

## 5 Installation from containers

### Docker

Zabbix provides [Docker](#) images for each Zabbix component as portable and self-sufficient containers to speed up deployment and update procedure.

Zabbix components come with MySQL and PostgreSQL database support, Apache2 and Nginx web server support. These images are separated into different images.

### Docker base images

Zabbix components are provided on Ubuntu, Alpine Linux and CentOS base images:

Image	Version
<a href="#">alpine</a>	3.4
<a href="#">ubuntu</a>	trusty
<a href="#">centos</a>	latest

All images are configured to rebuild latest images if base images are updated.

Zabbix agent, Zabbix proxy, Zabbix server and Zabbix appliance images are based on Alpine Linux version 3.4, all other images are based on Alpine Linux 3.7.

### Docker file sources

Everyone can follow Docker file changes using the Zabbix [official repository](#) on [github.com](#). You can fork the project or make your own images based on official Docker files.

### Structure

All Zabbix components are available in the following Docker repositories:

- Zabbix appliance with MySQL database support and Nginx web-server - [zabbix/zabbix-appliance](#)
- Zabbix agent - [zabbix/zabbix-agent](#)
- Zabbix server
  - Zabbix server with MySQL database support - [zabbix/zabbix-server-mysql](#)
  - Zabbix server with PostgreSQL database support - [zabbix/zabbix-server-pgsql](#)
- Zabbix web-interface
  - Zabbix web-interface based on Apache2 web server with MySQL database support - [zabbix/zabbix-web-apache-mysql](#)
  - Zabbix web-interface based on Apache2 web server with PostgreSQL database support - [zabbix/zabbix-web-apache-pgsql](#)
  - Zabbix web-interface based on Nginx web server with MySQL database support - [zabbix/zabbix-web-nginx-mysql](#)
  - Zabbix web-interface based on Nginx web server with PostgreSQL database support - [zabbix/zabbix-web-nginx-pgsql](#)

- Zabbix proxy
  - Zabbix proxy with SQLite3 database support - [zabbix/zabbix-proxy-sqlite3](#)
  - Zabbix proxy with MySQL database support - [zabbix/zabbix-proxy-mysql](#)
- Zabbix Java Gateway - [zabbix/zabbix-java-gateway](#)

Additionally there is SNMP trap support. It is provided as additional repository ([zabbix/zabbix-snmptools](#)) based on Ubuntu Trusty only. It could be linked with Zabbix server and Zabbix proxy.

## Versions

Each repository of Zabbix components contains the following tags:

- latest - latest stable version of a Zabbix component based on Alpine Linux image
- alpine-latest - latest stable version of a Zabbix component based on Alpine Linux image
- ubuntu-latest - latest stable version of a Zabbix component based on Ubuntu image
- alpine-4.0-latest - latest minor version of a Zabbix 4.0 component based on Alpine Linux image
- ubuntu-4.0-latest - latest minor version of a Zabbix 4.0 component based on Ubuntu image
- alpine-4.0.\* - different minor versions of a Zabbix 4.0 component based on Alpine Linux image, where \* is the minor version of Zabbix component
- ubuntu-4.0.\* - different minor versions of a Zabbix 4.0 component based on Ubuntu image, where \* is the minor version of Zabbix component

## Usage

### Environment variables

All Zabbix component images provide environment variables to control configuration. These environment variables are listed in each component repository. These environment variables are options from Zabbix configuration files, but with different naming method. For example, `ZBX_LOGSLOWQUERIES` is equal to `LogSlowQueries` from Zabbix server and Zabbix proxy configuration files.

Some of configuration options are not allowed to change. For example, `PIDFile` and `LogType`.

Some of components have specific environment variables, which do not exist in official Zabbix configuration files:

Variable	Components	Description
DB_SERVER_HOST	Server Proxy Web interface	This variable is IP or DNS name of MySQL or PostgreSQL server. By default, value is <code>mysql-server</code> or <code>postgres-server</code> for MySQL or PostgreSQL respectively
DB_SERVER_PORT	Server Proxy Web interface	This variable is port of MySQL or PostgreSQL server. By default, value is <code>'3306'</code> or <code>'5432'</code> respectively.

