

5 从容器中安装

Docker

Zabbix 为每个组件都提供了 [Docker](#) 镜像，作为弹性和自给自足的容器，促使加快部署和更新过程。

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

Zabbix 组件支持 MySQL 和 PostgreSQL 数据库、Apache2 和 Nginx Web 服务器。这些镜像被分成多个不同的镜像。

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

Docker 的基础镜像

Zabbix 组件提供了 Ubuntu、Alpine Linux 和 CentOS 的基础镜像：

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

镜像	版本
alpine	3.4
ubuntu	trusty
centos	latest

如果基础镜像升级了，所有的镜像被配置为重建成最新版本的镜像。

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

Zabbix agent、Zabbix proxy、Zabbix server 和 Zabbix 应用镜像均基于 Alpine Linux 3.4 版，所有其他映像均基于 Alpine Linux 3.7。

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 源文件

每个人都可以在 [github.com](#) 上使用 Zabbix [官方镜像仓库](#)，并关注其 Docker 文件变更情况。您可以根据官方 Docker 文件复制此项目或制作自己的镜像。

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.

组件

所有 Zabbix 组件都可在以下 Docker 镜像仓库中使用：

All Zabbix components are available in the following Docker repositories:

- MySQL 数据库和 Nginx Web 服务器支持的 Zabbix 应用 - [zabbix/zabbix-appliance](#)
- Zabbix appliance with MySQL database support and Nginx web-server - [zabbix/zabbix-appliance](#)
- Zabbix agent - [zabbix/zabbix-agent](#)
- Zabbix server
 - MySQL 数据库支持的 Zabbix server - [zabbix/zabbix-server-mysql](#)
 - PostgreSQL 数据库支持的 Zabbix server - [zabbix/zabbix-server-pgsql](#)
- 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
 - 基于 Apache2 Web 服务器以及支持 MySQL 数据库的 Zabbix web 接口 - [zabbix/zabbix-web-apache-mysql](#)
 - 基于 Apache2 Web 服务器以及支持 PostgreSQL 数据库的 Zabbix web 接口 - [zabbix/zabbix-web-apache-pgsql](#)
 - 基于 Nginx Web 服务器以及支持 MySQL 数据库的 Zabbix web 接口 - [zabbix/zabbix-web-nginx-mysql](#)
 - 基于 Nginx Web 服务器以及支持 PostgreSQL 数据库的 Zabbix web 接口 - [zabbix/zabbix-web-nginx-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
 - SQLite3 数据库支持的 Zabbix proxy - [zabbix/zabbix-proxy-sqlite3](#)
 - MySQL 数据库支持的 Zabbix proxy - [zabbix/zabbix-proxy-mysql](#)
- 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](#)

此外，对于 SNMP trap 的支持，它仅作为基于 Ubuntu Trusty 的额外镜像仓库 ([zabbix/zabbix-snmptaps](#)) 提供。它可以与 Zabbix server 和 Zabbix proxy 关联。

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

版本

Zabbix 组件的每个镜像仓库都包含了下列标签：

- latest - 基于 Alpine Linux 镜像的最新稳定版的 Zabbix 组件；
- alpine-latest - 基于 Alpine Linux 镜像的最新稳定版的 Zabbix 组件[]latest stable version of a Zabbix component based on Alpine Linux image
- ubuntu-latest - 基于 Ubuntu 镜像的最新稳定版的 Zabbix 组件[]latest stable version of a Zabbix component based on Ubuntu image

- `alpine-4.0-latest` - 基于 Alpine Linux 镜像的最新次要版本的 Zabbix 4.0 组件 [latest minor version of a Zabbix 4.0 component based on Alpine Linux image]
- `ubuntu-4.0-latest` - 基于 Ubuntu 镜像的最新次要版本的 Zabbix 4.0 组件 [latest minor version of a Zabbix 4.0 component based on Ubuntu image]
- `alpine-4.0.*` - 基于 Alpine Linux 镜像的不同次要版本的 Zabbix 4.0 组件，其中 * 代表 Zabbix 组件的次要版本；
- `ubuntu-4.0.*` - 基于 Ubuntu 镜像的不同次要版本的 Zabbix 4.0 组件，其中 * 代表 Zabbix 组件的次要版本。

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

使用方法

环境变量

所有 Zabbix 组件镜像都提供环境变量来控制配置。 这些环境变量在每个组件镜像仓库中列出。这些环境变量是 Zabbix 配置文件中的选项，但具有不同的命名方法。 例如，`ZBX_LOGSLOWQUERIES` 等于来自 Zabbix server 和 Zabbix proxy 配置文件的 `LogSlowQueries`

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.

