

Инструменты по мониторингу инфраструктуры

ZABBIX

Ссылки

- ◆ Официальный сайт
- ◆ Официальная документация
- ◆ Github проекта

Назначение технологии

Zabbix - это программное обеспечение для мониторинга многочисленных параметров сети, жизнеспособности и целостности серверов, виртуальных машин, приложений, сервисов, баз данных, веб-сайтов, облачных сред и многого другого. Он использует гибкий механизм оповещений, что позволяет пользователям настраивать уведомления основанные на e-mail практически на любое событие. Такой подход позволяет быстро реагировать на проблемы с серверами.

Zabbix предлагает отличные функции отчетности и визуализации данных основанные на данных истории. Что делает его идеальным при планировании мощностей.

Все отчеты и статистика, 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.

Архитектура

1. **Zabbix-сервер** — ядро системы, которое дистанционно контролирует сетевые сервисы и является хранилищем, в котором содержатся все конфигурационные, статистические и оперативные данные. Он является тем субъектом в программном обеспечении Zabbix, который оповещает администраторов о проблемах с контролируемым оборудованием.
2. **Zabbix-прокси** собирает данные о производительности и доступности от имени Zabbix-сервера. Все собранные данные заносятся в буфер на локальном уровне и передаются Zabbix-серверу, к которому принадлежит прокси-сервер. Zabbix-прокси является идеальным решением для дистанционного контроля филиалов и других точек, в т.ч. сетей, не имеющих местных администраторов. Он может быть также использован для распределения нагрузки одного Zabbix-сервера. В этом случае, прокси только собирает данные, тем самым на сервер ложится меньшая нагрузка на ЦПУ и на устройства ввода/вывода.
3. **Zabbix-агент** — программа контроля локальных ресурсов и приложений (таких как накопители, оперативная память, статистика процессора и т. д.) на сетевых системах, эти системы должны работать с запущенным Zabbix-агентом.
4. **Веб-интерфейс** — часть Zabbix-сервера, и, как правило (но не обязательно), запускается на том же физическом узле, что и Zabbix-сервер. Работает на PHP, требует веб-сервер (например: NGINX, Apache httpd).

В систему мониторинга уже встроен ряд стандартных метрик:

- нагрузка на процессор, в том числе отдельными процессами
- объём свободной оперативной памяти
- активность жёсткого диска
- объём свободной физической памяти
- сетевая активность
- пинг

А также прочие проверки общего назначения и для самых распространённых сервисов, таких как веб-сервер, СУБД, SSH, Telnet, VMware, NTP, POP, SMTP, FTP и других.

Чтобы задать реакцию при отклонении каких-либо метрик от нормы, используются специальные условия — *триггеры*.

Примеры использования

- ◆ Пинг отсутствует пять минут, выводится уведомление администратору и выполняется команда перезапуска сервиса.
- ◆ Свободного места на жёстком диске осталось меньше 10%, сработает аварийный триггер и чтобы он выключился, значение должно превышать 30%.
- ◆ Проверка доступности веб-сервера, когда система мониторинга имитирует запросы браузера.
- ◆ Сбор желаемых данных с использованием пользовательских интервалов.
- ◆ Имитация нажатия мышкой на веб-сайте, для проверки функционала и времени ответа
- ◆ Визуализация собранных данных с гибкой настройкой их просмотра(графики, карты сети)

Установим MySQL(Если ДБ не установлена)

- `sudo apt update`
- `wget https://dev.mysql.com/get/mysql-apt-config_0.8.22-1_all.deb`
- `dpkg -i mysql-apt-config_0.8.22-1_all.deb`

Подтверждаем установку

- `sudo apt update`
- `sudo apt install mysql-server`

Вводим свой пароль для root'а

Установка сервера и клиента, установление связи между ними.

