Partie Cours

Liens Utiles

> Cette page a pour objectif de regrouper l'ensemble des liens exploités lors du cours. Merci de signaler tout oubli. "

AlternativeTo

AlternativeTo vous permet de chercher des alternatives à vos sites ou logiciels préférés.

Awesome-Selfhosted

awesome-selfhosted est une liste de solutions gratuites et self-hosted. Il existe également une version web.

Conventionnal Commits

Conventionnal Commits est un standard d'écriture de commit de versionning, ce standard sera à suivre lors du projet.

Dagger.io

Dagger io permet la création et l'utilisation d'une CI/CD en locale, s'appuyant sur les containers.

Docker

Docker est un moteur de containerization

Docker Hub

Docker Hub est la registry par defaut de Docker. On y trouve des images a déployer tel quel, ou a exploiter comme base pour ses propres images.

I. Build Automation

> Build automation is the process of automating the creation of a software build and the associated processes including: compiling [...], packaging [...], and running automated tests. "
>
> cf. **Wikipedia : Build Automation**

Le principe de l'automatisation de la construction logiciel n'est pas vraiment récent, quand on vois qu'un outil comme Make date de 1976. Dans cette première partie du cours, ne reviendrons aux bases et exploireront notamment la compilation du langage C ainsi que de son automatisation avec Make.

Exercice

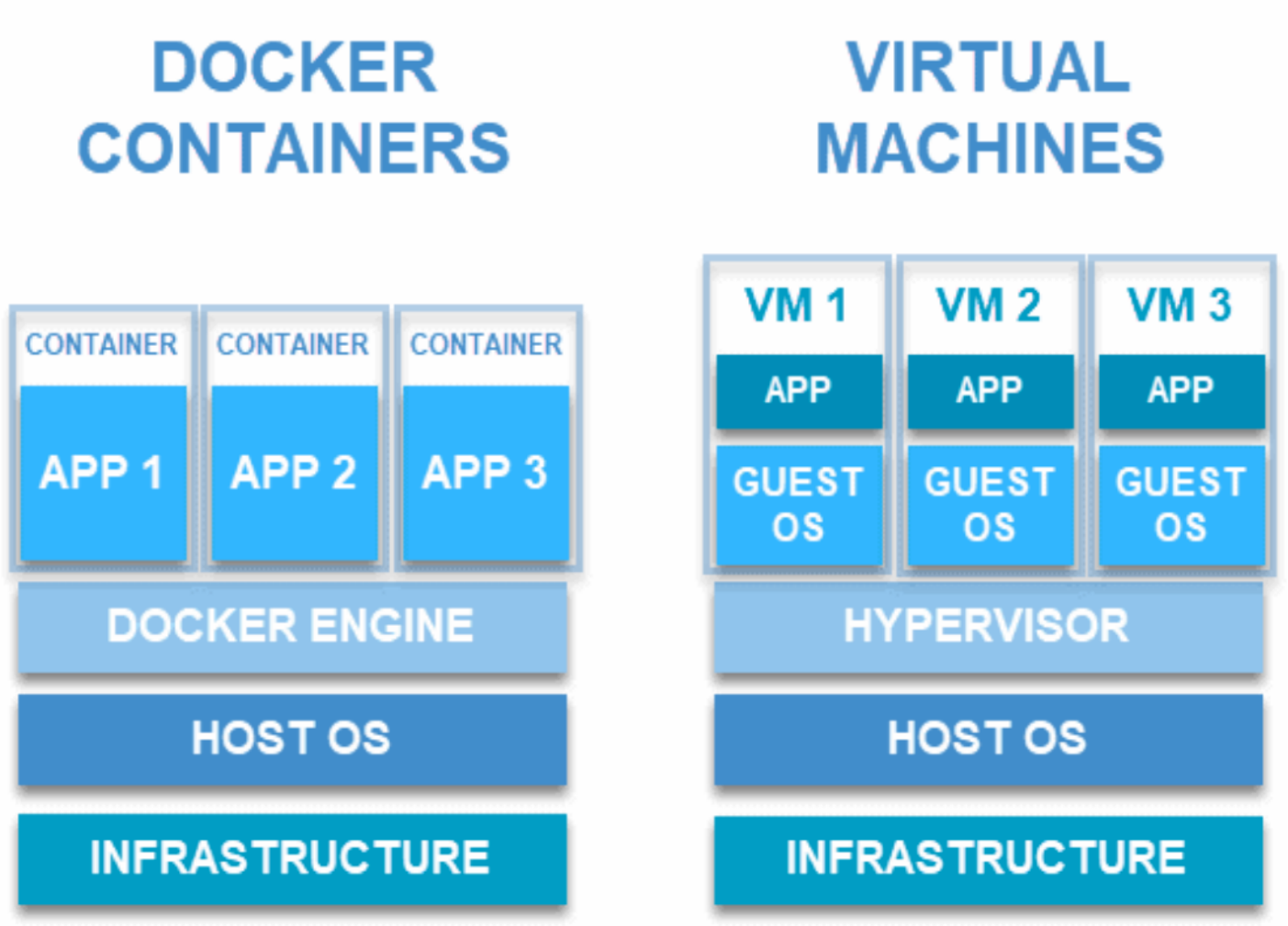
II. Containerization

> ## 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



> cf. Samuel Delepoulle - 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 délà 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 légé qu'une VM car il n'embarque pas d'OS complet
- La **Facilité** de gestion

En pratique

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

Exercices :