

Docker

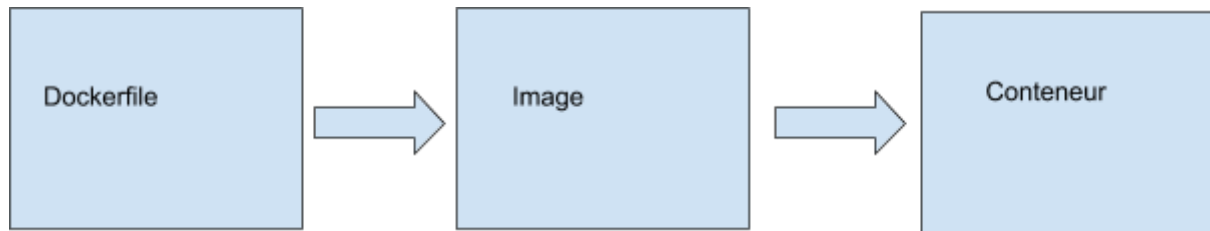


Image : Une image Docker est un fichier qui une fois lancé crée un conteneur docker. Une image est construite à partir d'un dockerfile

Dockerfile : Un dockerfile est un fichier texte qui contient une suite de commandes permettant la création d'une image.

Container: Un conteneur est une unité logicielle qui regroupe le code et toutes ses dépendances afin que l'application s'exécute rapidement et de manière fiable d'un environnement informatique à un autre.

volume : Un volume Docker fournit un mécanisme pour assurer la persistance des données d'un conteneur ou lui permettre « d'échanger » des données avec d'autres conteneurs partageant le même volume.

Un volume Docker est initialisé lors de la création d'un conteneur et est monté dans le conteneur comme un système de fichiers. Si l'image du conteneur contient déjà des données dans le répertoire dans lequel est monté le système de fichiers, ces données sont copiées dans le nouveau volume lors de son initialisation. Les volumes de données ont l'avantage particulier de survivre à l'arrêt du conteneur et par défaut, Docker n'efface pas les volumes, même lors de la destruction d'un conteneur associé. En fait, il est même impossible d'effacer un volume tant que ce dernier est référencé par un conteneur.

cheat sheet:

Container Management:

- **docker create** image dossier
- **docker run** image : create & start container
-
- **docker start** container
- **docker stop** container
- **docker kill** container
- **docker restart** container
- **docker pause** container
- **docker unpause** container
- **docker rm** container

- **docker rm -f** container : remove running container

Inspecting the container:

- **docker ps** : list running containers
- **docker ps -a**: list all containers
- **docker logs** container
- **docker top** container : list the processes running inside the container
- **docker diff** container: show the differences with the image
- **docker inspect** container : show le-level infos in json format

Interacting with the container :

- **docker attach** container //
- **docker cp** container:path hostpath|- : copy files from container
- **docker cp** hostpath|- container:path : copy files to container
- **docker export** container : export the content of a container as a tar archive
- **docker exec** container args : run a command in an existing container
- **docker wait** container : wait until the container terminates and return the exit code
- **docker commit** container image : create an image from a container

Image management:

- **docker images** : lis all local images
- **docker rmi** image : remove image
- **docker history** image: show the list of ancestors of the image
- **docker inspect** image : show low-level infos in json format
- **docker tag** image tag : tag an image

Dockerfile:

- FROM : specify base image (can be "scratch")
- RUN : executes specified command
- ENTRYPOINT: Specify the command to execute the container (app main)
- CMD: specify the command at the time of the execution of the image
- COPY : copy files from host machine to container image
- ENV : add environment variables
- EXPOSE : open designated port
- WORKDIR : change current directory
- LABEL : metadata
- VOLUME: specify mount point
- WORKDIR: set the working directory in the container

Liens utiles :

<https://hub.docker.com/> : les images utilisables

<https://docs.docker.com/> : doc

Docker compose :

<https://jstobigdata.com/docker-compose-cheatsheet/> : cheat sheet géniale

<https://docs.docker.com/compose/compose-file/#dockerfile> : How to create a .yml file