

5 Installation depuis les containers

Docker

Zabbix fournit des images [Docker](#) pour chaque composant Zabbix en tant que conteneurs portables et auto-suffisants pour accélérer le déploiement et mettre à jour la procédure.

Les composants Zabbix sont fournis avec la prise en charge de base de données MySQL et PostgreSQL, le support des serveurs web Apache2 et Nginx. Ces images sont séparées en différentes images.

Images de base Docker

Les composants Zabbix sont fournis sur les images de base Ubuntu, Alpine Linux et CentOS :

Image	Version
alpine	3.4
ubuntu	sûre
centos	dernière

Toutes les images sont configurées pour reconstruire les dernières images si les images de base sont mises à jour.

Les images de l'agent Zabbix, du proxy Zabbix, du serveur Zabbix et de l'appliance Zabbix sont basées sur Linux Alpine version 3.4, toutes les autres images sont basées sur Linux Alpine 3.7.

Sources des fichiers Docker

Tout le monde peut suivre les modifications du fichier Docker en utilisant le [dépôt officiel](#) de Zabbix sur [github.com](#). Vous pouvez faire un fork du projet ou créer vos propres images basées sur les fichiers Docker officiels.

Structure

Tous les composants Zabbix sont disponibles dans les dépôts Docker suivants :

- Appliance Zabbix avec prise en charge de base de données MySQL et serveur Web Nginx - [zabbix/zabbix-appliance](#)
- Agent Zabbix - [zabbix/zabbix-agent](#)
- Zabbix server
 - Serveur Zabbix avec prise en charge de base de données MySQL - [zabbix/zabbix-server-mysql](#)
 - Serveur Zabbix avec prise en charge de base de données PostgreSQL - [zabbix/zabbix-server-pgsql](#)
- Interface Web Zabbix
 - Interface Web Zabbix basée sur un serveur web Apache2 avec prise en charge de base de données MySQL - [zabbix/zabbix-web-apache-mysql](#)

- Interface Web Zabbix basée sur un serveur web Apache2 avec prise en charge de base de données PostgreSQL - [zabbix/zabbix-web-apache-pgsql](#)
- Interface Web Zabbix basée sur un serveur web Nginx avec prise en charge de base de données MySQL - [zabbix/zabbix-web-nginx-mysql](#)
- Interface Web Zabbix basée sur un serveur web Nginx avec prise en charge de base de données PostgreSQL - [zabbix/zabbix-web-nginx-pgsql](#)
- Proxy Zabbix
 - Proxy Zabbix avec prise en charge de base de données SQLite3 - [zabbix/zabbix-proxy-sqlite3](#)
 - Proxy Zabbix avec prise en charge de base de données MySQL - [zabbix/zabbix-proxy-mysql](#)
- Passerelle Java Zabbix - [zabbix/zabbix-java-gateway](#)

De plus, il existe une prise en charge des trappes SNMP. Il est fourni en tant que dépôt supplémentaire ([zabbix/zabbix-snmptraps](#)) basé uniquement sur Ubuntu Trusty. Il peut être lié au serveur Zabbix et au proxy Zabbix.

Versions

Chaque dépôt de composants Zabbix contient les étiquettes suivantes :

- latest - dernière version stable d'un composant Zabbix basé sur l'image Linux Alpine.
- alpine-latest - dernière version stable d'un composant Zabbix basé sur l'image Linux Alpine.
- ubuntu-latest - dernière version stable d'un composant Zabbix basé sur l'image Ubuntu
- alpine-4.0-latest - dernière version mineure d'un composant Zabbix 4.0 basé sur l'image Linux Alpine
- ubuntu-4.0-latest - dernière version mineure d'un composant Zabbix 4.0 basé sur l'image Ubuntu
- alpine-4.0.* - différentes versions mineures d'un composant Zabbix 4.0 basé sur l'image Linux Alpine, où * est la version mineure du composant Zabbix
- ubuntu-4.0.* - différentes versions mineures d'un composant Zabbix 4.0 basé sur l'image Linux Alpine, où * est la version mineure du composant Zabbix

Utilisation

Variables d'environnement

Toutes les images des composants Zabbix fournissent des variables d'environnement pour contrôler la configuration. Ces variables d'environnement sont répertoriées dans chaque dépôt de composants. Ces variables d'environnement sont des options des fichiers de configuration Zabbix, mais avec une méthode de nommage différente. Par exemple, `ZBX_LOGSLOWQUERIES` est égal à `LogSlowQueries` des fichiers de configuration du serveur Zabbix et des proxys Zabbix.

