:

По умолчанию - расположение подписи по умолчанию карты сети

По нижнему краю - ниже иконки

По левой стороне - слева

По правой стороне - справа

По верхнему краю - выше иконки

Расположение подписи

Узел сети

Введите узел сети, если тип элемента 'Узел сети'. Это поле с функцией автодополнения, таким образом после начала ввода имени узла сети, будет автоматически предложен список совпадающих узлов сети в выпадающем

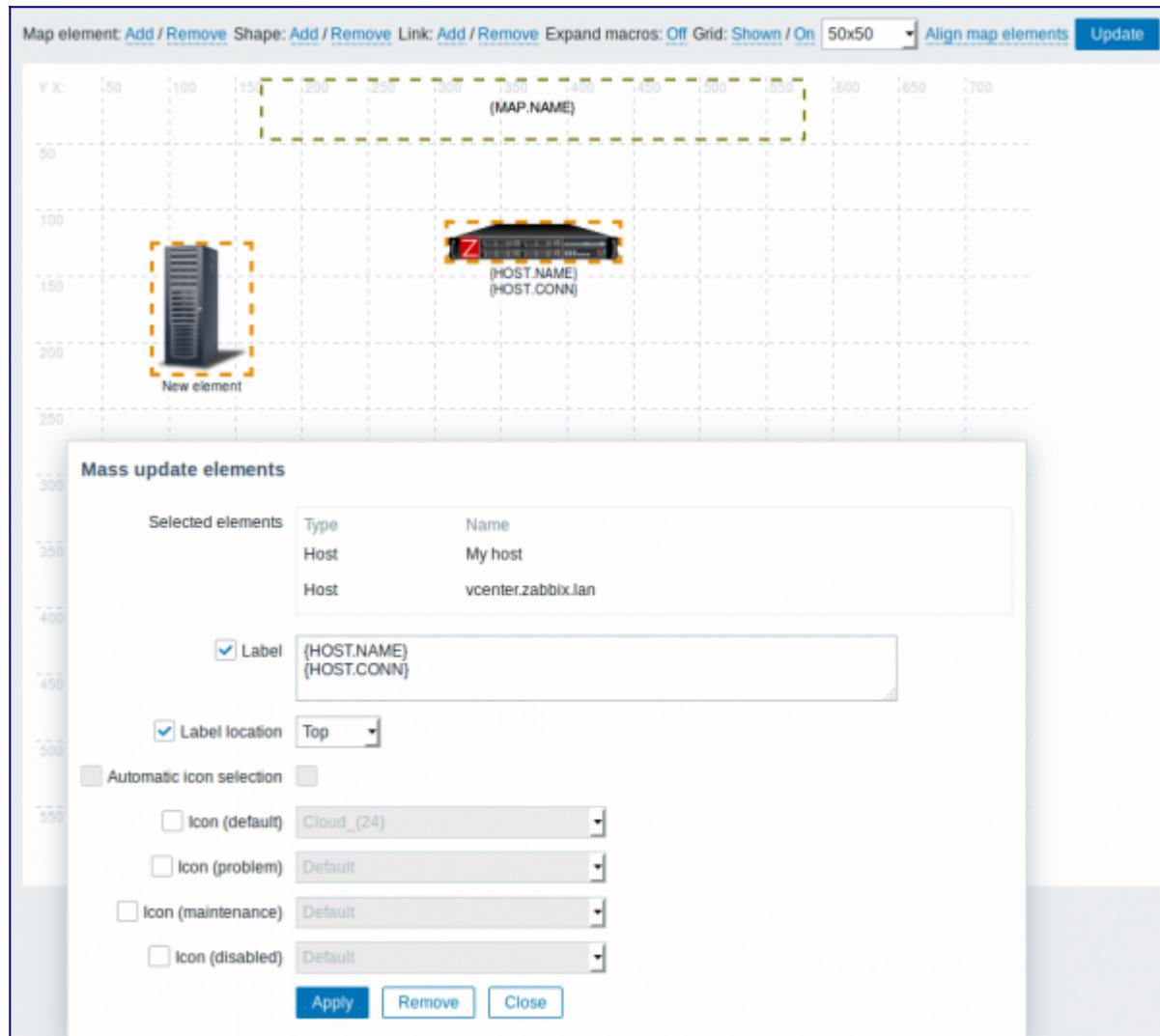
Параметр	Описание	
Карта сети	<p>меню. Прокрутите список, чтобы выбрать необходимый. Нажмите на 'x', чтобы удалить выбранный.</p> <p>Выберите карту сети, если тип элемента 'Карта сети'.</p> <p>Если тип элемента 'Триггер', выберите один или более триггеров в поле <i>Новые триггера</i> снизу и нажмите на <i>Добавить</i>.</p> <p>Порядок выбранных триггеров можно изменить, но только в пределах одной важности триггеров. Несколько выбранных триггеров также влияют на раскрытие макросов {HOST.*} как в режиме конструктора, так и в режиме просмотра.</p> <p>1 В режиме конструктора первые отображаемые макросы {HOST.*} будут раскрыты в зависимости от первого триггера в списке (на основании важности триггера).</p> <p>2 Режим просмотра зависит от параметра Отображения проблем в общих атрибутах карты сети.</p> <ul style="list-style-type: none"> * Если выбран режим <i>Развёртывание одиночной проблемы</i>, тогда первые отображаемые макросы {HOST.*} будут раскрыты в зависимости от последней обнаруженной проблемы триггера (вне зависимости от важности) или от первого триггера в списке (в случае отсутствия обнаруженных проблем); * Если выбран режим <i>Количество проблем и раскрывать наиболее критичную</i>, тогда первые отображаемые макросы {HOST.*} будут раскрыты в зависимости от важности триггера. <p>Выберите группу узлов сети, если тип элемента 'Группа узлов сети'. Это поле с функцией автодополнения, таким образом после начала ввода имени группы узлов сети, будет автоматически предложен список совпадающих групп узлов сети в выпадающем меню. Прокрутите список, чтобы выбрать необходимую. Нажмите на 'x', чтобы удалить выбранную.</p> <p>Вы можете выбрать группу элементов данных, позволяющую отображать только те проблемные триггеры, которые принадлежат указанной группе элементов данных.</p> <p>Это поле доступно для типов элементов данных узла сети и группы узлов сети и поддерживается начиная с Zabbix 2.4.0.</p>	
Группа узлов сети	<p>Выберите группу узлов сети, если тип элемента 'Группа узлов сети'. Это поле с функцией автодополнения, таким образом после начала ввода имени группы узлов сети, будет автоматически предложен список совпадающих групп узлов сети в выпадающем меню. Прокрутите список, чтобы выбрать необходимую. Нажмите на 'x', чтобы удалить выбранную.</p> <p>Вы можете выбрать группу элементов данных, позволяющую отображать только те проблемные триггеры, которые принадлежат указанной группе элементов данных.</p> <p>Это поле доступно для типов элементов данных узла сети и группы узлов сети и поддерживается начиная с Zabbix 2.4.0.</p>	
Группа элементов данных	<p>Иконка (по умолчанию)</p> <p>Автоматический выбор иконки</p> <p>Иконки</p> <p>Координата X</p> <p>Координата Y</p> <p>URL'ы</p>	<p>Используемая иконка.</p> <p>В этом случае соответствие иконок будет определять, какую иконку отображать.</p> <p>Вы можете выбрать различные иконки для элемента для отображения этих случаях: по умолчанию, проблема, обслуживание, деактивирован.</p> <p>Координата X элемента карты сети.</p> <p>Координата Y элемента карты сети.</p> <p>URL'ы можно указать для каждого отдельного элемента. Они будут отображаться как ссылки, если пользователь нажмёт на элементе в разделе мониторинга. Если элемент имеет собственные URL'ы, а также URL'ы уровня карты сети для его типа элемента, то ссылки будут объединены в одно меню. Макросы, которые можно использовать в URL'ах карт сети: {MAP.ID}, {HOSTGROUP.ID}, {HOST.ID}, {TRIGGER.ID}</p> <p>Добавленные элементы не сохраняются автоматически. Если вы покинете эту страницу, все изменения могут быть потеряны.</p> <p>Поэтому будет хорошей идеей нажать на кнопке Обновить в правом верхнем углу. При нажатии, все изменения сохраняются независимо от того что вы выберите в дальнейшем во всплывающем окне.</p> <p>Выбранные опции сетки также сохраняются для каждой карты сети.</p>

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 (опция доступна начиная с Zabbix 2.0).

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, {HOSTNAME} для подписи элемента).



Связи элементов

После того, как вы расставите элементы на карте, самое время соединить их. Для соединения двух элементов вы должны сначала выбрать их. Когда элементы будут выбраны, нажмите на *Добавить* справа от подписи *Связь*.

При наличии созданной связи, диалог одного элемента теперь содержит дополнительный раздел *Связи*. Нажмите на *Изменить* для редактирования атрибутов связи.

Map element: Add / Remove Shape: Add / Remove Link: Add / Remove Expand macros: Off Grid: Shown / On 50x50 Align map elements Update

The screenshot shows a network monitoring application interface. At the top, there are buttons for managing map elements (Add / Remove), shapes (Add / Remove), links (Add / Remove), expanding macros (Off), and grid settings (50x50). Below this is a toolbar with alignment options (Align map elements) and an update button.

The main area displays a map with a host icon labeled "New element" connected to another host icon labeled "[HOST.NAME] (HOST.CONN)" via a green line labeled "100Mbps". A dashed yellow box highlights the connection line.