Установим репозиторий Zabbix

- `wget https://repo.zabbix.com/zabbix/6.0/debian/pool/main/z/zabbix-release/zabbix-release_6.0-1+debian11_all.deb`
- `dpkg -i zabbix-release_6.0-1+debian11_all.deb`
- `apt update`

Установим Zabbix сервер, веб-интерфейс и агент

- `apt install zabbix-server-mysql zabbix-frontend-php zabbix-apache-conf zabbix-sql-scripts zabbix-agent`

Создадим базу данных(Для этого нужна установленная ДБ, в нашем случае MySQL)

- `mysql -uroot -p`
- `%password%`
- `mysql> create database zabbix character set utf8mb4 collate utf8mb4_bin;`
- `mysql> create user zabbix@localhost identified by '%password%';`
- `mysql> grant all privileges on zabbix.* to zabbix@localhost;`
- `mysql> quit;`
- `zcat /usr/share/doc/zabbix-sql-scripts/mysql/server.sql.gz | mysql -uzabbix -p zabbix`

Настроим базу данных для Zabbix. Для этого отредактируем файл `/etc/zabbix/zabbix_server.conf`

```
DBHost=localhost
```

```
DBName=zabbix
```

```
DBUser=zabbix
```

```
DBPassword=%password%
```

Запустим процессы Zabbix сервера и агента

- `systemctl restart zabbix-server zabbix-agent apache2`
- `systemctl enable zabbix-server zabbix-agent apache2`