一些配置选项是不允许更改的。例如，`PIDFile` 和 `LogType`
Some of configuration options are not allowed to change. For example, `PIDFile` and `LogType`.

其中，一些组件有特定的环境变量，而这些环境变量在官方 Zabbix 配置文件并不存在：

变量	组件	描述
<code>DB_SERVER_HOST</code>	Server Proxy Web interface	这个变量指的是 MySQL 或 PostgreSQL 的 IP 或 DNS [默认情况下，这个值根据 MySQL 和 PostgreSQL [分别为 <code>mysql-server</code> 或 <code>postgres-server</code>
<code>DB_SERVER_PORT</code>	Server Proxy Web interface	这个变量指的是 MySQL 或 PostgreSQL 的端口。默认情况下，这个值根据 MySQL 和 PostgreSQL [分别为 <code>'3306'</code> 或 <code>'5432'</code> 。

MYSQL_USER	Server Proxy Web-interface	MySQL 数据库用户。 默认情况下，这个值为 'zabbix' []
MYSQL_PASSWORD	Server Proxy Web interface	MySQL 数据库密码。 默认情况下，这个值为 'zabbix' []
MYSQL_DATABASE	Server Proxy Web interface	Zabbix 数据库库名。 默认情况下，这个值根据 Zabbix server 和 Zabbix proxy [] 分别为 'zabbix' 和 'zabbix_proxy' []
POSTGRES_USER	Server Web interface	PostgreSQL 数据库用户。 默认情况下，这个值为 'zabbix' []
POSTGRES_PASSWORD	Server Web interface	PostgreSQL 数据库密码。 默认情况下，这个值为 'zabbix' []
POSTGRES_DB	Server Web interface	Zabbix 数据库库名。 默认情况下，这个值根据 Zabbix server 和 Zabbix proxy [] 分别为 'zabbix' 和 'zabbix_proxy' []
TZ	Web-interface	PHP 时区格式。所有支持的时区列表为 php.net 。 默认情况下，这个值为 'Europe/Riga' []
ZBX_SERVER_NAME	Web interface	Web 界面右上角显示的安装名称。 默认情况下，这个值为 'Zabbix Docker' []
ZBX_JAVAGATEWAY_ENABLE	Server Proxy	是否启用 Zabbix Java gateway 以采集与 Java 相关的检查数据。 默认情况下，这个值为 "false" []
ZBX_ENABLE_SNMP_TRAPS	Server Proxy	是否启用 SNMP trap feature 功能。这要求 zabbix-snmptaps 实例并共享 <code>/var/lib/zabbix/snmptaps</code> 卷到 Zabbix server 或 proxy []

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 '3306' or '5432' 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.
TZ	Web-interface	Timezone in PHP format. Full list of supported timezones are available on php.net . 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 zabbix-snmptests instance and shared volume <code>/var/lib/zabbix/snmptests</code> to Zabbix server or Zabbix proxy.