A detailed configuration dialog is open on the right side:

- Map element** section:
 - Type: Host
 - Label: New element
 - Label location: Default
 - * Host: My host (with a Select button)
 - Application: (with a Select button)
 - Automatic icon selection: (checkbox)
 - Icons section:

Default	Server_(96)
Problem	Default
Maintenance	Default
Disabled	Default
 - Coordinates: X: 89, Y: 127
- URLs** section:

Name	URL	Action
		Remove
Add		

 Buttons: Apply, Remove, Close
- Links** section:

Element name	Link indicators	Action
vcenter.zabbix.lan		Edit
- Label** section:

100Mbps

- Connect to**: vcenter.zabbix.lan
- Type (OK)**: Bold line
- Colour (OK)**: #00CC00
- Link indicators** section:

Trigger	Type	Colour	Action
Add			

 Buttons: Apply, Remove, Close

Атрибуты связей:

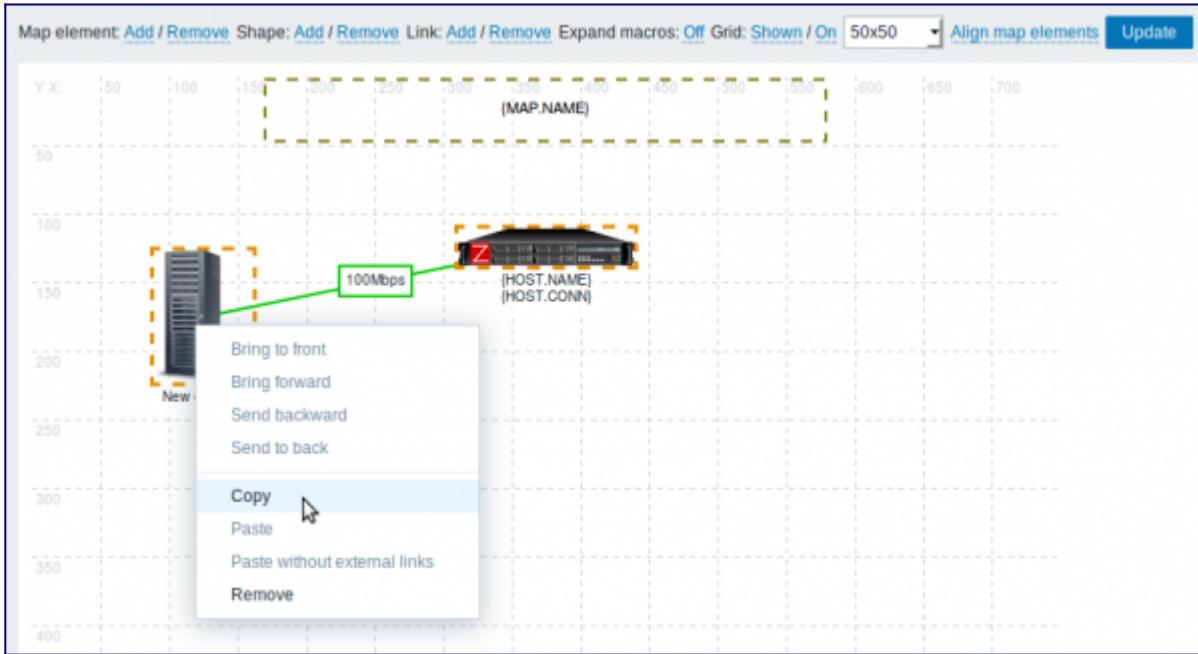
Параметр	Описание
Подпись	Подпись, которая отображается поверх связи. В этом поле поддерживается макрос <code>{узел_сети:ключ.функция(параметр)}</code> , но только функции триггеров avg, last, min и max с параметром в виде секунд.
Подключаться через	Элемент, к которому присоединяется связь.

Параметр	Описание
<i>Tun (OK)</i>	Стиль связи по умолчанию: Линия - одиночная линия Жирная линия - жирная линия Точечная линия - точки Пунктирная линия - пунктирная линия
<i>Цвет (OK)</i>	Цвет связи по умолчанию.
<u>Индикаторы связи</u>	Список триггеров соединенных со связью. В случае, если триггер в состоянии ПРОБЛЕМА, его стиль применяется к связи.

Перемещение и копирование-вставка элементов

Несколько элементов можно **переместить** в другое место на карте, нажав на одном из выбранных элементов и удерживая кнопку мыши нажатой переместить курсор в желаемое место.

Один или много элементов можно **скопировать** выбрав эти элементы, затем нажать на выбранном элементе правой кнопкой мыши и из меню выбрать **Копировать**.



Чтобы вставить скопированные элементы, нажмите в области карты правой кнопкой мыши и из меню выберите **Вставить**. Опция **Вставить без внешних связей** вставит элементы и оставит только те связи, которые имеются между выбранными элементами.

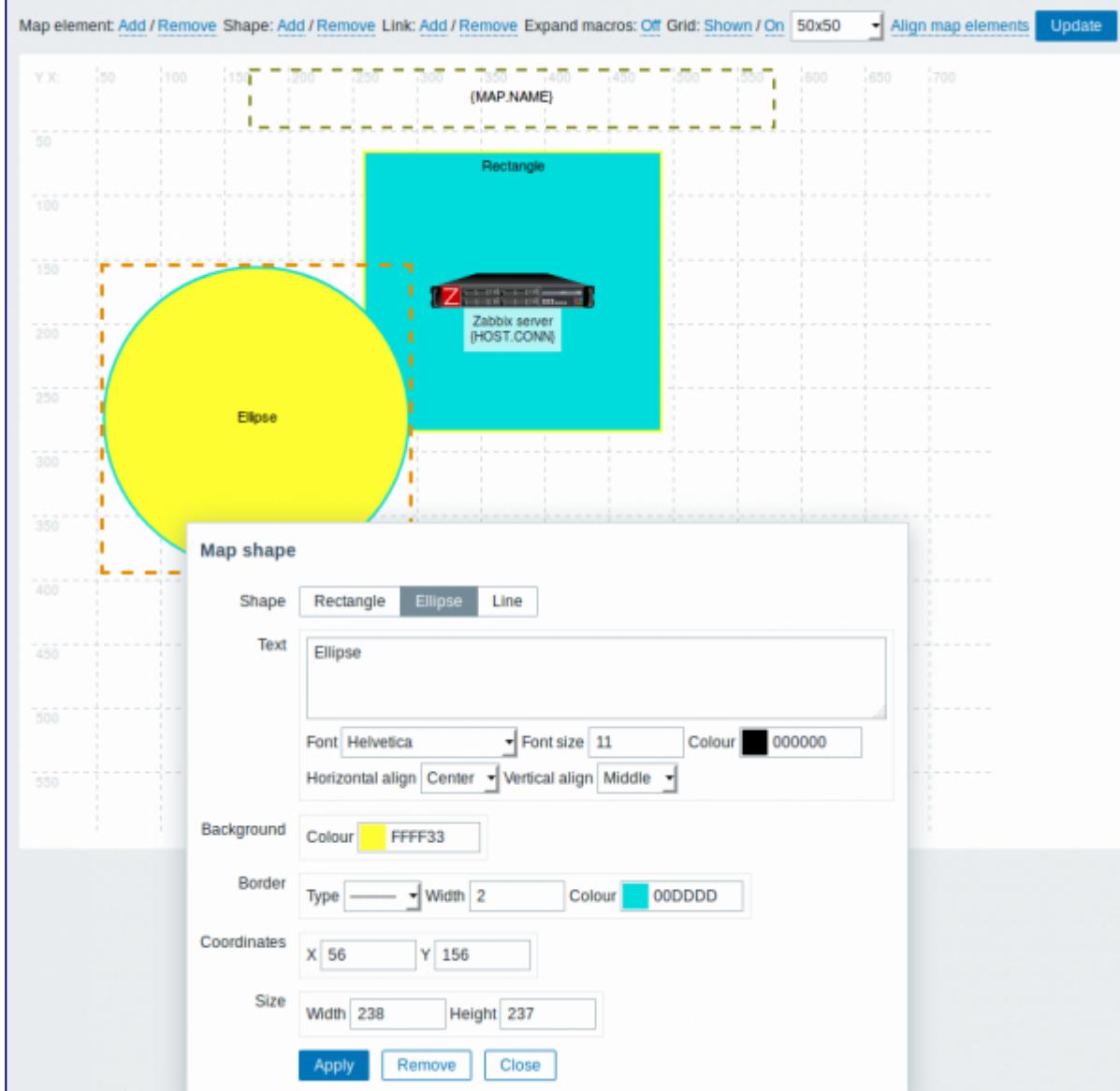
Копирование-вставка работает только в одном окне браузера. Быстрые сочетания клавиш не поддерживаются.

Добавление фигур

В дополнение к элементам карты имеется также возможность добавления некоторых фигур. Фигуры не являются элементами карты; они являются лишь визуальным представлением. Например, фигуру прямоугольника можно использовать фоном для группировки некоторых узлов сети. Можно добавить фигуры прямоугольника и эллипса.

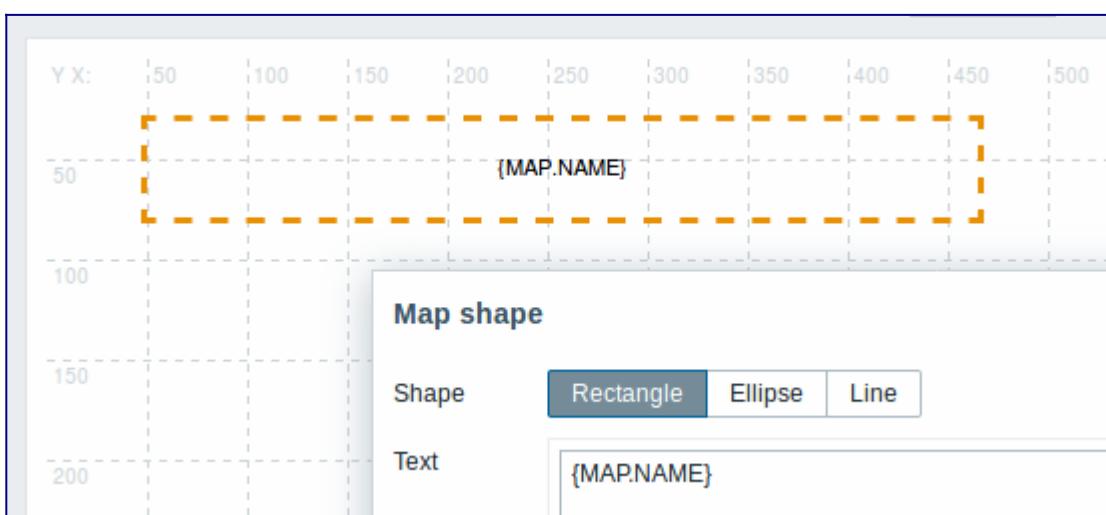
Чтобы добавить фигуру, нажмите на **Добавить** после надписи **Фигура**. Новая фигура появится в левом верхнем углу карты сети. Переместите её в любое место куда пожелаете.

Новая фигура добавляется с цветами по умолчанию. При нажатии на фигуре отобразится диалог и вы сможете изменить внешний вид фигуры, добавить текст и так далее.



Чтобы выбрать фигуры, выберите одну и затем удерживая клавишу *Ctrl* выберите остальные. Когда выбрано несколько фигур, можно будет массово обновить общие свойства, аналогично элементам карты сети.

К фигурам можно добавлять текст. Чтобы отображать только текст, фигуру можно сделать невидимой, удалив рамку фигуры (выберите 'Нет' в поле Рамка). Например, обратите внимание на то, как макрос {MAP.NAME}, отображаемый на снимке экрана выше, в действительности является фигурой прямоугольника с текстом, что можно увидеть нажав на макрос:



{MAP.NAME} раскрывается в заданное имя карты при просмотре карты сети.