Certaines options de configuration ne sont pas modifiables. Par exemple, `PIDFile` et `LogType`.

Certains composants ont des variables d'environnement spécifiques, qui n'existent pas dans les fichiers de configuration officiels de Zabbix :

Variable	Composants	Description
DB_SERVER_HOST	Serveur Proxy Interface Web	Cette variable est l'adresse IP ou le nom DNS du serveur MySQL ou PostgreSQL. Par défaut, la valeur est <code>mysql-server</code> ou <code>postgres-server</code> pour MySQL ou PostgreSQL respectivement
DB_SERVER_PORT	Serveur Proxy Interface Web	Cette variable est le port du serveur MySQL ou PostgreSQL. Par défaut, la valeur est '3306' ou '5432' respectivement.
MYSQL_USER	Serveur Proxy Interface Web	Utilisateur de la base de données MySQL. Par défaut, la valeur est 'zabbix'.
MYSQL_PASSWORD	Serveur Proxy Interface Web	Mot de passe de la base de données MySQL. Par défaut, la valeur est 'zabbix'.
MYSQL_DATABASE	Serveur Proxy Interface Web	Nom de la base de données. Par défaut, la valeur est 'zabbix' pour le serveur Zabbix et 'zabbix_proxy' pour le proxy Zabbix.
POSTGRES_USER	Serveur Interface Web	utilisateur de la base de données PostgreSQL. Par défaut, la valeur est 'zabbix'.
POSTGRES_PASSWORD	Serveur Interface Web	Mot de passe de la base de données PostgreSQL. Par défaut, la valeur est 'zabbix'.
POSTGRES_DB	Serveur Interface Web	Nom de la base de données. Par défaut, la valeur est 'zabbix' pour le serveur Zabbix et 'zabbix_proxy' pour le proxy Zabbix.
PHP_TZ	Interface Web	Fuseau horaire au format PHP. La liste complète des fuseaux horaires pris en charge est disponible sur php.net . Par défaut, la valeur est 'Europe/Riga'.
ZBX_SERVER_NAME	Interface Web	Nom de l'installation Zabbix visible dans le coin supérieur droit de l'interface Web. Par défaut, la valeur est 'Zabbix Docker'
ZBX_JAVAGATEWAY_ENABLE	Serveur Proxy	Permet la communication avec la passerelle Java Zabbix pour collecter les vérifications liées à Java. Par défaut, la valeur est "false"
ZBX_ENABLE_SNMP_TRAPS	Serveur Proxy	Active la fonctionnalité de trap SNMP. Il requiert l'instance zabbix-snmptests et le volume partagé <code>/var/lib/zabbix/snmptests</code> au niveau du serveur Zabbix ou au proxy Zabbix.

Volumes

Les images permettent d'utiliser certains points de montage. Ces points de montage sont différents et dépendent du type de composant Zabbix :

Volume	Description
Agent Zabbix	
<code>/etc/zabbix/zabbix_agentd.d</code>	Le volume permet d'inclure des fichiers <code>*.conf</code> et d'étendre l'agent Zabbix en utilisant la fonctionnalité <code>UserParameter</code>
<code>/var/lib/zabbix/modules</code>	Le volume permet de charger des modules supplémentaires et d'étendre l'agent Zabbix à l'aide de la fonctionnalité LoadModule

