

## II. Containerization

### **What is a container?**

A container is a sandboxed process running on a host machine that is isolated from all other processes running on that host machine. "

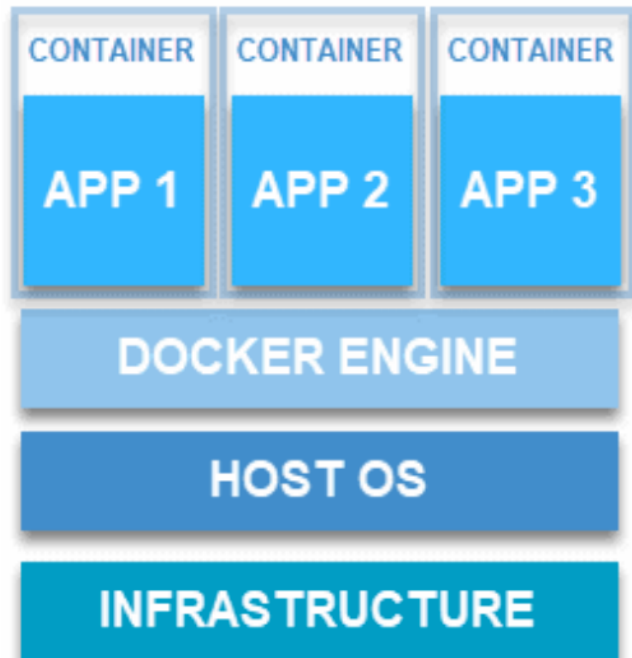
cf. [Docker : What is a container ?](#)

## Principe d'un container

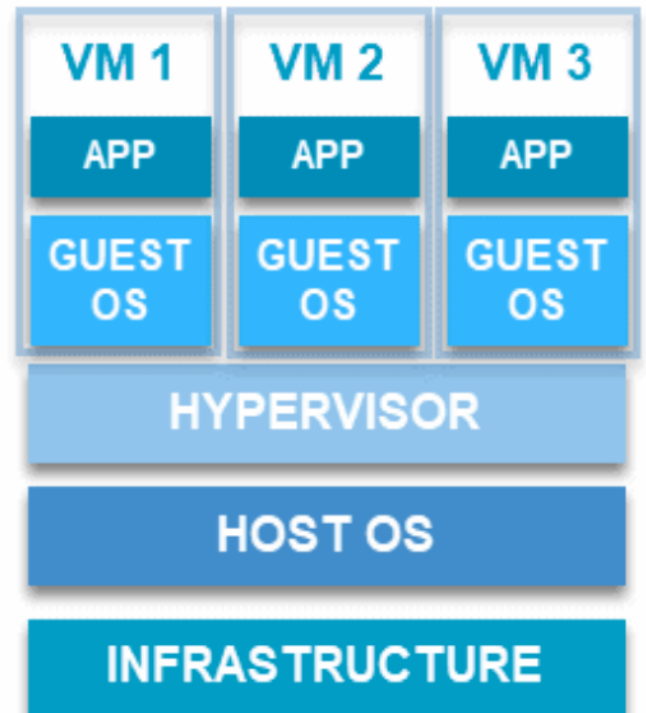
Un container est le package relativement isolé de son OS host regroupant une application et l'ensemble de ses dépendances.

## Différence entre containers et VMs

# DOCKER CONTAINERS



# VIRTUAL MACHINES



cf. [Samuel Delepouille - BUT2 - Emulation / Virtualisation / Conteneurisation](#) "

## Avantages des containers

- La **Portabilité** et la **Répétibilité**
  - Une fois défini, une image permet de créer un container identique :
    - Quelque soit l'host
    - Autant de fois que nécessaire
    - De manière fiable et répétable
- L'**Isolation** des dépendances
  - Au delà de l'isolation des applications en elles mêmes, le principale avantage découlant de l'isolation des containers et l'isolation des dépendances, même si X application dépendent de X versions différentes de la même librairie, il n'y a pas de risque d'incompatibilité, chaque application étant isolé dans un container
- Les **Performances**

- Un container est de part sa conception plus léger qu'une VM car il n'embarque pas d'OS complet
- La **Facilité** de gestion

# En pratique

Nous allons réaliser deux exercices simples pour réappréhender les concepts de base de Docker :

## Exercices :

• II. a) Dockerfile

• II. b) Docker Compose

---

Revision #4

Created 18 October 2023 15:44:48 by Victor BADIN-LIMOU

Updated 15 November 2024 09:11:24 by Victor BADIN-LIMOU