

Docker-compose overview

Table of Contents

- [Docker-compose overview](#)
 - [Présentation de Docker Compose](#)
 - [Caractéristiques](#)
 - [Plusieurs environnements isolés sur un seul hôte](#)
 - [Préserver les données de volume lorsque des conteneurs sont créés](#)
 - [Recréer uniquement les conteneurs qui ont changé](#)
 - [Variables et déplacement d'une composition entre environnements](#)
- [Installation de Docker Compose](#)
 - [Installation sur Centos](#)

Présentation de Docker Compose

Compose est un outil de définition et d'exécution d'applications Docker contenant plusieurs conteneurs. Avec Compose, on utilise un fichier YAML pour configurer les services de l'application, puis, avec une seule commande, pour créer et démarrer tous les services à partir d'une configuration.

Compose fonctionne dans tous les environnements: production, préparation, développement, tests, ainsi que des flux de travaux CI.

Utiliser Compose est fondamentalement un processus en trois étapes:

- Définir l'environnement de votre application avec un fichier Docker afin qu'il puisse être reproduit n'importe où.
- Définir les services qui composent votre application dans docker-compose.yml afin qu'ils puissent être exécutés ensemble dans un environnement isolé.
- Exécuter `docker-compose up` pour démarrer et exécuter l'intégralité de l'application.

Un fichier docker-compose.yml ressemble à ceci:

```
version: '3'
services:
  web:
    build: .
    ports:
      - "5000:5000"
    volumes:
      - ./code
      - logvolume01:/var/log
    links:
      - redis
  redis:
```

```
image: redis
volumes:
  logvolume01: {}
```

Compose propose des commandes permettant de gérer l'ensemble du cycle de vie de votre application:

- Démarrer, arrêter et reconstruire des services
- Afficher le statut des services en cours d'exécution
- Diffuser la sortie du journal des services en cours d'exécution
- Exécuter une commande unique sur un service

Caractéristiques

Les caractéristiques de Compose qui le rendent efficace sont:

- Plusieurs environnements isolés sur un seul hôte
- Préserver les données de volume lorsque des conteneurs sont créés
- Recréer uniquement les conteneurs qui ont changé
- Variables et déplacement d'une composition entre environnements

Plusieurs environnements isolés sur un seul hôte

Compose utilise un nom de projet pour isoler les environnements les uns des autres et peut être utilisé dans plusieurs contextes différents:

- Sur un hôte dev, pour créer plusieurs copies d'un même environnement, par exemple lorsque vous souhaitez exécuter une copie stable pour chaque branche de fonctionnalité d'un projet
- Sur un serveur de CI, pour éviter toute interférence entre les générations, on peut définir le nom du projet par un numéro de compilation unique.
- Sur un hôte partagé ou un hôte dev, pour empêcher différents projets, pouvant utiliser les mêmes noms de services, d'interférer les uns avec les autres

Le nom de projet par défaut est le nom de base du répertoire de projet. On peut définir un nom de projet personnalisé à l'aide de l'option de ligne de commande `-p` ou de la variable d'environnement `COMPOSE_PROJECT_NAME`.

Préserver les données de volume lorsque des conteneurs sont créés

Lorsque le docker-compose s'exécute, il détecte les conteneurs des exécutions précédentes et copie les volumes de l'ancien conteneur dans le nouveau. Ce processus garantit que toutes les données créées dans les volumes N'est pas perdu.

Recréer uniquement les conteneurs qui ont changé

Compose met en cache la configuration utilisée pour créer un conteneur. Lorsqu'on redémarre un

service qui n'a pas changé, Compose réutilise les conteneurs existants. La réutilisation de conteneurs signifie que l'on peut modifier très rapidement un environnement.

Variables et déplacement d'une composition entre environnements

Compose prend en charge les variables du fichier Compose que l'on peut utiliser pour personnaliser une composition pour différents environnements ou différents utilisateurs.

On peut étendre un fichier Compose à l'aide du champ `extend` ou en créant plusieurs fichiers Compose (voir les extensions pour plus de détails).

Installation de Docker Compose

Installation sur Centos

Installer des packages supplémentaires pour Enterprise Linux

```
$ sudo yum install epel-release
```

Installer python-pip

```
$ sudo yum install -y python-pip
```

Ensuite, installer Docker Compose:

```
$ sudo pip install docker-compose
```

Il faut également mettre à jour les paquets Python sous CentOS 7 pour que docker-compose s'exécute correctement:

```
$ sudo yum upgrade python*
```

Pour vérifier le succès de l'installation de Docker Compose, exécuter:

```
$ docker-compose --version
```

From:

<http://www.ouarte.garden/> - **dwndoc**

Permanent link:

<http://www.ouarte.garden/doku.php?id=virtualisation:docker-compose>

Last update: **2025/02/19 10:59**