<code>/var/lib/zabbix/enc</code>	Le volume est utilisé pour stocker les fichiers liés à TLS. Ces noms de fichiers sont spécifiés à l'aide des variables d'environnement ZBX_TLSCAFILE, ZBX_TLSCRLFILE, ZBX_TLSKEY_FILE et ZBX_TLSPSKFILE
Serveur Zabbix	
<code>/usr/lib/zabbix/alertscripts</code>	Le volume est utilisé pour les scripts d'alerte personnalisés. C'est le paramètre <code>AlertScriptsPath</code> dans zabbix_server.conf
<code>/usr/lib/zabbix/externalscripts</code>	Le volume est utilisé par des vérifications externes . C'est le paramètre <code>ExternalScripts</code> dans zabbix_server.conf
<code>/var/lib/zabbix/modules</code>	Le volume permet de charger des modules supplémentaires et d'étendre le serveur Zabbix à l'aide de la fonctionnalité LoadModule
<code>/var/lib/zabbix/enc</code>	Le volume est utilisé pour stocker les fichiers liés à TLS. Ces noms de fichiers sont spécifiés à l'aide des variables d'environnement ZBX_TLSCAFILE, ZBX_TLSCRLFILE, ZBX_TLSKEY_FILE et ZBX_TLSPSKFILE
<code>/var/lib/zabbix/ssl/certs</code>	Le volume est utilisé comme emplacement des fichiers de certificat client SSL pour l'authentification du client. C'est le paramètre <code>SSLCertLocation</code> dans zabbix_server.conf
<code>/var/lib/zabbix/ssl/keys</code>	Le volume est utilisé comme emplacement des fichiers de clé privée SSL pour l'authentification du client. C'est le paramètre <code>SSLKeyLocation</code> dans zabbix_server.conf
<code>/var/lib/zabbix/ssl/ssl_ca</code>	Le volume est utilisé comme emplacement des fichiers d'autorité de certification (CA) pour la vérification du certificat du serveur SSL. C'est le paramètre <code>SSLCALocation</code> dans zabbix_server.conf
<code>/var/lib/zabbix/snmptraps</code>	Le volume est utilisé comme emplacement du fichier <code>snmptraps.log</code> . Il peut être partagé par le conteneur <code>zabbix-snmptraps</code> et hérité à l'aide de l'option Docker <code>volumes_from</code> lors de la création d'une nouvelle instance du serveur Zabbix. La fonctionnalité de traitement des traps SNMP peut être activée en utilisant le volume partagé et en basculant la variable d'environnement <code>ZBX_ENABLE_SNMP_TRAPS</code> sur 'true'
<code>/var/lib/zabbix/mibs</code>	Le volume permet d'ajouter de nouveaux fichiers MIB. Il ne supporte pas les sous-répertoires, toutes les MIBs doivent être placées dans <code>/var/lib/zabbix/mibs</code>
Proxy Zabbix	
<code>/usr/lib/zabbix/externalscripts</code>	Le volume est utilisé par des vérifications externes . C'est le paramètre <code>ExternalScripts</code> dans zabbix_proxy.conf
<code>/var/lib/zabbix/modules</code>	Le volume permet de charger des modules supplémentaires et d'étendre le serveur Zabbix à l'aide de la fonctionnalité LoadModule
<code>/var/lib/zabbix/enc</code>	Le volume est utilisé pour stocker les fichiers liés à TLS. Ces noms de fichiers sont spécifiés à l'aide des variables d'environnement ZBX_TLSCAFILE, ZBX_TLSCRLFILE, ZBX_TLSKEY_FILE et ZBX_TLSPSKFILE
<code>/var/lib/zabbix/ssl/certs</code>	Le volume est utilisé comme emplacement des fichiers de certificat client SSL pour l'authentification du client. C'est le paramètre <code>SSLCertLocation</code> dans zabbix_proxy.conf
<code>/var/lib/zabbix/ssl/keys</code>	Le volume est utilisé comme emplacement des fichiers de clé privée SSL pour l'authentification du client. C'est le paramètre <code>SSLKeyLocation</code> dans zabbix_proxy.conf
<code>/var/lib/zabbix/ssl/ssl_ca</code>	Le volume est utilisé comme emplacement des fichiers de l'autorité de certificat (CA) pour la vérification SSL du certificat du serveur for SSL server certificate verification. It is the <code>SSLCALocation</code> parameter in zabbix_proxy.conf