卷

镜像中允许使用一些挂载点。根据 Zabbix 组件类型，这些挂载点各不相同：

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

卷	描述
Zabbix agent	
<code>/etc/zabbix/zabbix_agentd.d</code>	该卷允许包含 <code>*.conf</code> 文件并使用 <code>UserParameter</code> 功能扩展 Zabbix agent
<code>/var/lib/zabbix/modules</code>	该卷允许通过 <code>LoadModule</code> 功能加载额外的模块以扩展 Zabbix agent
<code>/var/lib/zabbix/enc</code>	该卷用于存放 TLS 相关的文件。这些文件名指定使用 <code>ZBX_TLSCAFILE</code> 、 <code>ZBX_TLSCRLFILE</code> 、 <code>ZBX_TLSKEY_FILE</code> 和 <code>ZBX_TLSPSKFILE</code> 环境变量。
Zabbix server	
<code>/usr/lib/zabbix/alertscripts</code>	该卷用于自定义告警脚本。即在 <code>zabbix_server.conf</code> 中的 <code>AlertScriptsPath</code> 参数。
<code>/usr/lib/zabbix/externalscripts</code>	该卷用于 <code>外部检查</code> 。即在 <code>zabbix_server.conf</code> 中的 <code>ExternalScripts</code> 参数。
<code>/var/lib/zabbix/modules</code>	该卷允许通过 <code>LoadModule</code> 功能加载额外的模块以扩展 Zabbix agent
<code>/var/lib/zabbix/enc</code>	该卷用于存放 TLS 相关的文件。这些文件名指定使用 <code>ZBX_TLSCAFILE</code> 、 <code>ZBX_TLSCRLFILE</code> 、 <code>ZBX_TLSKEY_FILE</code> 和 <code>ZBX_TLSPSKFILE</code> 环境变量。
<code>/var/lib/zabbix/ssl/certs</code>	该卷用于存放客户端认证的 SSL 客户端认证文件。即在 <code>zabbix_server.conf</code> 中的 <code>SSLCertLocation</code> 参数。
<code>/var/lib/zabbix/ssl/keys</code>	该卷用于存放客户端认证的 SSL 私钥文件。即在 <code>zabbix_server.conf</code> 中的 <code>SSLKeyLocation</code> 参数。
<code>/var/lib/zabbix/ssl/ssl_ca</code>	该卷用于存放 SSL 服务器证书认证的证书颁发机构(CA)文件。即在 <code>zabbix_server.conf</code> 中的 <code>SSLCALocation</code> 参数。
<code>/var/lib/zabbix/snmptests</code>	该卷用于存放 <code>snmptests.log</code> 文件。它可由 <code>zabbix-snmptests</code> 容器共享，并在创建 Zabbix server 新实例时使用 Docker 的 <code>volumes_from</code> 选项继承。可以通过共享卷，并将 <code>ZBX_ENABLE_SNMP_TRAPS</code> 环境变量切换为 'true' 以启用 SNMP trap 处理功能。

<code>/var/lib/zabbix/mibs</code>	该卷允许添加新的 MIB 文件。它不支持子目录，所有的 MIB 文件必须位于 <code>/var/lib/zabbix/mibs</code> 下。
Zabbix proxy	
<code>/usr/lib/zabbix/externalscripts</code>	该卷用于使用 外部检查 。即在 <code>zabbix_proxy.conf</code> 中的 <code>ExternalScripts</code> 参数。
<code>/var/lib/zabbix/modules</code>	该卷允许通过 LoadModule 功能加载额外的模块以扩展 Zabbix server
<code>/var/lib/zabbix/enc</code>	该卷用于存放 TLS 相关的文件。这些文件名指定使用 <code>ZBX_TLSCAFILE</code> 、 <code>ZBX_TLSCRLFILE</code> 、 <code>ZBX_TLSKEY_FILE</code> 和 <code>ZBX_TLSPSKFILE</code> 环境变量。
<code>/var/lib/zabbix/ssl/certs</code>	该卷用于存放客户端认证的 SSL 客户端认证文件。即在 <code>zabbix_proxy.conf</code> 中的 <code>SSLCertLocation</code> 参数。
<code>/var/lib/zabbix/ssl/keys</code>	该卷用于存放客户端认证的 SSL 私钥文件。即在 <code>zabbix_proxy.conf</code> 中的 <code>SSLKeyLocation</code> 参数。
<code>/var/lib/zabbix/ssl/ssl_ca</code>	该卷用于存放 SSL 服务器证书认证的证书颁发机构(CA)文件。即在 <code>zabbix_proxy.conf</code> 中的 <code>SSLCALocation</code> 参数。
<code>/var/lib/zabbix/snmptraps</code>	该卷用于存放 <code>snmptraps.log</code> 文件。它可由 <code>zabbix-snmptraps</code> 容器共享，并在创建 Zabbix server 新实例时使用 Docker 的 <code>volumes_from</code> 选项继承。可以通过共享卷，并将 <code>ZBX_ENABLE_SNMP_TRAPS</code> 环境变量切换为 'true' 以启用 SNMP trap 处理功能。
<code>/var/lib/zabbix/mibs</code>	该卷允许添加新的 MIB 文件。它不支持子目录，所有的 MIB 文件必须位于 <code>/var/lib/zabbix/mibs</code> 下。
基于 Apache2 Web 服务器的 Zabbix Web 接口	
<code>/etc/ssl/apache2</code>	该卷允许为 Zabbix Web 接口启用 HTTPS。该卷必须包含为 Apache2 SSL 连接准备的 <code>ssl.crt</code> 和 <code>ssl.key</code> 两个文件。
基于 Nginx Web 服务器的 Zabbix Web 接口	
<code>/etc/ssl/nginx</code>	该卷允许为 Zabbix Web 接口启用 HTTPS。该卷必须包含为 Nginx SSL 连接装备的 <code>ssl.crt</code> 和 <code>ssl.key</code> 两个文件。
Zabbix snmptraps	
<code>/var/lib/zabbix/snmptraps</code>	该卷包含了以接收到的 SNMP traps 命名的 <code>snmptraps.log</code> 日志文件。
<code>/var/lib/zabbix/mibs</code>	该卷允许添加新的 MIB 文件。它不支持子目录，该 MIB 文件必须位于 <code>/var/lib/zabbix/mibs</code> 下。