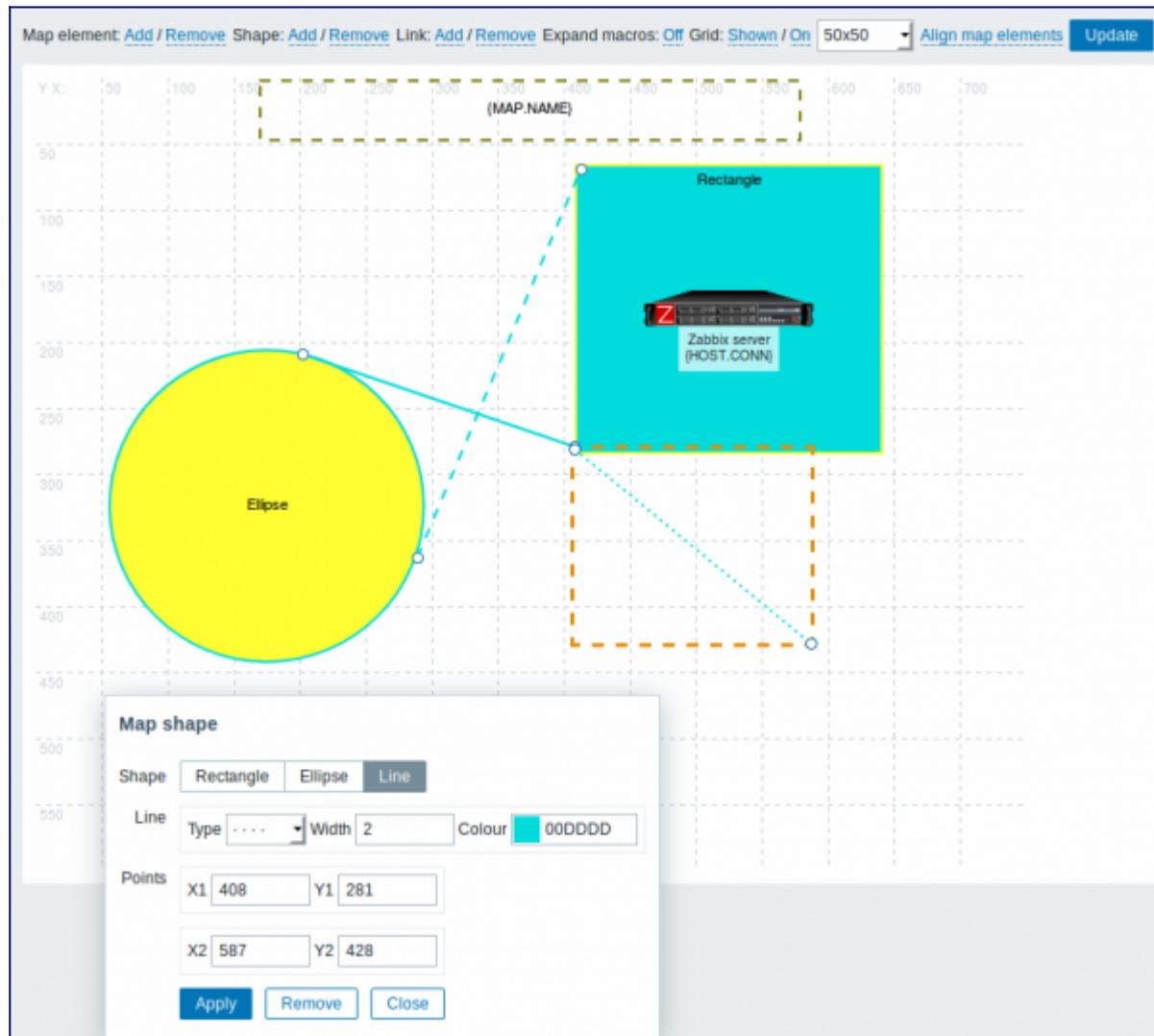
Если в тексте используются гиперссылки, они станут нажимаемыми при просмотре карты сети.

Перенос строк текста всегда “включен” в пределах фигур. Однако, в пределах эллипса строки переносятся так, как будто эллипс это прямоугольник. Перенос слов не реализован, поэтому длинные слова (слова, которые не вписываются в фигуру) не переносятся, но замаскированы (на странице конструктора) или обрезаны (другие страницы с картами).

Добавление линий

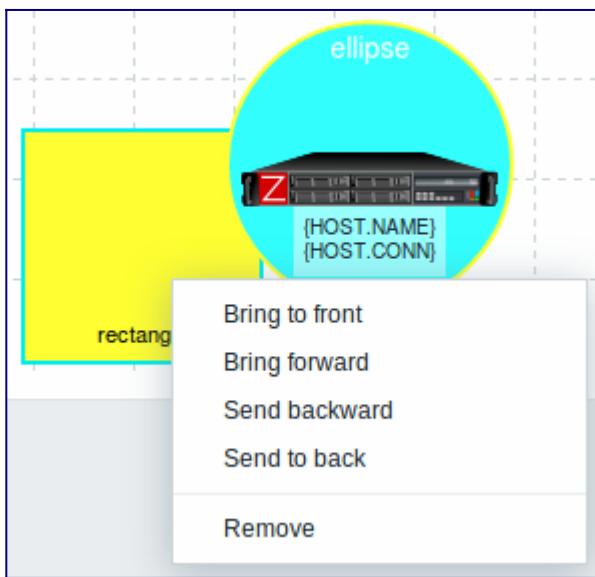
В дополнение к фигурам имеется также возможность добавления некоторых линий. Линии можно использовать для создания связей между элементами или фигурами на карте сети.

Чтобы добавить линию, нажмите на *Добавить* после надписи *Фигура*. Новая фигура появится в левом верхнем углу карты сети. Выберите её и в диалоге редактирования нажмите на *Линия*, чтобы изменить фигуру на линию. Затем настройте свойства линии, такие как тип, ширина, цвет и остальные.



Очередность фигур и линий

Чтобы поместить одну фигуру спереди от других (или наоборот), нажмите на фигуре правой кнопкой мыши и выберите необходимый пункт во всплывающем меню к фигуре.

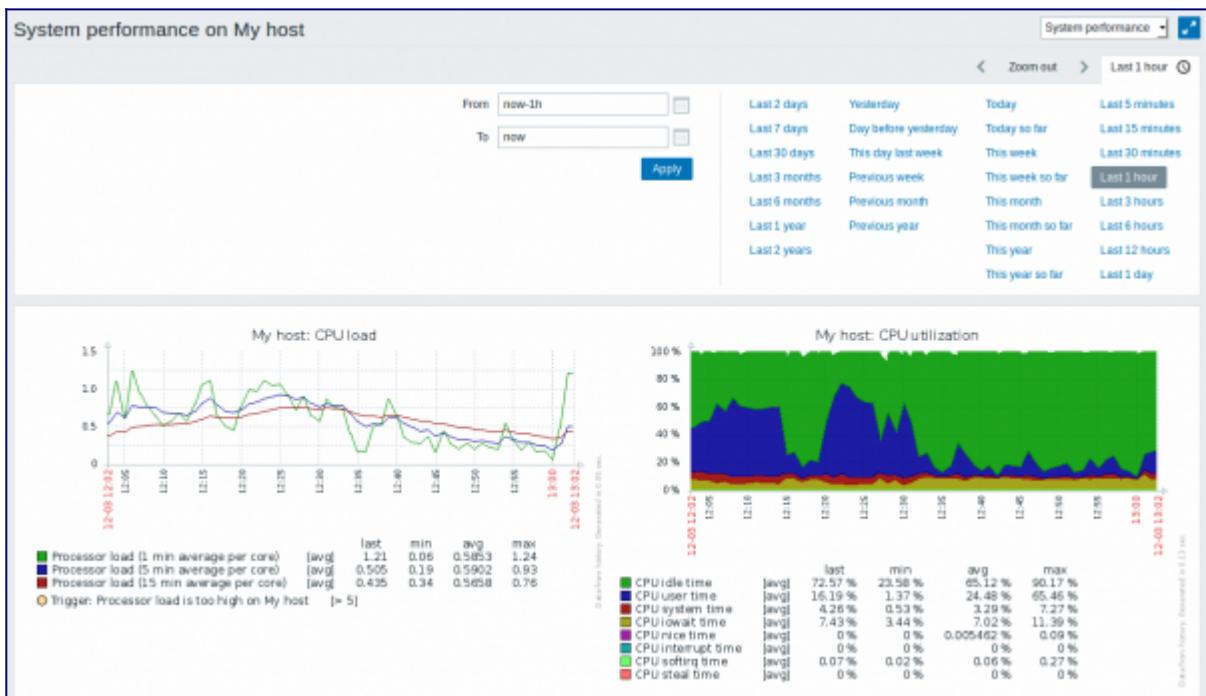


7.6.2.2 Элементы группы узлов сети

Обзор

Этот раздел объясняет как добавить тип элемента “Группа узлов сети” при настройке [карты сети](#).

Настройка



Все обязательные поля ввода отмечены красной звёздочкой.

Эта таблица состоит из параметров, характерных для типа элемента *Группа узлов сети*:

Параметр

Описание

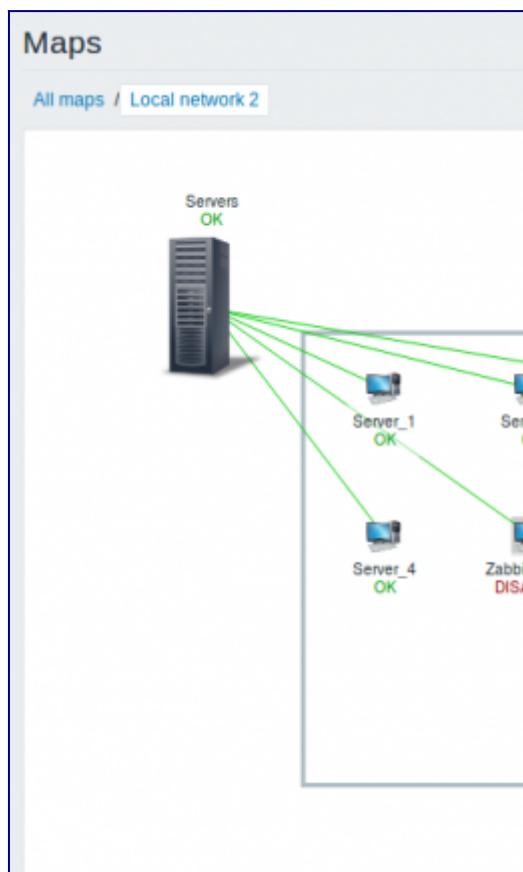
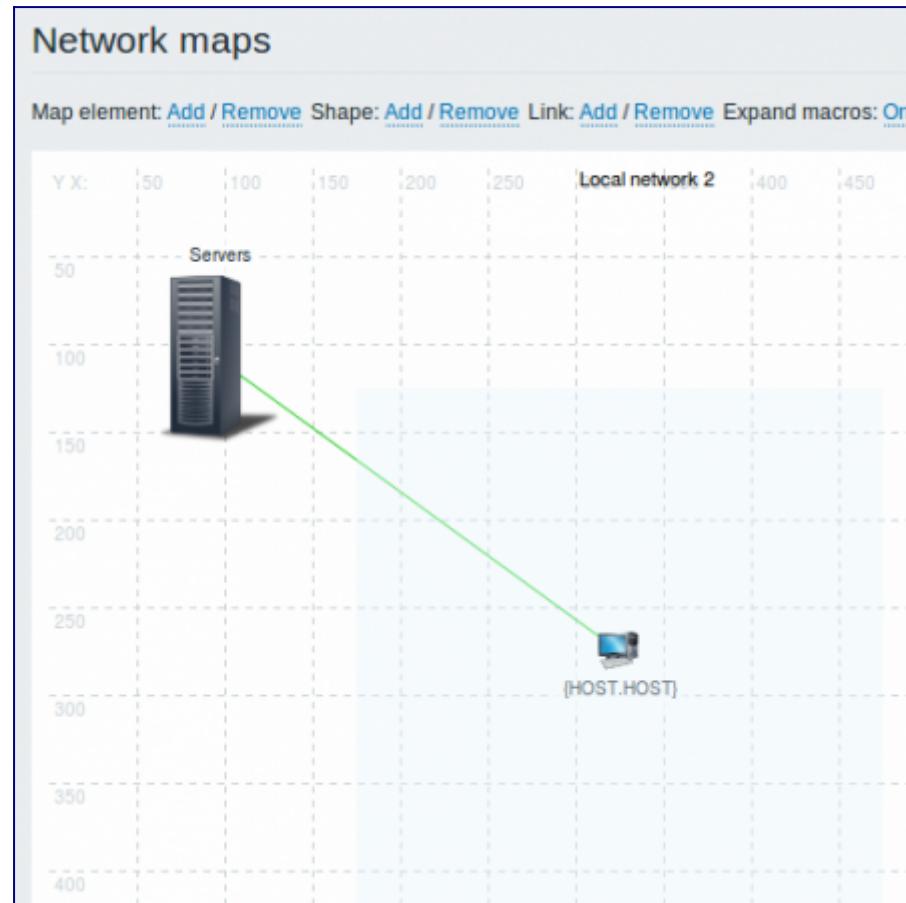
<i>Тип</i>	Выберите Тип элемента: Группа узлов сети - иконка, которая представляет состояние всех триггеров всех узлов сети, которые принадлежат выбранной группе
<i>Отображение</i>	Опции отображения: Группа узлов сети - выбор этой опции приведет к тому, что одна иконка будет отображать соответствующую информацию о конкретной группе узлов сети Элементы группы узлов сети - выбор этой опции приведет к тому, что несколько иконок будут отображать соответствующую информацию по каждому отдельному элементу (узлу сети) из конкретной группы узлов сети
<i>Тип области</i>	Эта опция доступна, если выбран параметр “Элементы группы узлов сети”:

Параметр	Описание
	Вписать в карту сети - все элементы группы узлов сети располагаются равномерно по всей карте сети
	Пользовательский размер - настройка вручную области карты сети для отображения всех элементов группы узлов сети
	Эта опция доступна, если выбраны параметр “Элементы группы узлов сети” и параметр “Тип области” :
Размер области	Ширина - числовое значение, которое необходимо ввести, чтобы указать ширину области карты
	Высота - числовое значение, которое необходимо ввести, чтобы указать высоту области карты
Алгоритм размещения	Сетка – единственная доступная опция отображения всех элементов группы узлов сети
	Подпись к иконке, любая строка. В подписях можно использовать макросы и многострочные строки. Если типом элемента карты сети является “Группа узлов сети”, тогда ввод определенных макросов влияет на отображение на карте сети соответствующей информации по каждомуциальному отдельному узлу сети. Например, если используется макрос {HOST.IP}, при редактировании карты сети отобразится только сам макрос {HOST.IP}, тогда как просмотр карты сети будет включать и отображать уникальный IP адрес каждого узла сети
Подпись	

Просмотр элементов групп узлов сети

Эта опция доступно только, если выбрана опция отображения “Элементы группы узлов сети”. При выборе “Элементы группы узлов сети” опцией *отобразить*, вы сначала увидите только одну иконку для группы узлов сети. Однако, когда вы сохраните карту сети и перейдете в просмотр карты сети, вы увидите, что карта сети включает в себя все элементы (узлы сети) заданной группы узлов сети:

Вид редактирования карты сети



Обратите внимание, как используется макрос {HOST.NAME}. В диалоге редактирования карты сети имя макроса не раскрывается, тогда как при просмотре карты сети отображаются все уникальные имена узлов сети.

7.6.2.3 Индикаторы связей

Обзор

Вы можете назначить несколько триггеров на [связь](#) между элементами на карте сети. Когда эти триггеры перейдут в состояние проблемы, связь сможет отразить это.

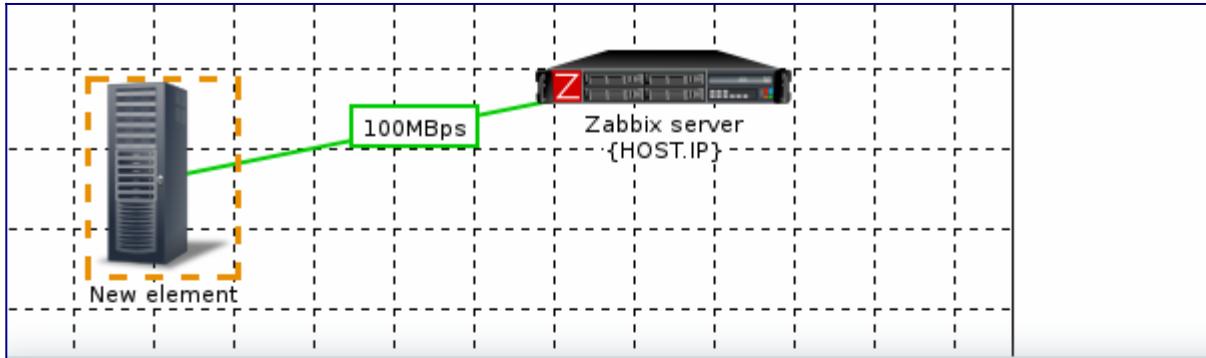
Когда вы настраиваете связь, вы указываете значения по умолчанию для типа и цвета связи. Когда вы назначаете триггеры к связи, вы можете назначить разные типы и цвета у связи с помощью этих триггеров.

Если какие-либо из этих триггеров перейдут в состояние проблема, их стиль связи и цвет будет отображен на связи. Возможно, цвет по умолчанию вашей связи был зеленой линией. Теперь, когда триггер в состоянии проблема, ваша связь стала жирной красной линией (если вы определили её так).

Настройка

Для назначения триггеров как индикаторов связи, сделайте следующее:

- выберите элемент карты
- нажмите на *Изменить* в разделе *Связи* у соответствующей связи
- нажмите на *Добавить* в блоке *Индикаторы связи* и выберите один или более триггеров



Map element

Type	<input type="button" value="Host"/>								
Label	New element								
Label location	<input type="button" value="Default"/>								
*Host	<input type="text" value="New host"/> <input type="button" value="Select"/>								
Application	<input type="text"/> <input type="button" value="Select"/>								
Automatic icon selection	<input type="checkbox"/>								
Icons	<table border="1"> <tr> <td>Default</td> <td><input type="button" value="Server_(96)"/></td> </tr> <tr> <td>Problem</td> <td><input type="button" value="Server_(128)"/></td> </tr> <tr> <td>Maintenance</td> <td><input type="button" value="Server_(24)"/></td> </tr> <tr> <td>Disabled</td> <td><input type="button" value="Default"/></td> </tr> </table>	Default	<input type="button" value="Server_(96)"/>	Problem	<input type="button" value="Server_(128)"/>	Maintenance	<input type="button" value="Server_(24)"/>	Disabled	<input type="button" value="Default"/>
Default	<input type="button" value="Server_(96)"/>								
Problem	<input type="button" value="Server_(128)"/>								
Maintenance	<input type="button" value="Server_(24)"/>								
Disabled	<input type="button" value="Default"/>								
Coordinates	X <input type="text" value="89"/> Y <input type="text" value="127"/>								
URLs	<table border="1"> <thead> <tr> <th>NAME</th> <th>URL</th> </tr> </thead> <tbody> <tr> <td><input type="text"/></td> <td><input type="text"/></td> </tr> <tr> <td colspan="2"><input type="button" value="Add"/></td> </tr> </tbody> </table> <input type="button" value="Apply"/> <input type="button" value="Remove"/> <input type="button" value="Close"/>	NAME	URL	<input type="text"/>	<input type="text"/>	<input type="button" value="Add"/>			
NAME	URL								
<input type="text"/>	<input type="text"/>								
<input type="button" value="Add"/>									
Links	<table border="1"> <thead> <tr> <th>ELEMENT NAME</th> <th>LINK INDICATORS</th> </tr> </thead> <tbody> <tr> <td>Zabbix server</td> <td>New host: Zabbix agent on New host is unreachable for 5 minutes</td> </tr> </tbody> </table>	ELEMENT NAME	LINK INDICATORS	Zabbix server	New host: Zabbix agent on New host is unreachable for 5 minutes				
ELEMENT NAME	LINK INDICATORS								
Zabbix server	New host: Zabbix agent on New host is unreachable for 5 minutes								
Label	<input type="text" value="100MBps"/>								
Connect to	<input type="button" value="Zabbix server"/>								
Type (OK)	<input type="button" value="Bold line"/>								
Colour (OK)	<input type="color" value="#00CC00"/>								
Link indicators	<table border="1"> <thead> <tr> <th>TRIGGER</th> <th>TYPE</th> <th>COLO</th> </tr> </thead> <tbody> <tr> <td>New host: Zabbix agent on New host is unreachable for 5 minutes</td> <td><input type="button" value="Line"/></td> <td><input type="color" value="#FF0000"/> DI</td> </tr> </tbody> </table>	TRIGGER	TYPE	COLO	New host: Zabbix agent on New host is unreachable for 5 minutes	<input type="button" value="Line"/>	<input type="color" value="#FF0000"/> DI		
TRIGGER	TYPE	COLO							
New host: Zabbix agent on New host is unreachable for 5 minutes	<input type="button" value="Line"/>	<input type="color" value="#FF0000"/> DI							

