



Conteneurs & Docker

Damien Montmoulinex - Toulouse Ynov Campus 2022/2023
Master Cloud, Sécurité & Infrastructure / DevOps

Module Conteneurs et Orchestration

Introduction



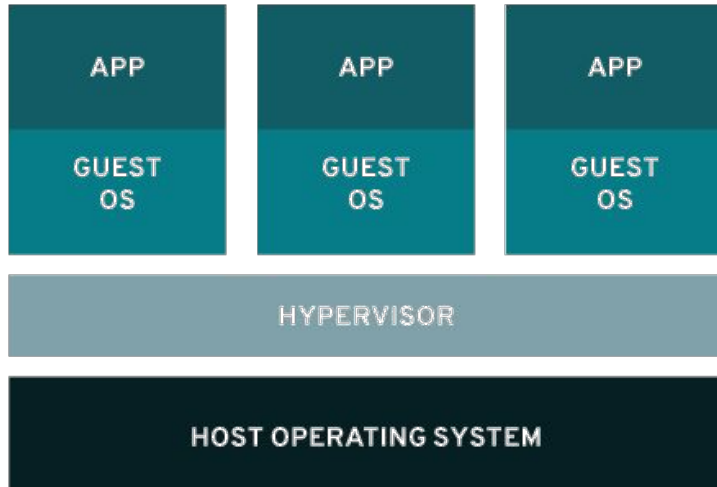
L'outil DevOps incontournable ! Il réunit Dev et Ops autour d'une technologie commune !

Différentes technologies de conteneurs :

- LXC
- Docker
- Podman

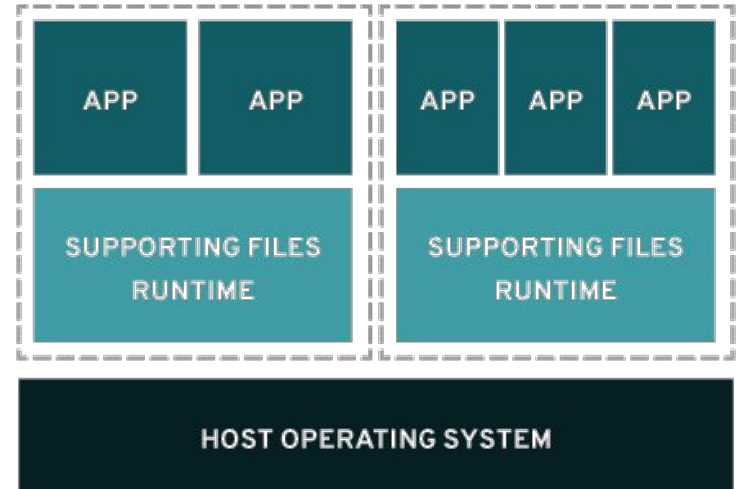
Virtualisation et Conteneurs

VIRTUALIZATION



VS.