MYSQL_USER	Server Proxy Web-interface	MySQL database user. By default, value is 'zabbix'.
MYSQL_PASSWORD	Server Proxy Web interface	MySQL database password. By default, value is 'zabbix'.
MYSQL_DATABASE	Server Proxy Web interface	Zabbix database name. By default, value is 'zabbix' for Zabbix server and 'zabbix_proxy' for Zabbix proxy.
POSTGRES_USER	Server Web interface	PostgreSQL database user. By default, value is 'zabbix'.
POSTGRES_PASSWORD	Server Web interface	PostgreSQL database password. By default, value is 'zabbix'.
POSTGRES_DB	Server Web interface	Zabbix database name. By default, value is 'zabbix' for Zabbix server and 'zabbix_proxy' for Zabbix proxy.
PHP_TZ	Web-interface	Timezone in PHP format. Full list of supported timezones are available on <a href="http://php.net">php.net</a> . By default, value is 'Europe/Riga'.
ZBX_SERVER_NAME	Web interface	Visible Zabbix installation name in right top corner of the web interface. By default, value is 'Zabbix Docker'
ZBX_JAVAGATEWAY_ENABLE	Server Proxy	Enables communication with Zabbix Java gateway to collect Java related checks. By default, value is "false"
ZBX_ENABLE_SNMP_TRAPS	Server Proxy	Enables SNMP trap feature. It requires <b>zabbix-snmptests</b> instance and shared volume <code>/var/lib/zabbix/snmptests</code> to Zabbix server or Zabbix proxy.

## Volumes

The images allow to use some mount points. These mount points are different and depend on Zabbix component type:

Volume	Description
<b>Zabbix agent</b>	
<code>/etc/zabbix/zabbix_agentd.d</code>	The volume allows to include <code>*.conf</code> files and extend Zabbix agent using the <code>UserParameter</code> feature
<code>/var/lib/zabbix/modules</code>	The volume allows to load additional modules and extend Zabbix agent using the <code>LoadModule</code> feature
<code>/var/lib/zabbix/enc</code>	The volume is used to store TLS-related files. These file names are specified using <code>ZBX_TLSCAFILE</code> , <code>ZBX_TLSCRLFILE</code> , <code>ZBX_TLSKEY_FILE</code> and <code>ZBX_TLSPSKFILE</code> environment variables
<b>Zabbix server</b>	
<code>/usr/lib/zabbix/alertscripts</code>	The volume is used for custom alert scripts. It is the <code>AlertScriptsPath</code> parameter in <code>zabbix_server.conf</code>
<code>/usr/lib/zabbix/externalscripts</code>	The volume is used by <code>external checks</code> . It is the <code>ExternalScripts</code> parameter in <code>zabbix_server.conf</code>
<code>/var/lib/zabbix/modules</code>	The volume allows to load additional modules and extend Zabbix server using the <code>LoadModule</code> feature

