

# Containers

Docker est un logiciel qui est pas mal utilisé pour installer des services sur un serveur.

L'avantage c'est que Docker va virtualiser les services dans des genres de "mini-machine virtuelles". Ce qui permet :

- D'avoir un environnement isolé et constant. Permettant de déployer quelque chose quel que soit l'état de la machine hôte.
- Déployer des services rapidement et facilement
- Permet une plus grande extensibilité
- Plus sécurisé grâce à l'isolation
- Il est aussi possible de revenir dans des états précédents d'un pod

C'est notamment ce qui est utilisé quand on a utilisé Gitlab-CI pour l'automatisation.

Et c'est ce que l'on va de nouveau utiliser quand on va utiliser Kubernetes pour avoir une plus grande extensibilité.

## Pourquoi j'aime Docker

Il y a beaucoup d'avantages que je trouve à Docker.

- La capacité de compiler un programme sans être parasité par d'autres choses installées sur l'ordi (exemple versions plus récentes de certaines librairies)
- La capacité d'avoir une vue sur toute l'infrastructure (environnement, ports, domaines, images, dossiers de configuration, commandes d'installation, versions, lien avec d'autres services, etc)
- La capacité de tester des choses sans avoir à se soucier d'annuler les changements ou de brick son système.
- Pouvoir créer des images qui fonctionnent sur n'importe quel serveur tournant n'importe quel OS avec n'importe quel infrastructure (portabilité)
- Pouvoir facilement orchestrer les différents services à l'aide d'un simple fichier. Et même pouvoir gérer plusieurs serveurs en même temps.
- Pouvoir faire facilement des backups de tout, en ayant tout dans une image sans dépendre de dépendances etc

## Quelques concepts à comprendre

Un **volume** est un moyen de rendre certaines données persistantes. Car quand vous re-créez un conteneur, tout le reste sera réinitialisé. Le volume lie un dossier sur la machine hôte à un dossier dans le conteneur. Et il y a deux types de volumes:

- Soit le volume est défini et géré par Docker (et se trouve donc dans `/var/lib/docker/volumes`) ou dans quel cas le contenu initial qui était dans le dossier sur le conteneur prends le dessus
- Soit le volume est un *bind*, c'est à dire un lien direct à un dossier de la machine cible. Dans quel cas le contenu initial sur la machine hôte prends le dessus. Donc si le dossier/fichiers sur la machine hôte n'existe pas, ils se feront aussi supprimer au moment du lancement du conteneur.

Pour utiliser un bind et quand même faire en sorte de garder les données du côté conteneur, il vaut mieux alors définir la mise en place des données dans le dossier au moment du `entrypoint.sh` (on va voir ça quand on vera la dockerization)

Un **réseau** est un lien entre plusieurs conteneurs qui leur permet de communiquer sans avoir besoin d'exposer les ports de ces derniers. On va en savoir plus sur le chapitre sur `traefik` (reverse-proxy)

Une **conteneur** est ce qui tourne sur Docker, un conteneur peut être lié à un volume, à des réseaux, etc. Un conteneur fait tourner une **image** qui aura été préalablement créée.

Une **image** est en quelque sorte l'état "de base" d'un conteneur. Elle contient toutes les informations initiales nécessaires, le script qui a permis de la configurer, la commande de lancement du conteneur et tous les fichiers qui y ont été ajouté. C'est une genre d'"archive" de machine virtuelle si on veut.

Un **registry** est un répertoire pour y envoyer ou télécharger des images. Le plus connus de tous est le [Docker Hub](#)

# Installation

Pour l'installer vous pouvez suivre les [instructions suivantes](#) (évitez Docker Desktop si possible)

---

Revision #1

Created 26 April 2023 19:09:32 by SnowCode

Updated 26 April 2023 19:09:32 by SnowCode