关于更多的信息请在 Docker Hub 的 Zabbix 官方镜像仓库查看。

For additional information use Zabbix official repositories in Docker Hub.

使用方法实例

示例 1

该示例示范了如何使用内置 MySQL 数据库、Zabbix server、基于 Nginx Web 服务器的 Zabbix Web 界面和 Zabbix Java gateway 来运行 Zabbix 应用。

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 应用实例将 10051/TCP 端口 Zabbix trapper 和 80/TCP 端口 HTTP 暴露给主机。
Zabbix appliance instance exposes 10051/TCP port (Zabbix trapper) and 80/TCP port (HTTP) to host machine.

示例 2

该示例示范了如何运行 MySQL 数据库支持、基于 Nginx Web 服务器的 Zabbix Web 界面和 Zabbix Java gateway

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. 首先，启动空的 MySQL 服务器实例。

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. 其次，启动 Zabbix Java gateway 实例

2. Start Zabbix Java gateway instance

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

3. 然后，启动 Zabbix server 实例，并将其关联到已创建的 MySQL server 实例。

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 实例将 10051/TCP 端口 Zabbix trapper 暴露给主机。
Zabbix server instance exposes 10051/TCP port (Zabbix trapper) to host machine.

4. 最后，启动 Zabbix Web 界面，并将其关联到已创建的 MySQL server 和 Zabbix server 实例。

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 界面实例将 80/TCP 端口[HTTP]暴露给主机。

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

示例 3

该示例示范了如何运行 PostgreSQL 数据库支持的 Zabbix server[基于 Nginx Web 服务器的 Zabbix Web 界面和 SNMP trap 功能。

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. 首先，启动空的 PostgreSQL server 实例。

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. 其次，启动 Zabbix snmptraps 实例。

2. Start Zabbix snmptraps instance

```
# 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
```

Zabbix snmptrap 实例将 162/UDP 端口[SNMP traps]暴露给主机。

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

3. 然后，启动 Zabbix server 实例，并将其关联到已创建的 PostgreSQL server 实例。

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 实例将 10051/TCP 端口[]Zabbix trapper[]暴露给主机。

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

4. 最后，启动 Zabbix Web 界面，并将其关联到已创建的 PostgreSQL server 和 Zabbix server 实例。

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 界面实例将 443/TCP 端口[]HTTPS[]暴露给主机。

/etc/ssl/nginx 目录必须包含具有所需名称的证书。

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 为 Docker 提供了定义和运行复杂 Zabbix 组件的 compose 文件。这些 compose 文件可以在 github.com: <https://github.com/zabbix/zabbix-docker> 上的 Zabbix docker 官方镜像仓库中找到。这些 compose 文件作为示例添加，并支持广泛。例如[]Zabbix proxy 支持 MySQL 和 SQLite3[]

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.