<code>/var/lib/zabbix/snmptraps</code>	Le volume est utilisé comme emplacement du fichier <code>snmptraps.log</code> . Il peut être partagé par le conteneur <code>zabbix-snmptraps</code> et hérité en utilisant l'option Docker <code>volumes_from</code> lors de la création d'une nouvelle instance du serveur Zabbix. La fonctionnalité de traitement des traps SNMP peut être activée en utilisant le volume partagé et en basculant la variable d'environnement <code>ZBX_ENABLE_SNMP_TRAPS</code> sur <code>'true'</code>
<code>/var/lib/zabbix/mibs</code>	Le volume permet d'ajouter de nouveaux fichiers MIB. Il ne supporte pas les sous-répertoires, toutes les MIBs doivent être placées dans <code>/var/lib/zabbix/mibs</code>
Interface Web Zabbix basée sur un serveur Web Apache2	
<code>/etc/ssl/apache2</code>	Le volume permet d'activer HTTPS pour l'interface web Zabbix. Le volume doit contenir les deux fichiers <code>ssl.crt</code> et <code>ssl.key</code> préparés pour les connexions SSL Apache2
Interface Web Zabbix basée sur un serveur Web Nginx	
<code>/etc/ssl/nginx</code>	Le volume permet d'activer HTTPS pour l'interface web Zabbix. Le volume doit contenir les deux fichiers <code>ssl.crt</code> , <code>ssl.key</code> et <code>dhparam.pem</code> préparés pour les connexions SSL Nginx
snmptraps Zabbix	
<code>/var/lib/zabbix/snmptraps</code>	Le volume contient le fichier journal <code>snmptraps.log</code> nommé avec les traps SNMP reçus
<code>/var/lib/zabbix/mibs</code>	Le volume permet d'ajouter de nouveaux fichiers MIB. Il ne supporte pas les sous-répertoires, toutes les MIBs doivent être placées dans <code>/var/lib/zabbix/mibs</code>

Pour plus d'informations, utilisez les dépôts officiels de Zabbix dans Docker Hub.

Exemples d'utilisation

Exemple 1

L'exemple démontre comment exécuter une appliance Zabbix construite avec une base de données MySQL, un serveur Zabbix, une interface Web Zabbix basée sur un serveur web Nginx et une passerelle Java Zabbix.

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

L'instance de l'appliance Zabbix expose le port TCP/10051 (trapper Zabbix) et le port TCP/80 (HTTP) à la machine hôte.

Exemple 2

L'exemple démontre comment exécuter le serveur Zabbix avec prise en charge de base de données MySQL, l'interface Web Zabbix basée sur le serveur Web Nginx et la passerelle Java Zabbix.

1. Démarrez une instance de serveur MySQL vide

```
# 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. Démarrez l'instance de passerelle Java Zabbix

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

3. Démarrez l'instance de serveur Zabbix et liez-la à l'instance de serveur MySQL créée

```
# 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
```

L'instance de serveur Zabbix expose le port TCP/10051 (trapper Zabbix) à la machine hôte.

4. Lancez l'interface Web de Zabbix et liez l'instance avec le serveur MySQL créé et les instances de serveur Zabbix

```
# 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
```

L'instance d'interface Web Zabbix expose le port TCP/80 (HTTP) à la machine hôte.

Exemple 3

L'exemple démontre comment exécuter le serveur Zabbix avec la prise en charge de base de données PostgreSQL, l'interface Web Zabbix basée sur le serveur Web Nginx et la fonctionnalité de traps SNMP.

1. Démarrez l'instance de serveur PostgreSQL vide

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

2. Démarrez l'instance 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
```

L'instance snmptrap de Zabbix expose le port UDP/162(traps SNMP) à la machine hôte.

3. Démarrez l'instance de serveur Zabbix et liez la à l'instance de serveur PostgreSQL créée.

```
# 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
```

L'instance de serveur Zabbix expose le port TCP/10051 (trapper Zabbix) à la machine hôte.

4. Démarrez l'interface Web de Zabbix et liez l'instance avec le serveur PostgreSQL créé et les instances de serveur Zabbix.

```
# 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
```

L'instance de l'interface Web Zabbix expose le port TCP/443 (HTTPS) à la machine hôte.
Le répertoire `/etc/ssl/nginx` doit contenir un certificat avec un nom requis.

Docker Compose

Zabbix fournit également des fichiers compose pour définir et exécuter des composants Zabbix multi-conteneurs dans Docker. Ces fichiers compose sont disponibles dans le dépôt officiel Zabbix-docker sur github.com: <https://github.com/zabbix/zabbix-docker>. Ces fichiers compose sont ajoutés à titre d'exemples, ils sont surchargés. Par exemple, ils contiennent des proxys avec la prise en charge MySQL et SQLite3.

