

Accueil > Blog > Docker et Docker-Compose : les commandes à connaître

# Docker et Docker-Compose : les commandes de base à connaître

Technologie, Docker  
5 min.

Yohan Gracia

Mise à jour  
12/04/2019

*Êtes-vous réellement confiant dans votre utilisation de Docker ? Si ce n'est pas le cas, cet article sur la conteneurisation d'une application legacy peut vous aider. Docker est maintenant utilisé avec énormément de technologies, et vous pourriez avoir besoin d'apprendre vite ou de raviver de vieux souvenirs. Voici une liste des commandes fondamentales à connaître pour utiliser Docker et Docker-Compose.*

S'inscrire à la newsletter de Padok



Ne manquez plus les actualités DevOps et inscrivez-vous à la newsletter de Padok dès à présent !

S'abonner maintenant

## Sommaire

1. Un peu de contexte
2. Commandes de monitoring
3. Commandes de runtime





## 3. Commandes de logs

### Un peu de contexte:

Un conteneur est assez semblable à une VM. Alors que la VM est une machine entièrement nouvelle (d'un point de vue software) construite sur une machine physique, un conteneur n'a de son côté pas tous les composants habituels d'une machine a habituellement. Pour être plus précis, il n'a pas d'OS entier, mais seulement ce qui est nécessaire pour faire tourner ses applications. Il est construit à partir d'une image, qui correspond à sa configuration.

Vous avez peut-être entendu parler de Docker-compose. La différence entre [Docker et Docker-compose](#) est simple: les commandes Docker ne concernent qu'un seul conteneur (ou image) à la fois, alors que Docker-compose gère plusieurs [conteneurs Docker](#).

### Commandes de monitoring:

Les commandes suivantes sont, à mes yeux, les premières que vous avez besoin de connaître quand vous utilisez Docker.

`docker ps (-a)`

`docker ps` vous affiche toutes les instances environnement. Si vous ajoutez l'option `-a`, a

`docker images (-a)`

Cette commande vous montre les images que vous avez construites, et le `-a` vous montre les images intermédiaires.

`docker network ls`

`docker-compose ps`

#### S'inscrire à la newsletter de Padok



Ne manquez plus les actualités DevOps et inscrivez-vous à la newsletter de Padok dès à présent !

[S'abonner maintenant](#)





## Commandes de runtime:

Vous avez maintenant besoin d'images et de conteneurs pour tester les commandes précédentes.

```
docker-compose up (-d) (--build)
```

```
docker-compose stop
```

La docker-compose est la plus simple car vous n'avez besoin que de 2 commandes : up et stop. stop est assez explicite et stop (mais ne supprime pas) vos conteneurs, mais up nécessite plus d'explications : cela va construire vos images si elles ne le sont pas déjà, et va démarrer vos dockers. Si vous voulez re-build vos images, utilisez l'option --build (vous pouvez aussi utiliser la commande docker-compose build pour uniquement construire des images). L'option -d, qui signifie "detach" fait tourner les conteneurs en tâche de fond.

```
docker build (-t <NAME>) <PATH>/<URL>
```

Avec Docker, vous avez besoin d'une commande séparée pour construire votre image, où vous pouvez spécifier le nom de votre image et vous devez spécifier le PATH ou URL selon votre contexte (cela peut être un repo git).

```
docker run (-d) (-p <hostPort>:<containerPort>) <image>
```

run crée le conteneur en utilisant l'image que vous spécifiez. Nous vous recommandons d'ajouter des paramètres. Vous pouvez aussi spécifier des ports à exposer. L'option -d, qui signifie "detach" fait tourner les conteneurs en tâche de fond.

```
docker start <ID>/<NAME>
```

```
docker stop <ID>/<NAME>
```

Le start and stop ne devraient pas être trop compliqués à comprendre, mais il faut noter que vous pouvez "start" uniquement des conteneurs qui sont déjà arrêtés, donc déjà build avec la commande run.

### S'inscrire à la newsletter de Padok

Ne manquez plus les actualités DevOps et inscrivez-vous à la newsletter de Padok dès à présent !

[S'abonner maintenant](#)





Cette commande vous permet de lancer un shell sur votre conteneur. Je préfère utiliser `/bin/bash` mais votre conteneur peut ne pas avoir bash d'installé, et seulement "sh" qui est plus courant (surtout sur les alpine). Si vous avez des configurations spéciales dans votre conteneur, vous aurez peut-être besoin d'utiliser des arguments supplémentaires pour vous y connecter. Cette commande peut vous permettre de faire bien plus, je vous recommande donc de lire cette [doc](#) pour trouver des informations supplémentaires..

## Commandes de suppression:

Ces commandes permettent de supprimer vos conteneurs et vos images. Vous en aurez probablement besoin pour libérer de l'espace disque.

```
docker rm <ID>/<NAME>
```

```
docker-compose rm
```

Le `docker rm` supprime seulement un conteneur alors que `docker-compose rm` supprime tous les conteneurs démarrés avec une commande `docker-compose`.

```
docker rmi <ID>/<NAME>
```

Docker `rmi` supprime l'image que vous passez en paramètre, ainsi que toutes les images intermédiaires utilisées pour la construire.

S'inscrire à la newsletter de Padok

Ne manquez plus les actualités DevOps et inscrivez-vous à la newsletter de Padok dès à présent !

[S'abonner maintenant](#)

## Commandes de logs:

Les commandes suivantes sont utiles quand vous devez déboguer certains de vos conteneurs (ou, plus souvent, l'application que vous déployez à l'intérieur).

```
docker logs <ID>/<NAME> (-f --tail <NBLINE>)
```

Cette commande affiche les logs du container passé en paramètre. Si vous utilisez l'option `-f` ou `--tail` vous pouvez suivre en live le flux de vos logs (`<NBLINE>` est le nombre de lignes que vous voulez afficher).





## `docker-compose logs (<ID>/<NAME>)`

L'option (<ID>/<NAME>) avec `docker-compose logs` vous permet de voir les logs d'un conteneur uniquement, au lieu de voir tous les logs. L'astuce ici est que si vous n'utilisez pas l'option `-d` quand vous utilisez `docker run` ou `docker-compose up` vous verrez vos logs directement (mais vous aurez besoin d'arrêter le conteneur pour quitter la vue). Cela peut toujours être utile pour déboguer des applications au démarrage.

Voici les commandes de base pour Docker et Docker-compose. N'hésitez pas à nous contacter si vous avez des questions sur [Docker et son intégration avec Kubernetes](#) ou que vous souhaitez que nous ajoutions certaines commandes à l'article

## Articles similaires

Qu'est ce que la sobriété

Technologie

6 min

Lire cet article

### S'inscrire à la newsletter de Padok



Ne manquez plus les actualités DevOps et inscrivez-vous à la newsletter de Padok dès à présent !

[S'abonner maintenant](#)