<i>/var/lib/zabbix/enc</i>	The volume is used to store TLS related files. These file names are specified using ZBX_TLSCAFILE, ZBX_TLSCRLFILE, ZBX_TLSKEY_FILE and ZBX_TLSPSKFILE environment variables
<i>/var/lib/zabbix/ssl/certs</i>	The volume is used as location of SSL client certificate files for client authentication. It is the SSLCertLocation parameter in zabbix_server.conf
<i>/var/lib/zabbix/ssl/keys</i>	The volume is used as location of SSL private key files for client authentication. It is the SSLKeyLocation parameter in <a href="#">zabbix_server.conf</a>
<i>/var/lib/zabbix/ssl/ssl_ca</i>	The volume is used as location of certificate authority (CA) files for SSL server certificate verification. It is the SSLCALocation parameter in <a href="#">zabbix_server.conf</a>
<i>/var/lib/zabbix/snmptraps</i>	The volume is used as location of snmptraps.log file. It could be shared by zabbix-snmptraps container and inherited using the volumes_from Docker option while creating a new instance of Zabbix server. SNMP trap processing feature could be enabled by using shared volume and switching the ZBX_ENABLE_SNMP_TRAPS environment variable to 'true'
<i>/var/lib/zabbix/mibs</i>	The volume allows to add new MIB files. It does not support subdirectories, all MIBs must be placed in /var/lib/zabbix/mibs
<b>Zabbix proxy</b>	
<i>/usr/lib/zabbix/externalscripts</i>	The volume is used by <a href="#">external checks</a> . It is the ExternalScripts parameter in <a href="#">zabbix_proxy.conf</a>
<i>/var/lib/zabbix/modules</i>	The volume allows to load additional modules and extend Zabbix server using the <a href="#">LoadModule</a> feature
<i>/var/lib/zabbix/enc</i>	The volume is used to store TLS related files. These file names are specified using ZBX_TLSCAFILE, ZBX_TLSCRLFILE, ZBX_TLSKEY_FILE and ZBX_TLSPSKFILE environment variables
<i>/var/lib/zabbix/ssl/certs</i>	The volume is used as location of SSL client certificate files for client authentication. It is the SSLCertLocation parameter in <a href="#">zabbix_proxy.conf</a>
<i>/var/lib/zabbix/ssl/keys</i>	The volume is used as location of SSL private key files for client authentication. It is the SSLKeyLocation parameter in <a href="#">zabbix_proxy.conf</a>
<i>/var/lib/zabbix/ssl/ssl_ca</i>	The volume is used as location of certificate authority (CA) files for SSL server certificate verification. It is the SSLCALocation parameter in <a href="#">zabbix_proxy.conf</a>
<i>/var/lib/zabbix/snmptraps</i>	The volume is used as location of snmptraps.log file. It could be shared by the zabbix-snmptraps container and inherited using the volumes_from Docker option while creating a new instance of Zabbix server. SNMP trap processing feature could be enabled by using shared volume and switching the ZBX_ENABLE_SNMP_TRAPS environment variable to 'true'
<i>/var/lib/zabbix/mibs</i>	The volume allows to add new MIB files. It does not support subdirectories, all MIBs must be placed in /var/lib/zabbix/mibs
<b>Zabbix web interface based on Apache2 web server</b>	
<i>/etc/ssl/apache2</i>	The volume allows to enable HTTPS for Zabbix web interface. The volume must contain the two ssl.crt and ssl.key files prepared for Apache2 SSL connections
<b>Zabbix web interface based on Nginx web server</b>	

<code>/etc/ssl/nginx</code>	The volume allows to enable HTTPS for Zabbix web interface. The volume must contain the two <code>ssl.crt</code> , <code>ssl.key</code> files and <code>dhparam.pem</code> prepared for Nginx SSL connections
<b>Zabbix snmptraps</b>	
<code>/var/lib/zabbix/snmptraps</code>	The volume contains the <code>snmptraps.log</code> log file named with received SNMP traps
<code>/var/lib/zabbix/mibs</code>	The volume allows to add new MIB files. It does not support subdirectories, all MIBs must be placed in <code>/var/lib/zabbix/mibs</code>

For additional information use Zabbix official repositories in Docker Hub.

## Usage examples

### Example 1

The example demonstrates how to run Zabbix appliance with built-in MySQL database, Zabbix server, Zabbix web interface based on the Nginx web server and Zabbix Java gateway.

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

Zabbix appliance instance exposes 10051/TCP port (Zabbix trapper) and 80/TCP port (HTTP) to host machine.

### Example 2

The example demonstrates how to run Zabbix server with MySQL database support, Zabbix web interface based on the Nginx web server and Zabbix Java gateway.

1. Start empty MySQL server instance

```
# 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. Start Zabbix Java gateway instance

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

3. Start Zabbix server instance and link the instance with created MySQL server instance

```
# 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
```

Zabbix server instance exposes 10051/TCP port (Zabbix trapper) to host machine.

4. Start Zabbix web interface and link the instance with created MySQL server and Zabbix server instances

```
# 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
```

Zabbix web interface instance exposes 80/TCP port (HTTP) to host machine.

### Example 3

The example demonstrates how to run Zabbix server with PostgreSQL database support, Zabbix web interface based on the Nginx web server and SNMP trap feature.

1. Start empty PostgreSQL server instance

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

2. Start Zabbix snmptraps instance

```
# docker run --name zabbix-snmptaps -t \
-v /zbx_instance/snmptaps:/var/lib/zabbix/snmptaps:rw \
-v /var/lib/zabbix/mibs:/usr/share/snmp/mibs:ro \
-p 162:162/udp \
-d zabbix/zabbix-snmptaps:latest
```

Zabbix snmptrap instance exposes the 162/UDP port (SNMP traps) to host machine.

### 3. Start Zabbix server instance and link the instance with created PostgreSQL server instance

