

## 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
<a href="#">alpine</a>	3.4
<a href="#">ubuntu</a>	trusty
<a href="#">centos</a>	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 теги:

- 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 <code>mysql-server</code> или <code>postgres-server</code> 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 Прокси 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'.
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 <a href="http://php.net">php.net</a> . 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 <b>zabbix-snmptests</b> экземпляр и разделяемый том <code>/var/lib/zabbix/snmptests</code> с Zabbix сервером или Zabbix прокси.

## Томы

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

Том	Описание
<b>Zabbix агент</b>	
<code>/etc/zabbix/zabbix_agentd.d</code>	Том позволяет включать <code>*.conf</code> файлы и расширять Zabbix агент используя функционал <code>UserParameter</code>
<code>/var/lib/zabbix/modules</code>	Том позволяет загружать дополнительные модули и расширять Zabbix агент используя функционал <a href="#">LoadModule</a>

<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>
<b>Zabbix сервер</b>	
<code>/usr/lib/zabbix/alertscripts</code>	Том используется для пользовательских скриптов оповещения. Том является параметром <code>AlertScriptsPath</code> в <a href="#">zabbix_server.conf</a>
<code>/usr/lib/zabbix/externalscripts</code>	Том используется для <a href="#">внешних проверок</a> . Том является параметром <code>ExternalScripts</code> в <a href="#">zabbix_server.conf</a>
<code>/var/lib/zabbix/modules</code>	Том позволяет загружать дополнительные модули и расширять Zabbix сервер используя функционал <a href="#">LoadModule</a>
<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 <a href="#">zabbix_server.conf</a>
<code>/var/lib/zabbix/ssl/keys</code>	Том используется для размещения файлов частных SSL ключей для аутентификации клиентов. Этот том является аналогом параметра <code>SSLKeyLocation</code> в <a href="#">zabbix_server.conf</a>
<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 <a href="#">zabbix_server.conf</a>
<code>/var/lib/zabbix/snmptraps</code>	Том используется как размещение <code>snmptraps.log</code> файла. Этот том может быть использован совместно 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 используя совместный том и переключив <code>ZBX_ENABLE_SNMP_TRAPS</code> переменную окружения в <code>'true'</code>
<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>
<b>Zabbix прокси</b>	
<code>/usr/lib/zabbix/externalscripts</code>	Том используется для <a href="#">внешних проверок</a> . Том является параметром <code>ExternalScripts</code> в <a href="#">zabbix_proxy.conf</a>
<code>/var/lib/zabbix/modules</code>	Том позволяет загружать дополнительные модули и расширять Zabbix прокси используя функционал <a href="#">LoadModule</a>
<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 <a href="#">zabbix_proxy.conf</a>
<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 <a href="#">zabbix_proxy.conf</a>
<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 <a href="#">zabbix_proxy.conf</a>

<code>/var/lib/zabbix/snmptraps</code>	Том используется как размещение <code>snmptraps.log</code> файла. Этот том может быть использован совместно 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 используя совместный том и переключив <code>ZBX_ENABLE_SNMP_TRAPS</code> переменную окружения в <code>'true'</code>
<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>
<b>Zabbix веб-интерфейс на основе Apache2 веб-сервера</b>	
<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
<b>Zabbix веб-интерфейс на основе Nginx веб-сервера</b>	
<code>/etc/ssl/nginx</code>	Том позволяет активировать HTTPS для Zabbix веб-интерфейса. Этот том должен содержать три файла <code>ssl.crt</code> , <code>ssl.key</code> и <code>dhparam.pem</code> подготовленные для Nginx SSL соединений
<b>Zabbix snmptraps</b>	
<code>/var/lib/zabbix/snmptraps</code>	Том содержит файл журнала с именем <code>snmptraps.log</code> с полученными SNMP трапами
<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>

Для получения дополнительной информации обратитесь к официальным 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 траппера) 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 предоставляет файлы наборов для определения и запуска нескольких контейнеров Zabbix компонент в Docker. Такие файлов наборов доступны в официальной репозитории Zabbix docker на github.com: <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
<code>docker-compose_v3_alpine_mysql_latest.yaml</code>	Файл набора запускает последнюю версию компонент Zabbix 4.0 на Alpine Linux с поддержкой MySQL базы данных.
<code>docker-compose_v3_alpine_mysql_local.yaml</code>	Файл набора собирает локально последнюю версию Zabbix 4.0 и запускает компоненты Zabbix на Alpine Linux с поддержкой MySQL базы данных.
<code>docker-compose_v3_alpine_pgsql_latest.yaml</code>	Файл набора запускает последнюю версию компонент Zabbix 4.0 на Alpine Linux с поддержкой PostgreSQL базы данных.
<code>docker-compose_v3_alpine_pgsql_local.yaml</code>	Файл набора собирает локально последнюю версию Zabbix 4.0 и запускает компоненты Zabbix на Alpine Linux с поддержкой PostgreSQL базы данных.
<code>docker-compose_v3_centos_mysql_latest.yaml</code>	Файл набора запускает последнюю версию компонент Zabbix 4.0 на CentOS 7 с поддержкой MySQL базы данных.
<code>docker-compose_v3_centos_mysql_local.yaml</code>	Файл набора собирает локально последнюю версию Zabbix 4.0 и запускает компоненты Zabbix на CentOS 7 с поддержкой MySQL базы данных.
<code>docker-compose_v3_centos_pgsql_latest.yaml</code>	Файл набора запускает последнюю версию компонент Zabbix 4.0 на CentOS 7 с поддержкой PostgreSQL базы данных.
<code>docker-compose_v3_centos_pgsql_local.yaml</code>	Файл набора собирает локально последнюю версию Zabbix 4.0 и запускает компоненты Zabbix на CentOS 7 с поддержкой PostgreSQL базы данных.
<code>docker-compose_v3_ubuntu_mysql_latest.yaml</code>	Файл набора запускает последнюю версию компонент Zabbix 4.0 на Ubuntu 14.04 с поддержкой MySQL базы данных.
<code>docker-compose_v3_ubuntu_mysql_local.yaml</code>	Файл набора собирает локально последнюю версию Zabbix 4.0 и запускает компоненты Zabbix на Ubuntu 14.04 с поддержкой MySQL базы данных.
<code>docker-compose_v3_ubuntu_pgsql_latest.yaml</code>	Файл набора запускает последнюю версию компонент Zabbix 4.0 на Ubuntu 14.04 с поддержкой PostgreSQL базы данных.



<code>docker-compose_v3_ubuntu_pgsql_local.yaml</code>	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.

From:

<https://www.zabbix.com/documentation/5.2/> - **Zabbix Documentation 5.2**

Permanent link:

<https://www.zabbix.com/documentation/5.2/ru/manual/installation/containers>

Last update: **2019/10/07 06:35**