以下为几个不同版本的 compose 文件：

There are a few different versions of compose files:

文件名	描述
<code>docker-compose_v3_alpine_mysql_latest.yaml</code>	该 compose 文件运行基于 Alpine Linux 的 Zabbix 4.0 最新版本的组件，支持 MySQL 数据库。
<code>docker-compose_v3_alpine_mysql_local.yaml</code>	该 compose 文件本地构建和运行基于 Alpine Linux 的 Zabbix 4.0 最新版本的组件，支持 MySQL 数据库。
<code>docker-compose_v3_alpine_pgsql_latest.yaml</code>	该 compose 文件运行基于 Alpine Linux 的 Zabbix 4.0 最新版本的组件，支持 PostgreSQL 数据库。
<code>docker-compose_v3_alpine_pgsql_local.yaml</code>	该 compose 文件本地构建和运行基于 Alpine Linux 的 Zabbix 4.0 最新版本的组件，支持 PostgreSQL 数据库。
<code>docker-compose_v3_centos_mysql_latest.yaml</code>	该 compose 文件运行基于 CentOS7 的 Zabbix 4.0 最新版本的组件，支持 MySQL 数据库。
<code>docker-compose_v3_centos_mysql_local.yaml</code>	该 compose 文件本地构建和运行基于 CentOS7 的 Zabbix 4.0 最新版本的组件，支持 MySQL 数据库。
<code>docker-compose_v3_centos_pgsql_latest.yaml</code>	该 compose 文件运行基于 CentOS7 的 Zabbix 4.0 最新版本的组件，支持 PostgreSQL 数据库。
<code>docker-compose_v3_centos_pgsql_local.yaml</code>	该 compose 文件本地构建和运行基于 CentOS7 的 Zabbix 4.0 最新版本的组件，支持 PostgreSQL 数据库。
<code>docker-compose_v3_ubuntu_mysql_latest.yaml</code>	该 compose 文件运行基于 Ubuntu 14.04 的 Zabbix 4.0 最新版本的组件，支持 MySQL 数据库。
<code>docker-compose_v3_ubuntu_mysql_local.yaml</code>	该 compose 文件本地构建和运行基于 Ubuntu 14.04 的 Zabbix 4.0 最新版本的组件，支持 MySQL 数据库。
<code>docker-compose_v3_ubuntu_pgsql_latest.yaml</code>	该 compose 文件运行基于 Ubuntu 14.04 的 Zabbix 4.0 最新版本的组件，支持 PostgreSQL 数据库。
<code>docker-compose_v3_ubuntu_pgsql_local.yaml</code>	该 compose 文件本地构建和运行基于 Ubuntu 14.04 的 Zabbix 4.0 最新版本的组件，支持 PostgreSQL 数据库。

Docker compose 文件支持 Docker Compose 2 版本和 Docker Compose 3 版本。
Available Docker compose files support both versions 2 and 3 of Docker Compose.

存储

Compose 文件已经配置支持主机上的存储。当你使用 Compose 文件运行 Zabbix 组件时 Docker Compose 将在其所在文件夹中创建一个 `zbx_env` 目录，该目录将包含于 [Volumes](#) 章节所述相同的结构，以用于数据库存储。

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.

此外，还有卷 `/etc/localtime` 和 `/etc/timezone` 下的文件为只读模式。

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

环境变量文件

在 `github.com` 上与存放 `compose` 文件的同一目录中，您可以在 `compose` 文件中找到每个组件的默认环境变量文件，这些环境变量文件的命令与 `.env_<type of component>` 类似。

In the same directory with `compose` files on `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>`.

示例

示例 1

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

这个命令将会为每个 Zabbix 组件下载最新的 Zabbix 4.0 镜像，并以 `detach` 模式运行。

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

不要忘记从 `github.com` 的 Zabbix 官方镜像仓库下载 `.env_<type of component>` 文件和 `compose` 文件。

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

示例 2

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

这个命令将会下载基于 Ubuntu 14.04 的镜像，并在本地构建 Zabbix 4.0 组件，以 `detach` 模式运行。

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/4.0/> - **Zabbix Documentation 4.0**

Permanent link:

<https://www.zabbix.com/documentation/4.0/zh/manual/installation/containers>

Last update: **2018/08/27 11:45**