```
# 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-snmptests \
  -d zabbix/zabbix-server-pgsql:latest
```

Zabbix server instance exposes the 10051/TCP port (Zabbix trapper) to host machine.

### 4. Start Zabbix web interface and link the instance with created PostgreSQL server and Zabbix server instances

```
# 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
```

Zabbix web interface instance exposes the 443/TCP port (HTTPS) to host machine. Directory */etc/ssl/nginx* must contain certificate with required name.

## Docker Compose

Zabbix provides compose files also for defining and running multi-container Zabbix components in Docker. These compose files are available in Zabbix docker official repository on github.com: <https://github.com/zabbix/zabbix-docker>. These compose files are added as examples, they are overloaded. For example, they contain proxies with MySQL and SQLite3 support.

There are a few different versions of compose files:

File name	Description
<code>docker-compose_v3_alpine_mysql_latest.yaml</code>	The compose file runs the latest version of Zabbix 4.0 components on Alpine Linux with MySQL database support.
<code>docker-compose_v3_alpine_mysql_local.yaml</code>	The compose file locally builds the latest version of Zabbix 4.0 and runs Zabbix components on Alpine Linux with MySQL database support.

<code>docker-compose_v3_alpine_pgsql_latest.yaml</code>	The compose file runs the latest version of Zabbix 4.0 components on Alpine Linux with PostgreSQL database support.
<code>docker-compose_v3_alpine_pgsql_local.yaml</code>	The compose file locally builds the latest version of Zabbix 4.0 and runs Zabbix components on Alpine Linux with PostgreSQL database support.
<code>docker-compose_v3_centos_mysql_latest.yaml</code>	The compose file runs the latest version of Zabbix 4.0 components on CentOS 7 with MySQL database support.
<code>docker-compose_v3_centos_mysql_local.yaml</code>	The compose file locally builds the latest version of Zabbix 4.0 and runs Zabbix components on CentOS 7 with MySQL database support.
<code>docker-compose_v3_centos_pgsql_latest.yaml</code>	The compose file runs the latest version of Zabbix 4.0 components on CentOS 7 with PostgreSQL database support.
<code>docker-compose_v3_centos_pgsql_local.yaml</code>	The compose file locally builds the latest version of Zabbix 4.0 and runs Zabbix components on CentOS 7 with PostgreSQL database support.
<code>docker-compose_v3_ubuntu_mysql_latest.yaml</code>	The compose file runs the latest version of Zabbix 4.0 components on Ubuntu 14.04 with MySQL database support.
<code>docker-compose_v3_ubuntu_mysql_local.yaml</code>	The compose file locally builds the latest version of Zabbix 4.0 and runs Zabbix components on Ubuntu 14.04 with MySQL database support.
<code>docker-compose_v3_ubuntu_pgsql_latest.yaml</code>	The compose file runs the latest version of Zabbix 4.0 components on Ubuntu 14.04 with PostgreSQL database support.
<code>docker-compose_v3_ubuntu_pgsql_local.yaml</code>	The compose file locally builds the latest version of Zabbix 4.0 and runs Zabbix components on Ubuntu 14.04 with PostgreSQL database support.

Available Docker compose files support both versions 2 and 3 of Docker Compose.

## Storage

Compose files are configured to support local storage on a host machine. Docker Compose will create a `zbx_env` directory in the folder with the compose file when you run Zabbix components using the compose file. The directory will contain the same structure as described above in the [Volumes](#) section and directory for database storage.

There are also volumes in read-only mode for `/etc/localtime` and `/etc/timezone` files.

## Environment files

In the same directory with compose files on [github.com](https://github.com) you can find files with default environment variables for each component in compose file. These environment files are named like `.env_<type`



of component>.

## Examples

### Example 1

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

The command will download latest Zabbix 4.0 images for each Zabbix component and run them in detach mode.

Do not forget to download `.env_<type of component>` files from github.com official Zabbix repository with compose files.

### Example 2

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

The command will download base image Ubuntu 14.04, then build Zabbix 4.0 components locally and run them in detach mode.

From:

<https://www.zabbix.com/documentation/4.2/> - **Zabbix Documentation 4.2**

Permanent link:

<https://www.zabbix.com/documentation/4.2/manual/installation/containers>

Last update: **2019/04/09 20:52**