CONTAINERS



Avantages des conteneurs

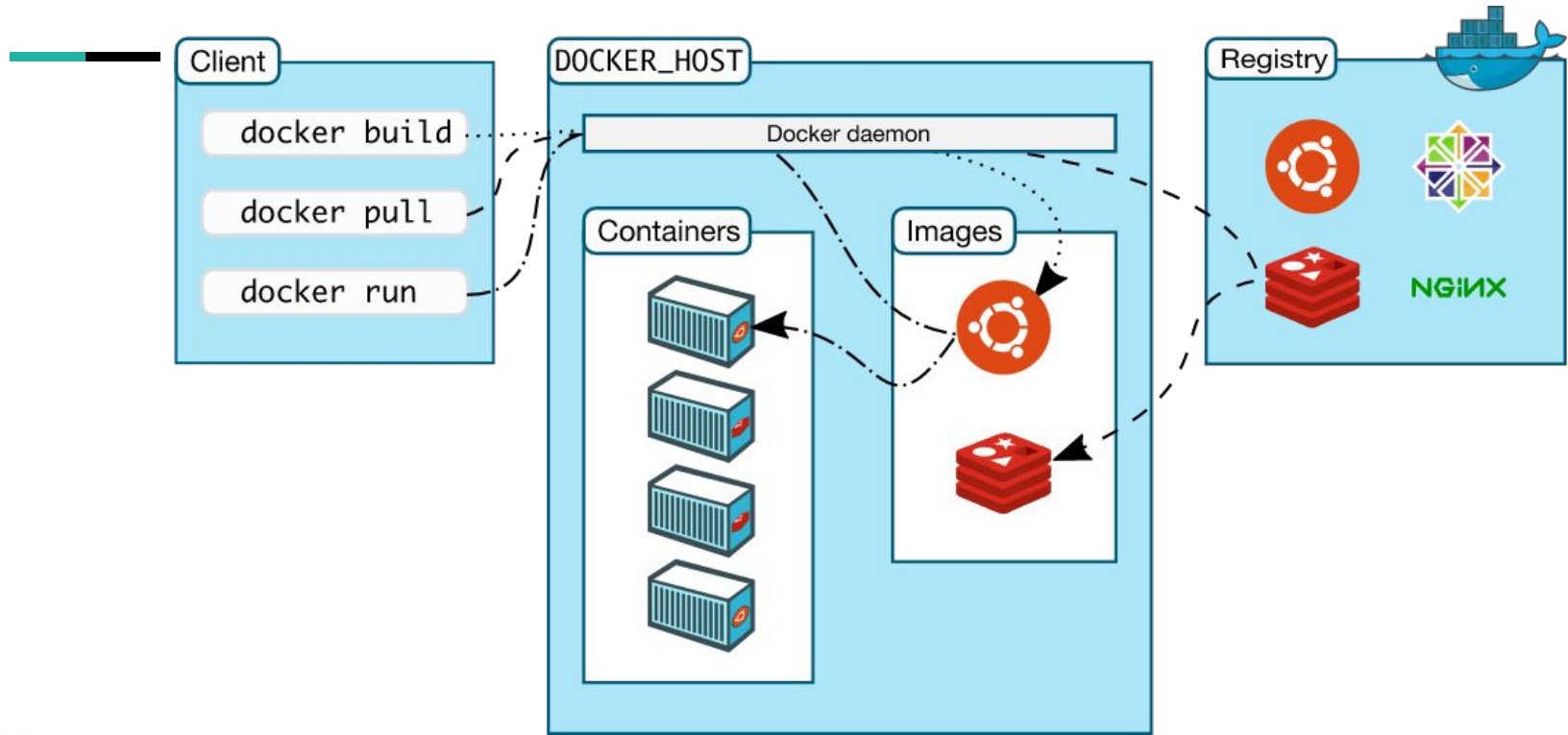
- Démarre plus rapidement qu'une machine virtuelle
- Plus léger (en général quelques centaines de Mo maximum)
 - Image alpine (encore plus petites)
 - Réduction de la surface d'attaque
- Multi-plateforme (Linux, Windows, MacOS)
 - Réponds à la problématique "ça fonctionne sur ma machine !"
 - Il peut rester quelques différences (architecture matérielle, plugins différents (volumes ou networks), ...)

Docker

Docker

- Distribué en open-source depuis 2013
- Une banque d'images communes et publiques (Docker Hub)
- Historiquement basé sur LXC, puis sur containerd

Docker - Fonctionnement général

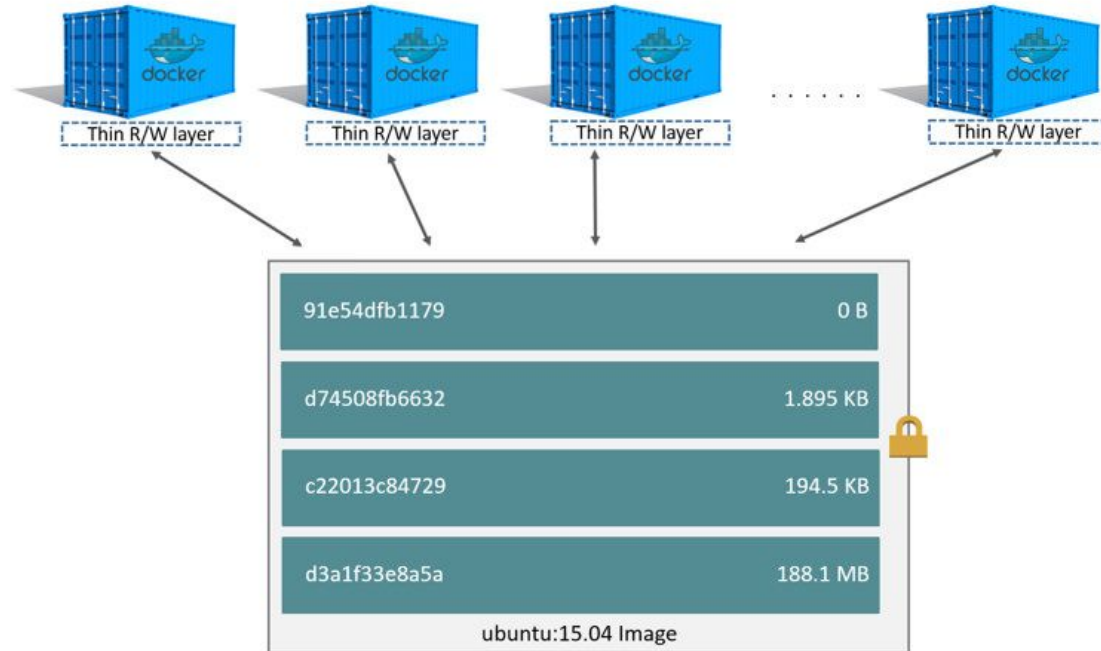


Images Docker



- Une image Docker contient tout ce dont une application à besoin pour s'exécuter correctement
- Comment obtenir une image :
 - Docker pull
 - Docker build
 - Docker commit