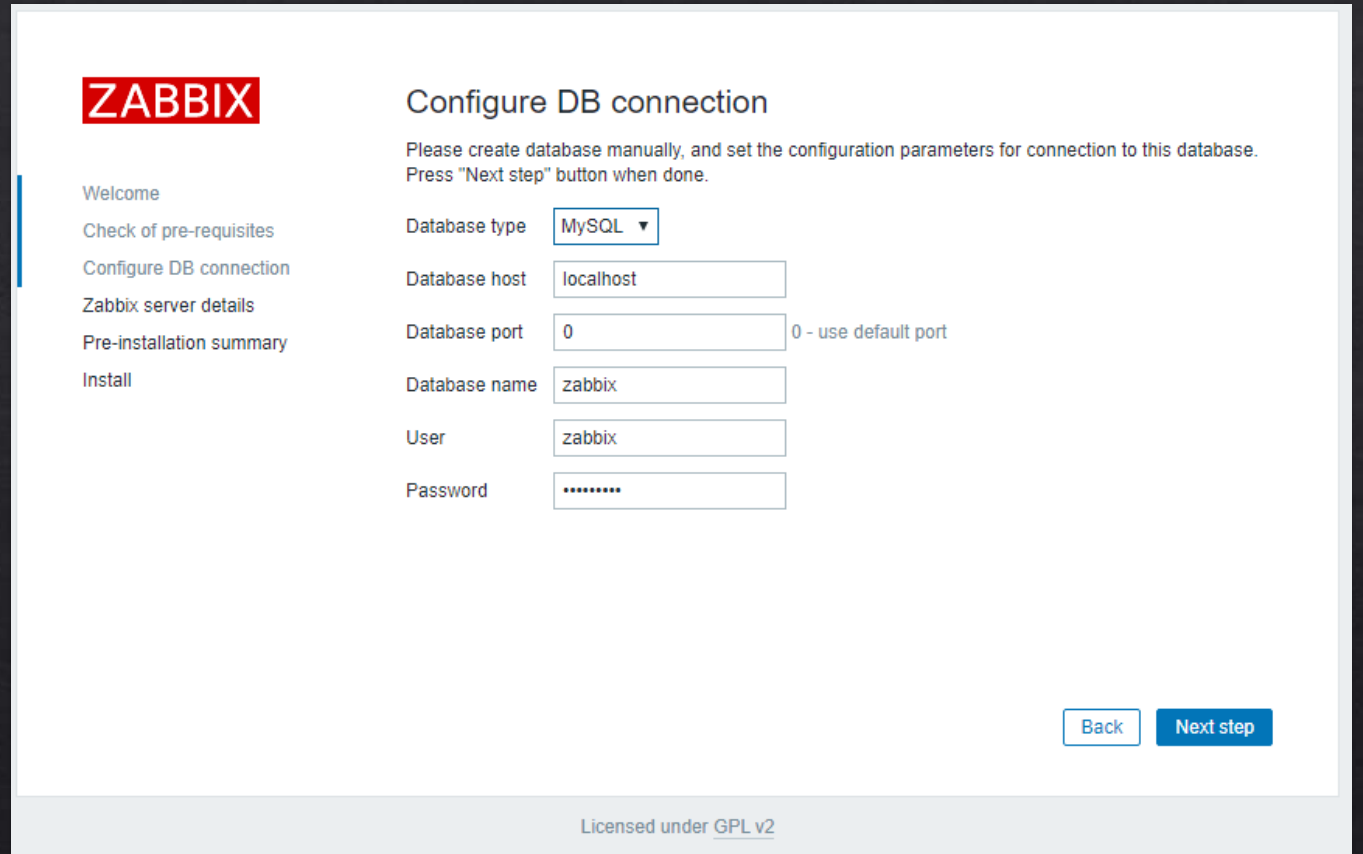
Настроим веб-интерфейс Zabbix

Откройте установленный веб-интерфейс:

```
http://%server_ip_or_name%/zabbix
```

Следуем
инструкциям веб-
интерфейса
используя прежние
данные.

После успешной
установки вас
попросит
залогиниться. По-
умолчанию логин
Admin , пароль *zabbix*



ZABBIX

Welcome
Check of pre-requisites
Configure DB connection
Zabbix server details
Pre-installation summary
Install

Configure DB connection

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

Database type:

Database host:

Database port: 0 - use default port

Database name:

User:

Password:

Licensed under [GPL v2](#)

Установим агента на клиентское устройство.

- `wget https://repo.zabbix.com/zabbix/5.0/debian/pool/main/z/zabbix-release/zabbix-release_5.0-2+debian11_all.deb`
- `sudo dpkg -i zabbix-release_5.0-2+debian11_all.deb`
- `sudo apt update`
- `sudo apt update`
- `sudo apt install zabbix-agent`

Добавим узел на сервере

- Настройка > Узлы сети
- Нажимаем “Создать узел связи” в правом верхнем углу
- Заполняем форму соответствующим образом.

Указываем имя файла.

Добавляем интерфейс Агент, указывая ip нашего клиентского устройства.

Добавляем подходящий шаблон. В нашем случае это “*Linux by Zabbix agent*”

(Можно сузить сбор данных выбрав другой шаблон вроде “*Linux CPU by Zabbix agent*” или “*Linux memory by Zabbix agent*” и другие)

Создадим группу

- Администрирование > Группы пользователей
- Нажимаем “Создать группу пользователей” в правом верхнем углу
- Указываем имя группы и права доступа. В нашем случае Templates/Operating systems на чтение.

Создадим пользователя

- Администрирование > Пользователи
- Нажимаем “Создать пользователя” в правом верхнем углу
- Указываем имя, группу, пароль и права доступа. В нашем случае User role
- Также можно добавить оповещения, выбрав их тип и время, когда данный пользователь свободен.

Настройка средств отображения агрегированной информации.

- ◆ Заходим на созданного пользователя. Это будет наш клиент.
- ◆ Выбираем Мониторинг > Панели > Все панели
- ◆ Пока что у нас есть только “Global view”. Справа сверху выбираем “Создать панель”
- ◆ Нажимая на свободное пространство, выбираем виджет, который хотим добавить на данную панель.
- ◆ Попробуем добавить один. Выбираем Тип “График классический”. Сам график выбираем “CPU usage”
- ◆ Так же добавляем виджеты с графиками “Network Traffic”, “Disc space usage”, “Disk utilization and queue”
- ◆ Справа сверху сохраняем изменения.

Ограничение доступа к системе мониторинга.

- ◆ Заходим на аккаунт Admin
- ◆ Выбираем Администрирование > Роли пользователей
- ◆ Выберем “User role”
- ◆ В блоке Мониторинг отключаем ненужные для данного пользователя возможности. Например: “Последние данные”, “Узлы сети”, “Карты сетей”
- ◆ Внизу страницы жмем “Обновить”.
- ◆ Заходим на аккаунт пользователя с данной ролью
- ◆ Проверяем отсутствие данных возможностей.

Задания

1. Добавьте нового пользователя. И сделайте так, чтобы у него была панель с информацией в виде графиков о размере жесткого диска и загруженности оперативной памяти.
2. Добавьте новый узел сети. Настройте панель с максимально возможной информацией только о CPU данного устройства.
3. Создайте собственный шаблон и примените его для любого подходящего узла сети. По возможности, примените триггеры для отслеживания определённых состояний устройства, которые вы выберете.