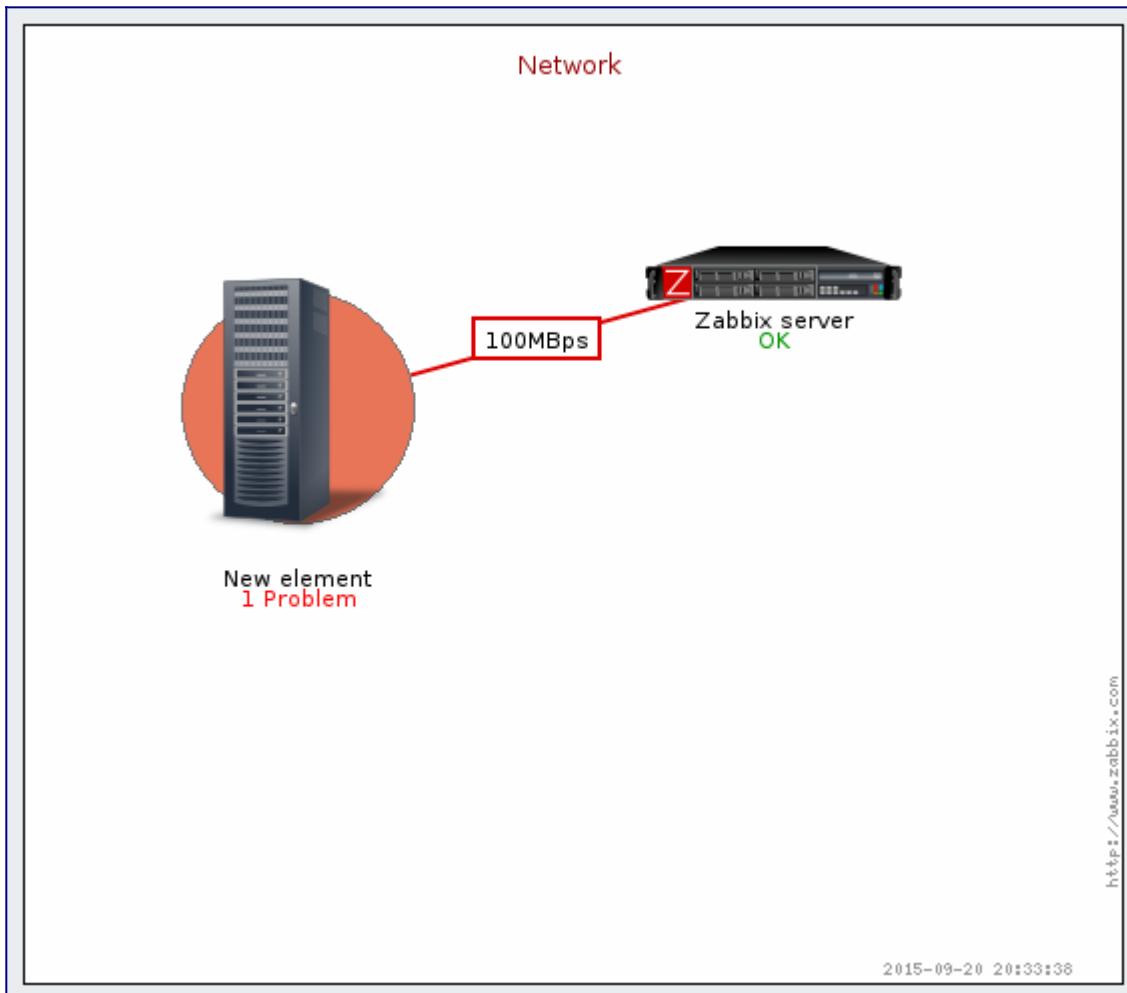
Все обязательные поля ввода отмечены красной звёздочкой.

Добавленные триггеры будут видны в списке *Индикаторы связи*.

Вы можете выбрать тип и цвет связи для каждого триггера напрямую из списка. Когда закончите, нажмите на *Применить*, закройте диалог и нажмите на *Обновить* для сохранения изменений на карте сети.

Просмотр

В *Мониторинг → Карты сети* отобразится соответствующий цвет у связи, если триггер перейдет в состояние проблема.



Если несколько триггеров перейдут в состояние проблема, то проблема с наивысшей важностью будет определять цвет и стиль связи. Если несколько триггеров с одинаковой важностью назначены на одну связь карты сети, то триггер с наименьшим ID будет иметь приоритет. Также обратите внимание что:

1. Настройки *Минимальная важность триггеров* и *Отображение подавленных проблем* из конфигурации карты сети влияют какие проблемы принимаются во внимание.
2. В случае триггеров с несколькими проблемами (множественная генерация проблем), у каждой проблемы может быть важность, которая отличается от важности триггера (изменена вручную), могут быть разные теги (из-за макросов) и может быть подавлена.

7.6.3 Комплексные экраны

Обзор

На комплексных экранах в Zabbix вы можете группировать информацию с различных источников для быстрого обзора данных на одном экране. Создание комплексных экранов очень простая и интуитивная задача.

По сути комплексный экран это таблица. Вы выбираете как много в таблице будет ячеек и какие элементы будут отображаться в этих ячейках. Следующие элементы могут быть отображены:

- простые графики
- прототипы простых графиков
- пользовательские графики
- прототипы пользовательских графиков
- карты сети
- другие комплексные экраны
- простая текстовая информация
- информация о сервере (обзор)
- информация о узлах сети (обзор)
- информация о триггере (обзор)
- события у узла сети или в группе узлов сети (состояние триггеров)
- проблемы по важности
- обзор данных
- часы
- история событий
- история недавних действий
- URL (данные полученные с другого места)

Глобальные комплексные экраны управляются в [Мониторинг → Комплексные экраны](#), где их можно настраивать, управлять и просматривать. Их также можно добавлять в раздел избранного на [Мониторинг → ПАНЕЛЬ](#).

[Комплексные экраны уровня узла сети](#) настраиваются на уровне шаблонов и затем генерируются для узлов сети, как только шаблон присоединяется к этим узлам сети.

Для настройки комплексного экрана сначала вы должны создать его, указав ему общие параметры и затем добавив к нему в ячейки индивидуальные элементы.

Все пользователи Zabbix (включая пользователей не администраторов) могут создавать комплексные экраны. Комплексные экраны имеют владельца - пользователя, который создал их.

Комплексные экраны можно сделать публичными или приватными. Публичные комплексные экраны видны всем пользователям.

Приватные комплексные экраны видимы только их владельцам. Владелец может давать общий доступ к приватным комплексным экранам другим пользователям и группам пользователей. Обычные (не Супер администраторы) пользователи могут назначать общий доступ только тем группам и пользователям, в которые они входят сами. Приватные комплексные экраны будут видны своим владельцам и пользователям с общим доступом к этому комплексному экрану так долго, пока они имеют права на чтение всех элементов комплексного экрана. Пользователи уровня Администратора, пока они имеют права на чтение всех элементов комплексного экрана, могут просматривать и редактировать приватные комплексные экраны независимо от того, являются ли они владельцами или входят ли в список пользователей общего доступа.

Как в случае с публичными, так и с приватными комплексными экранами, пользователь должен иметь по крайней мере права на чтение всех элементов комплексного экрана, чтобы его увидеть. Для

добавления элемента на комплексный экран пользователь также должен иметь права по крайней мере на чтение этого элемента.

Создание комплексного экрана

Для создания комплексного экрана, сделайте следующее:

- Перейдите в *Мониторинг* → *Комплексные экраны*
- Перейдите на просмотр всех комплексных экранов
- Нажмите на *Создать комплексный экран*

Вкладка **Комплексный экран** содержит общие атрибуты комплексного экрана:

The screenshot shows the 'Screen' tab of a dialog box. It contains the following fields:

- * Owner: Admin (Zabbix Administrator) with a 'Select' button.
- * Name: Zabbix server.
- * Columns: 2.
- * Rows: 2.
- Buttons: 'Add' and 'Cancel'.

The 'Sharing' tab is also visible at the top.

Все обязательные поля ввода отмечены красной звёздочкой.

Дайте вашему комплексному экрану уникальное имя и укажите количество колонок (вертикальных ячеек) и строк (горизонтальных ячеек).

Вкладка **Общий доступ** содержит как тип комплексного экрана, так и опции общего доступа (группы пользователей, пользователи) для приватных комплексных экранов:

The screenshot shows the 'Sharing' tab of the dialog box. It includes:

- Type: Public (selected).
- List of user group shares:

USER GROUPS	PERMISSIONS	ACTION
Zabbix administrators	Read-only	Read-write (highlighted)

Add
- List of user shares:

USERS	PERMISSIONS	ACTION
user (New User)	Read-only	Read-write (highlighted)

Add
- Buttons: 'Add' and 'Cancel'.

Параметр

Владелец

Тип

Выберите владельца комплексного экрана.

Выберите тип комплексного экрана:

Приватный - комплексный экран видим только выбранным группам пользователей и пользователям

Описание

Параметр

Список групп пользователей участников

Список пользователей участников

Нажмите на Добавить, чтобы сохранить комплексный экран.

Описание

Публичный - комплексный экран видим всем

Выберите группы пользователей, которые будут иметь доступ к этому комплексному экрану.

Вы можете разрешить доступ только на чтение или чтение-запись.

Выберите пользователей, которые будут иметь доступ к этому комплексному экрану.

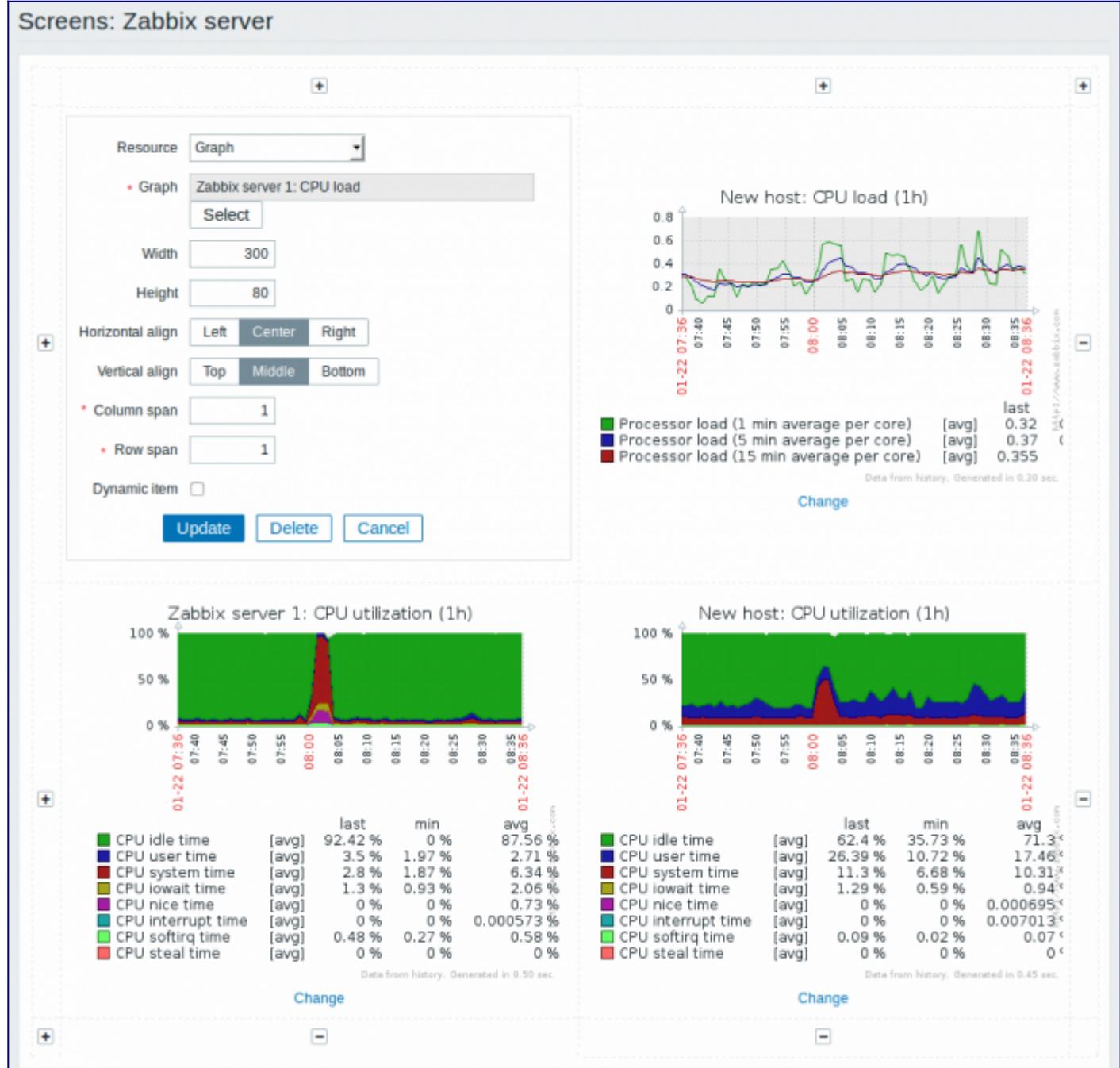
Вы можете разрешить доступ только на чтение или чтение-запись.

