

## **INCEPTION**

### **Docker containers:**

Il s'agit d'environnements informatiques légers et autonomes qui permettent d'exécuter un service.

### **A Docker image:**

Est un modèle en lecture seule qui contient un ensemble d'instructions pour créer un conteneur qui fonctionne sur la plateforme Docker. C'est là que vous spécifiez ce que vous voulez qu'il se passe dans le conteneur vise

### **Docker volumes:**

Sont des systèmes de fichiers montés sur les conteneurs Docker pour préserver les données générées par le conteneur en cours d'exécution. Les volumes sont stockés sur l'hôte, indépendamment du cycle de vie du conteneur. Cela permet aux utilisateurs de sauvegarder des données et de partager des systèmes de fichiers entre les conteneurs. systèmes de fichiers entre les conteneurs.

Permet d'échanger les données entre les conteneurs comme un fichier partagé.

### **A network :**

sert à permettre la communication entre les conteneurs. Docker dispose de ses propres pilotes réseau qui ont des propriétés prédéfinies. Pour ce projet, nous avons seulement besoin du réseau par défaut, qui est un réseau pont. Il sera précisé dans le docker-compose.yml pour pouvoir le spécifier ensuite dans les conteneurs concernés.

**Ressources utilisées:**

[https://github.com/iciamyplant/ft\\_services](https://github.com/iciamyplant/ft_services)

[https://github.com/t0mm4rx/ft\\_services](https://github.com/t0mm4rx/ft_services)

<https://docs.docker.com/get-started/overview/>

[https://hub.docker.com/\\_/mariadb](https://hub.docker.com/_/mariadb)

<https://docs.docker.com/samples/wordpress/>

[https://hub.docker.com/\\_/nginx](https://hub.docker.com/_/nginx)

<https://www.biaudelle.fr/mariadb-avec-docker-compose/>

## Introduction

Pour avoir les droits d'administrateur avec ton login, rien de bien compliqué:

```
sudo adduser login  
  
sudo usermod -aG sudo login  
  
sudo usermod -aG vboxsf login (pour le doc partage)
```

## First step Docker installation

On supprime la version actuelle de docker:

```
sudo apt-get remove docker-compose  
  
sudo rm -f /usr/local/bin/docker-compose
```

On passe maintenant à l'installation

Pour ce faire, rien de plus simple:

<https://docs.docker.com/engine/install/ubuntu/>

Pour les flemmards:

<https://get.docker.com/>

```
This script is meant for quick & easy install via:  
$ curl -fsSL https://get.docker.com -o get-docker.sh  
$ sh get-docker.sh
```

Pour éviter d'utiliser docker avec sudo sans cesse, on ajoute :

```
$ sudo usermod -aG docker login
```

En terme de sécurité, on est pas au max mais c'est pas grave, on est sur notre VM.

Pour l'installation de docker machine :

<https://github.com/docker/machine/releases>

Pour checker si tout est bien installé:

```
$ docker compose version
```

Ici dans cette commande, on a le CLI (version) en première partie , puis la deuxième partie on communique avec le serveur aussi appelé Daemon, c'est la dernière version en ligne, le fait d'avoir un retour nous annonce bien qu'on communique avec le serveur:

```
$ docker version
```

```
$ docker info
```

Permet d'avoir un peu plus d'info

```
$ docker
```

Pour plus de commandes

Petit tuto

<https://upcloud.com/resources/tutorials/wordpress-with-docker>

## Compréhension de l'environnement

Quelle est la différence entre une IMAGE et CONTAINER ?

Image est le binaire et les libraires pour faire tourner le container.

Le container est l'instance de l'image qui tourne.

Quelle est la différence entre alpine et ubuntu ?

la size, pour voir ca on fait un simple docker pull ubuntu, docker pull alpine, et docker image ls

Nginx: Serveur web très basique, il est très facile d'utilisation pour utiliser les fonctions de conteneur Docker

MariaDB:

Container ou VM ?

Le container est un processus restreint dans le système d'exploitation

## Commandes

### commandes de base

docker container ls

docker container ls -a (permet de lister meme les images stopped et removed)

docker top \$name (permet de lister les processus qui s'exécutent dans un conteneur spécifique)

ps aux permettra de montrer tous les processus en cours

docker stop \$name (stop le container)

Pour voir si tout fonctionne \$ curl localhost:8080 (par exemple)

### Cleaning:

You can use "prune" commands to clean up images, volumes, build cache, and containers. Examples include:

- `docker image prune` to clean up just "dangling" images
- `docker system prune` will clean up everything
- The big one is usually `docker image prune -a` which will remove all images you're not using. Use `docker system df` to see space usage.

### Informations sur un container:

`docker container top` (process list in one container)

`docker container inspect` (détail de la configuration d'un container, et ses metadatas, volumes config ect [c'est un fichier json])

`docker container stats` (performance stats pour tous les containers)

### commande dans les conteneurs:

(`"it"`, le `-t` permet de simuler un terminal avec ssh, `-i` garde la session ouverte et interactive)

`docker container run -it -> $new_container`

`docker container exec -it -> $container` (meme chose que run, mais a faire sur un container existant)

exemple go avec ubuntu:

`docker container run -it --name ubuntu ubuntu`

Si on veut relancer un container existant ?

`docker container run -ai --name ubuntu ubuntu`

### commande network conteneurs:

`docker container port $nomContainer`

### commande network : cli management:

`docker network ls` (permet de montrer les networks)

`docker network inspect`

`docker network create --driver (créer nouveau réseau)`

`docker network connect/disconnect ()` <https://blog.alphorm.com/reseau-docker-partie-1-bride/>

## LES IMAGES

Image layers:

Tout simplement dans une image, il y a plusieurs couches(layers)

- ubuntu
- apt
- env
- port (optional)
- copy(optional)

`docker history nomImage:latest`

`docker image inspect mongo`

## LES IMAGES TAGS

Tags sont des identifiants dans les images pour revenir vers d'autres versions.

c'est le même fonctionnement que github mais pour les versions de docker.

`docker image tag $nom $nom:titre`

`docker image push $name/$nom:titre`

## LES IMAGES

Lorsqu'on vient monter une image via la commande :

`sudo docker image build -t wordpress .`

(la compilation montre le build cache afin de ne pas rebuild par dessus si la ligne n'a pas change dans le dockerfile)

## NGINX

[https://hub.docker.com/\\_/nginx](https://hub.docker.com/_/nginx)

en gros docker pull ->

`docker image ls`

`docker run -it -d`

`docker ps -a`

`docker exec -it`

`docker log`

`docker history`