Instanciation d'une image



Conteneurs Docker



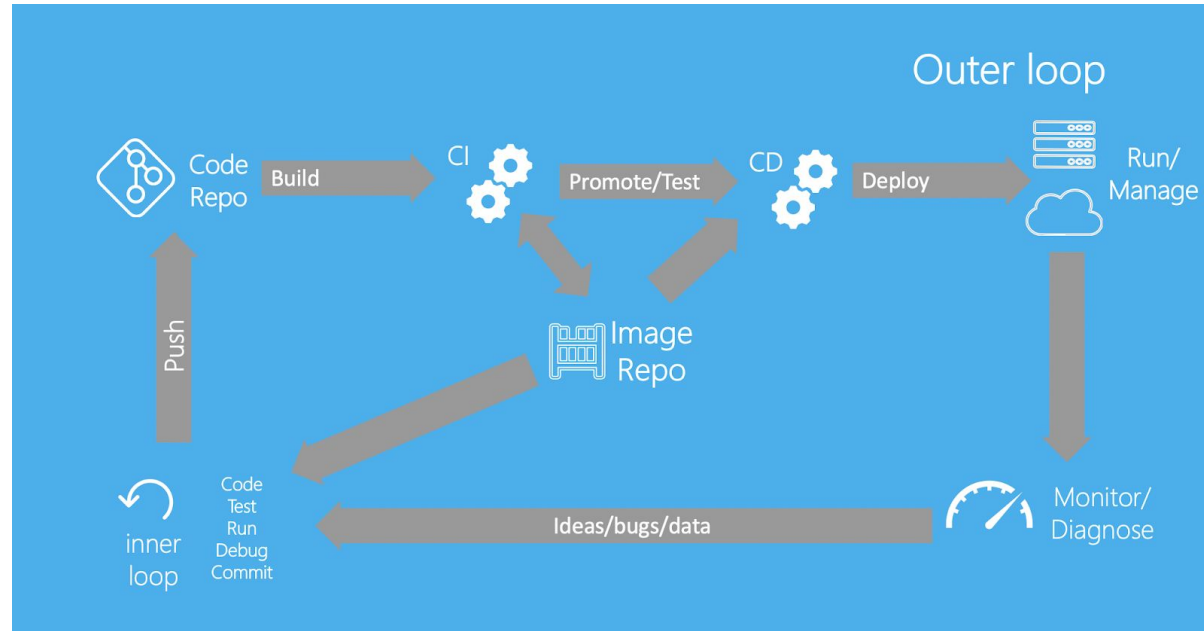
- Un conteneur docker correspond à l'instanciation d'une image (Analogie avec les classes et les objets en POO)
 - Contenu de l'image
 - Exécution de la commande définie dans l'image
 - Un layer supplémentaire (contexte d'exécution)
 - Un ou des réseaux
 - Des volumes

Registry Docker



- un registry Docker permet de stocker plusieurs repositories d'images
 - Chaque repository contient plusieurs images (ou plusieurs tags/versions d'une image)
- Docker Hub (contient de nombreuses applications officielles)
- Implémenter son propre Registry Privé
 - Registry officiel
 - Harbor (Projet CNCF)
 - Portus (OpenSUSE)

Registry : Un outil central pour utiliser des images



Commandes Docker



- **docker help** - Affiche l'aide
- **docker build** - Construit une image
- **docker run** - Exécute une image / Crée un conteneur à partir d'une image
- **docker exec** - Exécute une commande dans un conteneur / Permet de pénétrer dans le conteneur
- **docker pull / push** - Récupère ou envoie des images depuis/vers un registry
- **docker ps** - Affiche les conteneurs présents sur la machine
- **docker image ls** - Affiche les images présentes sur la machine
- ...
- <https://docs.docker.com/engine/reference/commandline/cli/>

Dockerfiles

Principales directives :


- **FROM** - Image à partir de laquelle se base cette nouvelle image
- **RUN** - Exécute des instructions lors du build de l'image
- **CMD** - Commande exécutée lors de l'instanciation de l'image en conteneur
- **EXPOSE** - Exposition d'un port
- ...