Добавление элементов

Для добавления элементов на комплексный экран, нажмите на Конструктор после имени комплексного экрана в списке.

В комплексном экране вы, возможно, увидите только ссылки с текстом Изменить. При нажатии на эти ссылки будет открыт диалог, в котором вы укажите, что отображать в каждой ячейке.

В существующем комплексном экране при нажатии на существующие элементы откроется диалог, в котором вы можете указать, что отображать.



Все обязательные поля ввода отмечены красной звёздочкой.

Атрибуты элементов комплексных экранов:

	Параметр	Описание
		Отображаемая информация в ячейке:
		Журнал действий - история недавних действий
		Часы - цифровые или аналоговые часы, которые показывают время на сервере или местное время
		Обзор данных - последние данные группы узлов сети
		График - один пользовательский график
		Прототип графика - пользовательский график из правила низко-уровневого обнаружения
		История событий - последние события
		События в группах узлов сети - состояния триггеров отфильтрованные по группе узлов сети (включая триггеры без событий, начиная с Zabbix 2.2)
		Информация о узле сети - высокогенерированная информация о узле сети
		События узла сети - состояния триггеров отфильтрованные по узлу сети (включая триггеры без событий, начиная с Zabbix 2.2)
<i>Ресурс</i>		Карта сети - одна карта сети
		Простой текст - простые текстовые данные
		Комплексный экран - комплексный экран (один комплексный экран может содержать другие комплексные экраны)
		Простой график - один простой график
		Прототип простого графика - простой график на основе элемента данных генерированного правилом низко-уровневого обнаружения
		Информация о системе - высокогенерированная информация о Zabbix сервере
		Проблемы по важности - отображение проблем по важности (похоже на ПАНЕЛЬ)
		Информация о триггерах - высокогенерированная информация о триггерах
		Обзор триггеров - состояние триггеров группы узлов сети
		URL - добавление содержимого из внешнего источника
		Смотрите также более подробную информацию о настройке каждого ресурса .
		Возможные значения:
<i>Выравнивание по горизонтали</i>		По центру
		По левой стороне
		По правой стороне
<i>Выравнивание по вертикали</i>		Возможные значения:
		По середине
		По верхнему краю
		По нижнему краю
<i>Объединение столбцов</i>		Расширение ячейки на указанное количество столбцов, работает так же как расширение столбцов в HTML.
<i>Объединение строк</i>		Расширение ячейки на указанное количество строк, работает так же как расширение строк в HTML.
		Обратите внимание на кнопки '+' и '-' с каждой стороны таблицы.
		При нажатии на '+' выше таблицы будет добавлена колонка. При нажатии на '-' ниже таблицы будет удалена колонка.
		При нажатии на '+' с левой стороны таблицы будет добавлена строка. При нажатии на '-' с правой стороны таблицы будет удалена строка.
		Если заданная высота графика менее 120 пикселов, то триггеры не будут отображены в легенде этого графика.

Динамические элементы

Для некоторых элементов имеется дополнительная опция, называемая *Динамический элемент*. При выборе этой опции сначала покажется, что ничего не изменилось.

Однако, если вы перейдете в *Мониторинг → Комплексные экраны*, вы поймете, что сейчас вам доступны дополнительные выпадающие меню для выбора узла сети. Таким образом у вас есть комплексный экран в котором некоторые элементы всегда отображают одну и ту же информацию, тогда как другие отображают информацию в зависимости от выбранного в данный момент узла сети.

Преимуществом этого метода является то, что вам не потребуется создавать дополнительные комплексные экраны только потому, что вы хотите увидеть те же самые графики содержащие данные из разных узлов сети.

Опция *Динамический элемент данных* доступна для нескольких элементов комплексного экрана:

- Графики (пользовательские графики)
- Прототипы графиков
- Простые графики
- Прототипы простых графиков
- Простой текст
- URL

При нажатии на динамический график, откроется этот график в полном виде; однако, в случае с пользовательскими графиками и прототипами графиков эта опция поддерживается только для узла сети выбранного по умолчанию (то есть с 'не выбран' узлом сети из выпадающего меню). При выборе другого узла сети из выпадающего меню, динамический график создается с использованием данных элементов данных выбранного узла сети и результирующий график не является ссылкой.

Динамический элемент URL не будет отображаться в *Мониторинг → Комплексные экраны*, если не выбран узел сети. Если не выбрано ни одного узла сети, то отобразится только сообщение "Не выбран узел сети".

7.6.3.1 Элементы комплексных экранов

Обзор

Этот раздел перечисляет доступные элементы [комплексных экранов](#) и содержит подробную информацию по настройке этих элементов комплексных экранов.

7.6.3.1.1 Журнал действий

В элементе журнала действий вы можете отобразить детали операций действий (оповещения, удаленные команды). Этот элемент является копией информации с *Отчеты → Аудит*.

Для настройки, выберите ресурсом **Журнал действий**:

Resource

* Show lines

Sort entries by

Vertical align Top Middle Bottom

* Column span

* Row span

Все обязательные поля ввода отмечены красной звёздочкой.

Вы можете указать следующие опции:

Отображать строки Задает как много строк из журнала действий будет отображаться в ячейке комплексного экрана.

Сортировка записей по:

Время (по убыванию или по возрастанию)

Тип (по убыванию или по возрастанию)

Состояние (по убыванию или по возрастанию)

Получатель (по убыванию или по возрастанию).

7.6.3.1.2 Часы

В элементе часов вы можете отобразить локальное, серверное или время конкретного узла сети.

Для настройки, выберите ресурсом Часы:

Resource

Time type

Width

Height

Horizontal align Left Center Right

Vertical align Top Middle Bottom

* Column span

* Row span

Вы можете указать следующие опции:

Тип времени Выберите локальное, серверное или время конкретного узла сети.

Элемент данных Выберите элемент данных для отображения времени. Для отображении времени на узле сети, используется элемент данных [system.localtime\[local\]](#). Этот элемент данных должен существовать у узла сети.

Это поле доступно только при выбранном **Время узла сети**.

Ширина Выберите ширину часов.

Высота Выберите высоту часов.

7.6.3.1.3 Обзор данных

В элементе обзора данных вы можете отобразить последние данные группы узлов сети. Этот элемент является копией информации с Мониторинг → Обзор (при выбранном там Типе равном *Данные*).

Для настройки, выберите ресурсом *Обзор данных*:

The dialog shows the following settings:

- Resource: Data overview
- * Group: Discovered hosts
- Application: CPU
- Hosts location: Top (selected)
- Vertical align: Top (selected)
- * Column span: 1
- * Row span: 1
- Add Cancel

Вы можете указать следующие опции:

Группа Выберите группу узлов сети.

Группа элементов данных Введите имя группы элементов данных.

Расположение узлов сети Выберите расположение узлов сети - слева или сверху.

7.6.3.1.4 График

В элементе графика вы можете отобразить один пользовательский график.

Для настройки, выберите ресурсом *График*:

The dialog shows the following settings:

- Resource: Graph
- * Graph: Zabbix server: CPU utilization
- Width: 500
- Height: 100
- Horizontal align: Center (selected)
- Vertical align: Top (selected)
- * Column span: 1
- * Row span: 1
- Dynamic item:
- Add Cancel

Вы можете указать следующие опции:

<i>Имя графика</i>	Выберите отображаемый график.
<i>Ширина</i>	Выберите ширину графика.
<i>Высота</i>	Выберите высоту графика.
<i>Динамический элемент</i>	Укажите графику отображать разные данные в зависимости от выбранного узла сети.

7.6.3.1.5 Прототип графика

В элементе прототипа графика вы можете отобразить пользовательский график из правила низкоуровневого обнаружения.

Для настройки, выберите ресурсом *Прототип графика*:

Resource: Graph prototype

* Graph prototype: Zabbix server: Network traffic on [#IFNAME]

* Max columns: 3

Width: 500

Height: 100

Horizontal align: Left Center Right (Center is selected)

Vertical align: Top Middle Bottom (Top is selected)

* Column span: 1

* Row span: 1

Dynamic item:

Вы можете указать следующие опции:

<i>Имя графика</i>	Выберите отображаемый прототип графика.
<i>Макс колонок</i>	Как много колонок со сгенерированными графиками должно отображаться в ячейке комплексного экрана.
<i>Ширина</i>	Полезно, когда генерируется много графиков из низкоуровневого обнаружения.
<i>Высота</i>	Выберите ширину графика.
<i>Динамический элемент</i>	Выберите высоту графика.
	Укажите графику отображать разные данные в зависимости от выбранного узла сети.

7.6.3.1.6 История событий

В элементе истории событий вы можете отобразить последние события.

Для настройки, выберите ресурсом *История событий*:

Resource

* Show lines

Vertical align Top Middle Bottom

* Column span

* Row span

Вы можете указать следующие опции:

Отображать строки Укажите как много строк событий будут отображаться в ячейке комплексного экрана.

7.6.3.1.7 События в группах узлов сети

В элементе событий в группе узлов сети вы можете отобразить детали проблем отфильтрованных по группе узлов сети.