Il existe quelques différentes versions des fichiers compose :

Nom de fichier	Description
<code>docker-compose_v3_alpine_mysql_latest.yaml</code>	Le fichier compose exécute la dernière version des composants Zabbix 4.0 sur Alpine Linux avec base de données MySQL.
<code>docker-compose_v3_alpine_mysql_local.yaml</code>	Le fichier compose construit localement la dernière version de Zabbix 4.0 et exécute les composants Zabbix sur Alpine Linux avec base de données MySQL.
<code>docker-compose_v3_alpine_pgsql_latest.yaml</code>	Le fichier compose exécute la dernière version des composants Zabbix 4.0 sur Alpine Linux avec base de données PostgreSQL.
<code>docker-compose_v3_alpine_pgsql_local.yaml</code>	Le fichier compose construit localement la dernière version de Zabbix 4.0 et exécute les composants Zabbix sur Alpine Linux avec base de données PostgreSQL.
<code>docker-compose_v3_centos_mysql_latest.yaml</code>	Le fichier compose exécute la dernière version des composants Zabbix 4.0 sur CentOS 7 avec base de données MySQL.
<code>docker-compose_v3_centos_mysql_local.yaml</code>	Le fichier compose construit localement la dernière version de Zabbix 4.0 et exécute les composants Zabbix sur CentOS 7 avec base de données MySQL.
<code>docker-compose_v3_centos_pgsql_latest.yaml</code>	Le fichier compose exécute la dernière version des composants Zabbix 4.0 sur CentOS 7 avec base de données PostgreSQL.
<code>docker-compose_v3_centos_pgsql_local.yaml</code>	Le fichier compose construit localement la dernière version de Zabbix 4.0 et exécute les composants Zabbix sur CentOS 7 avec base de données PostgreSQL.
<code>docker-compose_v3_ubuntu_mysql_latest.yaml</code>	Le fichier compose exécute la dernière version des composants Zabbix 4.0 sur Ubuntu 14.04 avec base de données MySQL.
<code>docker-compose_v3_ubuntu_mysql_local.yaml</code>	Le fichier compose construit localement la dernière version de Zabbix 4.0 et exécute les composants Zabbix sur Ubuntu 14.04 avec base de données MySQL.
<code>docker-compose_v3_ubuntu_pgsql_latest.yaml</code>	Le fichier compose exécute la dernière version des composants Zabbix 4.0 sur Ubuntu 14.04 avec base de données PostgreSQL.

<code>docker-compose_v3_ubuntu_pgsql_local.yaml</code>	Le fichier compose construit localement la dernière version de Zabbix 4.0 et exécute les composants Zabbix sur Ubuntu 14.04 avec base de données PostgreSQL.
--	--

Les fichiers compose de Docker disponibles prennent en charge les versions 2 et 3 de Docker Compose.

Stockage

Les fichiers compose sont configurés pour prendre en charge le stockage local sur une machine hôte. Docker Compose crée un répertoire `zbx_env` dans le dossier contenant le fichier compose lorsque vous exécutez des composants Zabbix à l'aide du fichier compose. Le répertoire contiendra la même structure que celle décrite ci-dessus dans la section [Volumes](#) et le répertoire de stockage de la base de données.

Il existe également des volumes en mode lecture seule pour les fichiers `/etc/localtime` et `/etc/timezone`.

Fichiers d'environnement

Dans le même répertoire avec les fichiers compose sur [github.com](#) vous pouvez trouver des fichiers avec des variables d'environnement par défaut pour chaque composant dans le fichier compose. Ces fichiers d'environnement sont nommés comme `.env_<type de composant>`.

Exemples

Exemple 1

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

La commande téléchargera les dernières images Zabbix 4.0 pour chaque composant Zabbix et les exécutera en mode détaché.

N'oubliez pas de télécharger les fichiers `.env_<type de composant>` sur le dépôt officiel Zabbix de [github.com](#) avec les fichiers compose.

Exemple 2

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

La commande téléchargera l'image de base Ubuntu 14.04, puis construira localement les composants de Zabbix 4.0 et les exécutera en mode détaché.

From:

<https://www.zabbix.com/documentation/4.0/> - **Zabbix Documentation 4.0**

Permanent link:

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

Last update: **2019/04/12 12:28**