Chaque directive donne lieu à un nouveau layer / une image intermédiaire

Documentation complète :

<https://docs.docker.com/engine/reference/builder/>

Docker via l'interface graphique

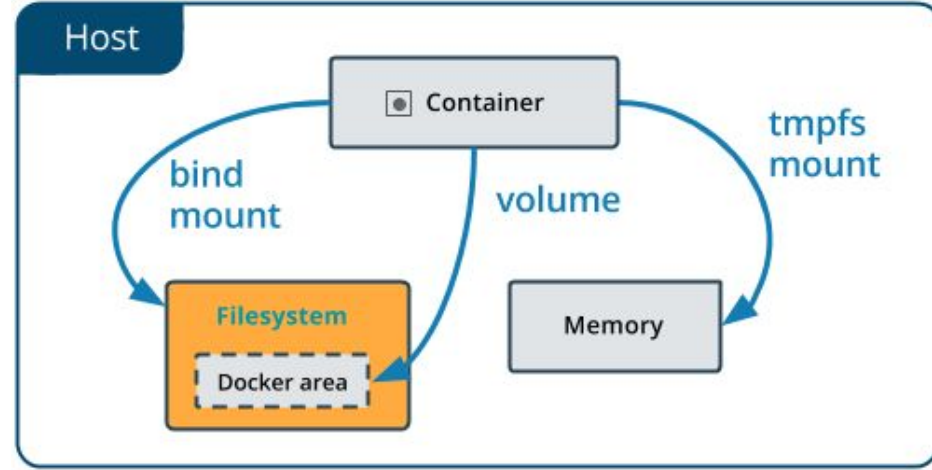
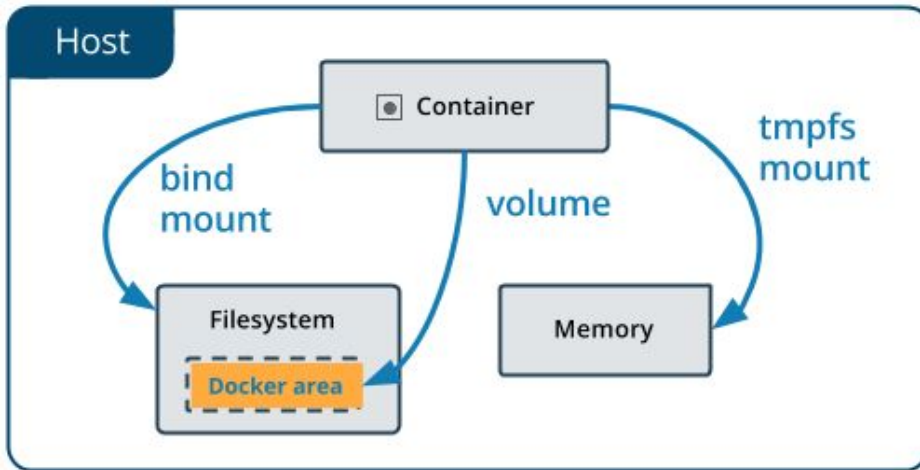
- 
- Portainer.io
 - Docker Desktop
 - Docker Station
-
- Permet d'avoir un aperçu plus visuel sur les conteneurs en fonctionnement
 - L'utilisation en ligne de commandes dispose en général de plus d'options que les interfaces graphiques

Volumes et bonnes pratiques

Les volumes Docker

On peut distinguer deux manières de partager des fichiers dans Docker pour deux besoins différents :

- Besoin de sauvegarder des données (data-volume)
- Besoin de modifier des fichiers pendant l'exécution du conteneurs (bind-mount)



Bonnes pratiques



- Un seul service par conteneur. Un conteneur n'est pas une VM
- Optimiser la création des images Docker par couche
- Supprimer tout ce qui n'est pas utile dans le conteneur (fichier / programmes)
 - Réduit la surface d'attaque pour votre application
 - Diminue la taille de votre image
- Gérer les tags des images avec le Semantic Versionning
- Utiliser autant que possible les images officielles & publiques disponible sur le Docker Hub

Docker-Compose

Organiser l'exécution de ses conteneurs

- Permet de décrire la configuration d'un ou plusieurs services Docker ainsi que :
 - Leurs volumes
 - Leurs réseaux
- Format de fichiers YAML
- <https://docs.docker.com/compose/compose-file/>

Ressources Supplémentaires

- <https://cloud.google.com/architecture/best-practices-for-building-containers>
- <https://cloud.google.com/container-registry/docs/container-best-practices>
- <https://www.portainer.io>