Цвет важности проблемы отображается на основе цвета важности начального триггера, но его можно скорректировать в экране [обновления проблемы](#).

Для настройки, выберите ресурсом *События в группах узлов сети*:

Resource

Group

* Show lines

Sort triggers by

Vertical align Top Middle Bottom

* Column span

* Row span

Вы можете указать следующие опции:

- | | |
|--------------------------------|---|
| <i>Группа</i> | Выберите группу узлов сети. |
| <i>Отображать строки</i> | Укажите как много строк проблем будут отображаться в ячейке комплексного экрана. |
| <i>Сортировать триггеры по</i> | Выберите из выпадающего списка сортировку проблем по последнему изменению, важности (оба по убыванию) или узлу сети (по возрастанию). |

7.6.3.1.8 Информация об узлах сети

В элементе информации об узлах сети вы можете отобразить высокоуровневую информацию по доступности узлов сети.

Для настройки, выберите ресурсом *Информация об узлах сети*:

The screenshot shows a configuration dialog for a 'Host info' resource. The 'Group' dropdown is set to 'Linux servers'. The 'Style' section has 'Horizontal' selected. Under 'Vertical align', 'Top' is selected. There are two required fields: 'Column span' (value 1) and 'Row span' (value 1). At the bottom are 'Add' and 'Cancel' buttons.

Вы можете указать следующие опции:

Группа Выберите группу(ы) узлов сети.

Стиль Выберите вертикальное или горизонтальное отображение.

7.6.3.1.9 События у узла сети

В элементе событий узла сети вы можете отобразить детали проблем отфильтрованных по узлу сети.

Цвет важности проблемы отображается на основе цвета важности начального триггера, но его можно скорректировать в экране [обновления проблемы](#).

Для настройки, выберите ресурсом *События у узла сети*:

The screenshot shows a configuration dialog for a 'Host issues' resource. The 'Host' dropdown is set to 'New host'. 'Show lines' is set to 25. 'Sort triggers by' is set to 'Last change (descending)'. Under 'Vertical align', 'Middle' is selected. There are two required fields: 'Column span' (value 1) and 'Row span' (value 1). At the bottom are 'Add' and 'Cancel' buttons.

Вы можете указать следующие опции:

Узел сети Выберите узел сети.

Отображать строки Укажите как много строк проблем будут отображаться в ячейке комплексного экрана.

Сортировать триггеры по Выберите из выпадающего списка сортировку проблем по последнему изменению, важности (оба по убыванию) или узлу сети (по возрастанию).

7.6.3.1.10 Карта сети

В элементе карты сети вы можете отобразить настроенную карту сети.

Для настройки, выберите ресурсом *Карта сети*:

Resource

* Map

Horizontal align

Vertical align

* Column span

* Row span

Вы можете указать следующие опции:

Карта сети Выберите отображаемую карту.

7.6.3.1.11 Простой текст

В элементе простого текста вы можете отобразить последние данные по элементу данных как простой текст.

Для настройки, выберите ресурсом *Простой текст*:

Resource

* Item

* Show lines

Show text as HTML

Vertical align

* Column span

* Row span

Dynamic item

Вы можете указать следующие опции:

Элемент данных

Выберите элемент данных.

Отображать строки

Укажите как много последних данных будут отображаться в ячейке комплексного экрана.

Показать текст в виде HTML

Задайте отображение текста HTML кодом.

Динамический элемент

Укажите отображение разных данных в зависимости от выбранного узла сети.

7.6.3.1.12 Комплексный экран

В элементе комплексного экрана вы можете отобразить другой комплексный экран Zabbix. Один комплексный экран может содержать в себе другие комплексные экраны.

Для настройки, выберите ресурсом *Комплексный экран*:

The dialog shows the following settings:

- Resource: Screen
- * Screen: Zabbix server2
- Vertical align: Middle
- * Column span: 1
- * Row span: 1
- Add button (blue)
- Cancel button (white)

Вы можете указать следующие опции:

Комплексный экран Выберите отображаемый комплексный экран.

7.6.3.1.13 Простой график

В элементе простого графика вы можете отобразить один простой график.

Для настройки, выберите ресурсом *Простой график*:

The dialog shows the following settings:

- Resource: Simple graph
- * Item: New host: Incoming network traffic on eth0
- Width: 500
- Height: 100
- Horizontal align: Center
- Vertical align: Middle
- * Column span: 1
- * Row span: 1
- Dynamic item:
- Add button (blue)
- Cancel button (white)

Вы можете указать следующие опции:

Элемент данных	Выберите элемент данных для простого графика.
Ширина	Выберите ширину графика.
Высота	Выберите высоту графика.
Динамический элемент	Укажите графику отображать разные данные в зависимости от выбранного узла сети.

7.6.3.1.14 Прототип простого графика

В элементе прототипа простого графика вы можете отобразить простой график на основе элемента данных сгенерированного низкоуровневым обнаружением.

Для настройки, выберите ресурсом *Прототип простого графика*:

Resource Simple graph prototype

* Item prototype New host: Incoming network traffic on {#IFNAME}

* Max columns 3

Width 500

Height 100

Horizontal align Left Center Right

Vertical align Top Middle Bottom

* Column span 1

* Row span 1

Dynamic item

Вы можете указать следующие опции:

Прототип элемента данных

Выберите прототип элемента данных для простого графика.

Макс колонок

Как много колонок со сгенерированными графиками должно отображаться в ячейке комплексного экрана.

Полезно, когда генерируется много графиков из низкоуровневого обнаружения.

Ширина

Выберите ширину графика.

Высота

Выберите высоту графика.

Динамический элемент

Укажите графику отображать разные данные в зависимости от выбранного узла сети.

7.6.3.1.15 Информация о системе

В элементе информации о системе вы можете отобразить высокоуровневую информацию о Zabbix и Zabbix сервере.

Для настройки, выберите ресурсом *Информация о системе*:

Resource System information

Vertical align Top Middle Bottom

* Column span 1

* Row span 1

7.6.3.1.16 Проблемы по важности

В этом элементе вы можете отобразить проблемы по важности аналогично виджету на панели.

Для настройки, выберите ресурсом *Проблемы по важности*:

Resource	Problems by severity	<input type="button" value="▼"/>
Vertical align	<input type="button" value="Top"/> <input type="button" value="Middle"/> <input type="button" value="Bottom"/>	
* Column span	<input type="text" value="1"/>	
* Row span	<input type="text" value="1"/>	
<input type="button" value="Add"/> <input type="button" value="Cancel"/>		

7.6.3.1.17 Информация о триггерах

В элементе информации о триггерах вы можете отобразить высокоуровневую информацию о статусах триггеров.

Для настройки, выберите ресурсом *Информация о триггерах*:

Resource	Trigger info	<input type="button" value="▼"/>
Group	Linux servers <input type="button" value="X"/>	<input type="button" value="Select"/>
Style	<input type="button" value="Horizontal"/> <input type="button" value="Vertical"/>	
Vertical align	<input type="button" value="Top"/> <input type="button" value="Middle"/> <input type="button" value="Bottom"/>	
* Column span	<input type="text" value="1"/>	
* Row span	<input type="text" value="1"/>	
<input type="button" value="Add"/> <input type="button" value="Cancel"/>		

Вы можете указать следующие опции:

Группа Выберите группу(ы) узлов сети.

Стиль Выберите вертикальное или горизонтальное отображение.

7.6.3.1.18 Обзор триггеров

В элементе обзора триггеров вы можете отобразить состояния триггеров из группы узлов сети. Этот элемент является копией информации с *Мониторинг → Обзор* (при выбранном там Типе равном *Триггеры*).

Для настройки, выберите ресурсом *Обзор триггеров*:

Resource	Trigger overview	<input type="button" value="▼"/>
* Group	Linux servers <input type="button" value="X"/>	<input type="button" value="Select"/>
Application		
Hosts location	<input type="button" value="Left"/> <input type="button" value="Top"/>	
Vertical align	<input type="button" value="Top"/> <input type="button" value="Middle"/> <input type="button" value="Bottom"/>	
* Column span	<input type="text" value="1"/>	
* Row span	<input type="text" value="1"/>	
<input type="button" value="Add"/> <input type="button" value="Cancel"/>		

Вы можете указать следующие опции:

Группа Выберите группу(ы) узлов сети.

Группа элементов данных Введите имя группы элементов данных.

Расположение узлов сети Выберите расположение узлов сети - слева или сверху.

7.6.3.1.19 URL

В элементе URL вы можете отобразить содержимое URL из внешнего источника.

Для настройки, выберите ресурсом *URL*:

The dialog box shows the following settings:

- Resource:** URL
- * URL:** (empty input field)
- Width:** 500
- Height:** 100
- Horizontal align:** Right (selected)
- Vertical align:** Middle (selected)
- * Column span:** 1
- * Row span:** 1
- Dynamic item:**

At the bottom are two buttons: **Add** (blue) and **Cancel**.

Вы можете указать следующие опции:

URL Введите отображаемый URL.

Ширина Выберите ширину окна.

Высота Выберите высоту окна.

Динамический элемент Укажите отображение разного содержимого URL в зависимости от выбранного узла сети.

Браузеры возможно не станут загружать HTTP страницу, которая включена в комплексный экран (при использовании элемента URL), если веб-интерфейс Zabbix доступен через протокол HTTPS.

7.6.4 Слайд-шоу

Обзор

В слайд-шоу вы можете указать какие [комплексных экранов](#) будет отображаться один за другим через заданные интервалы.

Иногда вы можете хотеть переключаться между некоторыми настроенными комплексными экранами. Хотя это можно сделать и вручную, но сделав это более, чем один или два раза, процесс может стать весьма утомительным занятием. Вот здесь функция слайд-шоу приходит к спасению.

Все пользователи Zabbix (включая пользователей не администраторов) могут создавать слайд-шоу. Слайд-шоу имеют владельца - пользователя, который создал их.

Слайд-шоу можно сделать публичными или приватными. Публичные слайд-шоу видимы всем пользователям, однако, они должны иметь по крайней мере права на чтение всех элементов слайд-

шоу, чтобы его увидеть. Для добавления элемента на слайд-шоу пользователь также должен иметь права по крайней мере на чтение этого элемента.

Приватные слайд-шоу видимы только их владельцам. Владелец может давать общий доступ к приватным слайд-шоу другим пользователям и группам пользователей. Обычные (не Супер администраторы) пользователи могут назначать общий доступ только тем группам и пользователям, в которые они входят сами. Приватные слайд-шоу будут видны своим владельцам и пользователям с общим доступом к этому слайд-шоу так долго, пока они имеют права на чтение всех элементов слайд-шоу. Пользователи уровня Администратора, пока они имеют права на чтение всех добавленных комплексных экранов, могут просматривать и редактировать приватные слайд-шоу независимо от того, являются ли они владельцами или входят ли в список пользователей общего доступа.

