



Commandes usuelles de Docker

Voici un aide-mémoire sur les commandes Docker les plus couramment employées.

`$ docker -machine ls`

Affiche la liste de toutes les machines.

`$ docker-machine ls -t 60`

Accorde 60 secondes pour réunir les instances requises en vue d'un déploiement dans plusieurs nuages.

`$ docker ps`

Énumère les conteneurs actifs.

`$ docker ps -a`

Énumère les conteneurs inactifs.

`$ docker ps -s`

Énumère les conteneurs actifs, avec leur taille et l'espace utilisé sur le disque virtuel.

`docker inspect <friendly-name | container-id>`

Donne des précisions sur un conteneur actif (son adresse IP, par exemple).

`docker port <friendly-name | container-id> 6379`

Indique le port externe attribué à un port interne. Dans l'exemple, le numéro du port externe est attribué au port interne 6379.

`docker logs <friendly-name | container-id>`

Affiche les messages rédigés par le conteneur pour Standard Error (STDERR) ou Standard Out (STDOUT).

`$ docker rm 'containername'`

Supprime un conteneur inactif.

`$ docker images`

Énumère les images téléchargées.

`$ docker rmi imagename`

Supprime une image.

`$ docker system df`

Indique l'espace disque utilisé par Docker.

`$ docker system events`

Indique les événements en temps réel sur le serveur.

`$ docker system info`

Affiche les informations sur le système.

`$ docker system prune`

`$ docker system prune -a --volumes`

Supprime les données inutilisées et nettoie l'environnement Docker. – *Attention : les ressources inactives seront supprimées!*

Exportation des conteneurs Docker avec préservation des couches de dossiers

Les commandes qui suivent permettent d'exporter et d'importer des conteneurs sans destruction des données qu'ils renferment. L'importation ne modifie en rien le répertoire de dossiers. Les annexes des volumes ne seront ni copiées ni sauvegardées, mais il est possible de les y rattacher à nouveau.

`$ docker export --output="latest.tar" containername`

Exporte le conteneur en sauvegardant les données qu'il renferme.

`$ docker import latest.tar imagename`

Importe l'image Docker dans une nouvelle image.

```
$ docker run --name nameofyourcontainer -d -it imagename /bin/sh
```

Teste l'image comme s'il s'agissait d'un conteneur.

```
$ docker exec -it containername /bin/bash
```

Se connecte à un conteneur pour vérifier s'il y a des problèmes ou pour en permettre l'adaptation.

```
$ docker build -t image-name:versionnumber .
```

Crée un fichier et une image Docker Compose à partir du répertoire avec Dockerfile.

```
$ docker run -d -p 80:80 image-name:ver
```

Démarre l'image qui vient d'être créée sous forme de conteneur avec attribution du port interne 80 au port 80 de l'hôte.

Don McCullough

Architecte en solutions, DAIR-ATIR

www.canarie.ca

don.mccullough@canarie.ca