Настройка

Для создания слайд-шоу сделайте следующее:

- Перейдите в *Мониторинг* → *Комплексные экраны*
- Выберите *Слайд-шоу* в выпадающем меню
- Перейдите к просмотру всех слайд-шоу
- Нажмите на *Создать слайд-шоу*

Вкладка **Слайд** содержит общие атрибуты слайд-шоу:

Slide	Sharing											
<p>* Owner Admin (Zabbix Administrator) <input type="button" value="Select"/></p> <p>* Name Zabbix administrators</p> <p>* Default delay 30s</p> <p>* Slides</p> <table><thead><tr><th></th><th>Delay</th><th>Action</th></tr></thead><tbody><tr><td>Screen</td><td>default</td><td>Remove</td></tr><tr><td>1 Zabbix server</td><td></td><td>Remove</td></tr><tr><td>2 Zabbix server2</td><td>15</td><td>Remove</td></tr></tbody></table> <p>Add</p> <p>Cancel</p>		Delay	Action	Screen	default	Remove	1 Zabbix server		Remove	2 Zabbix server2	15	Remove
	Delay	Action										
Screen	default	Remove										
1 Zabbix server		Remove										
2 Zabbix server2	15	Remove										

Все обязательные поля ввода отмечены красной звёздочкой.

Параметр	Описание
Владелец	Выберите владельца слайд-шоу. Указывать владельца обязательно.
Имя	Уникальное имя слайд-шоу.
Задержка по умолчанию	Как долго будет отображаться каждый комплексный экран по умолчанию, до перехода к следующему. Поддерживаются суффиксы времени , например 30s, 5m, 2h, 1d.
Слайды	Список комплексных экранов, которые ротируются. Нажмите на <i>Добавить</i> , чтобы выбрать комплексные экраны. Стрелки <i>Вверх/Вниз</i> до имени комплексного экрана позволяют перетаскивать комплексный экран выше и ниже для порядка сортировки просмотра. Если вы хотите отображать только, скажем, один график в слайд-шоу, создайте комплексный экран который содержит только один график.
Комплексный экран	Имя комплексного экрана.
Задержка	Пользовательское значение как долго комплексный экран будет отображаться, в секундах.

Параметр

Описание

Если задано значение 0, то будет использоваться значение из Задержка по умолчанию (в сек).

Действие

Нажмите на Удалить для удаления комплексного экрана из слайд-шоу.

Слайд-шоу из этого примера состоит из двух комплексных экранов, которые отображаются в следующем порядке:

Zabbix сервер ⇒ Отображается 30 секунд ⇒ Zabbix сервер2 ⇒ Отображается 15 секунд ⇒ Zabbix сервер ⇒ Отображается 30 секунд ⇒ Zabbix сервер2 ⇒ ...

Вкладка **Общий доступ** содержит как тип слайд-шоу, так и опции общего доступа (группы пользователей, пользователи) для приватных слайд-шоу:

USER GROUPS	PERMISSIONS	ACTION
Linux administrators	Read-only	Remove

USERS	PERMISSIONS	ACTION
guest	Read-only	Remove

Параметр

Описание

Tip

Выберите тип слайд-шоу:

Приватный - слайд-шоу видимо только выбранным группам пользователей и пользователям

Публичный - слайд-шоу видимо всем

Список групп пользователей участников

Выберите группы пользователей, которые будут иметь доступ к этому слайд-шоу.

Вы можете разрешить доступ только на чтение или чтение-запись.

Список пользователей участников

Выберите пользователей, которые будут иметь доступ к этому слайд-шоу.

Вы можете разрешить доступ только на чтение или чтение-запись.

Нажмите на Добавить, чтобы сохранить слайд-шоу.

Просмотр

Готовые слайд-шоу можно просмотреть в Мониторинг → Комплексные экраны, затем выберите Слайд-шоу из выпадающего списка и нажмите на имя слайд-шоу.

Выбрав нужный пункт Меню из выпадающего списка, вы можете ускорить или замедлить отображение, указав множитель задержки обновления:

REFRESH INTERVAL MULTIPLIER

- x0.25
- x0.5
- x1
- x1.5
- x2
- x3
- x4
- x5

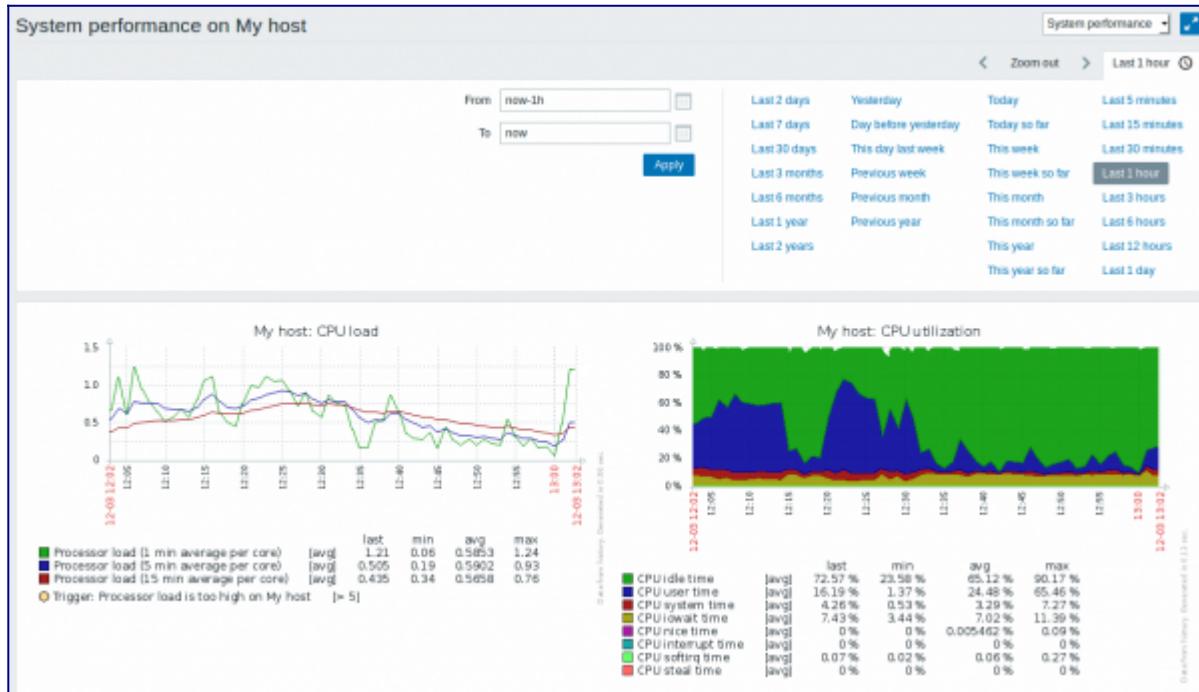
Если задержка менее 5 секунд (либо была указана задержка менее 5 секунд или она стала очень маленькой из-за множителя задержки обновления), тогда будет использоваться минимальная задержка в 5 секунд.

7.6.5 Комплексные экраны узлов сети

Обзор

Комплексные экраны узлов сети выглядят также как и [глобальные комплексные экраны](#), однако, комплексные экраны узлов сети отображают данные только о конкретном узле сети. Комплексные экраны узлов сети настраиваются на уровне [шаблона](#) и затем генерируются для узлов сети, как только шаблон присоединяется к этим узлам сети.

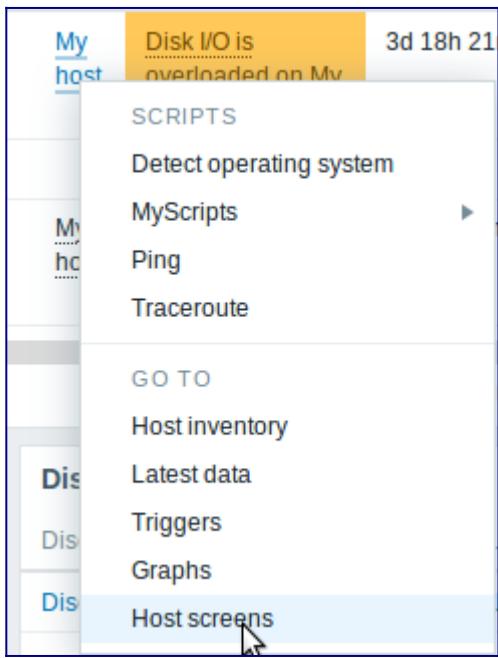
Комплексные экраны узлов сети *нельзя* настраивать или напрямую обращаться через раздел [Мониторинг](#) → [Комплексные экраны](#), который предназначается для глобальных комплексных экранов. Способы доступа к комплексным экранам узлов сети перечислены в этом разделе ниже.



Доступ к комплексным экранам узлов сети

Доступ к комплексным экранам возможен:

- Из [меню узлов сети](#), которое доступно во многих разделах веб-интерфейса:
 - нажмите на имени узла сети и затем выберите *Комплексные экраны узла сети* из всплывающего меню



- При поиске имени узла сети в [глобальном поиске](#):
 - нажмите на ссылку *Комплексные экраны*, указанную в результатах поиска
- При нажатии на имя узла сети в *Инвентаризация* → [Узлы сети](#):
 - нажмите на указанную ссылку *Комплексные экраны*

7.7 Шаблоны

Обзор

Шаблоном является набор объектов, который можно удобно и просто применить к нескольким узлам сети.

Объектами могут быть:

- элементы данных
- триггеры
- графики
- группы элементов данных
- комплексные экраны (*начиная с Zabbix 2.0*)
- правила низкоуровневого обнаружения (*начиная с Zabbix 2.0*)
- веб-сценарии (*начиная с Zabbix 2.2*)

Так как многие узлы сети в реальной жизни идентичны или более менее похожи, отсюда вытекает, что наборы объектов (элементы данных, триггеры, графики,...), созданные вами для одного узла сети, могут быть полезны для многих узлов сети. Конечно, вы можете скопировать их в каждый новый узел сети, но это потребовало бы много ручной работы. Вместо этого, используя шаблоны вы можете скопировать эти объекты в один шаблон и затем применить шаблон для любого количества требуемых узлов сети.

Когда шаблон присоединён к узлу сети, все объекты (элементы данных, триггеры, графики,...) из шаблона добавляются к этому узлу сети. Шаблоны соединяются непосредственно с каждым узлом сети (не к группе узлов сети).

Шаблоны обычно используются для группировки объектов конкретных сервисов или приложений (таких как Apache, MySQL, PostgreSQL, Postfix...) и затем применяются к узлам сети на которых эти сервисы запущены.

Еще одно преимущество в использовании шаблонов, это когда что-либо потребуется изменить на всех узлах сети. Изменение чего-либо на уровне шаблона распространяется на все присоединенные к этому шаблону узлы сети.

Таким образом, использование шаблонов является отличным способом снижения своей нагрузки и рационализации настройки Zabbix.

Перейдем к [созданию и настройке шаблона](#).