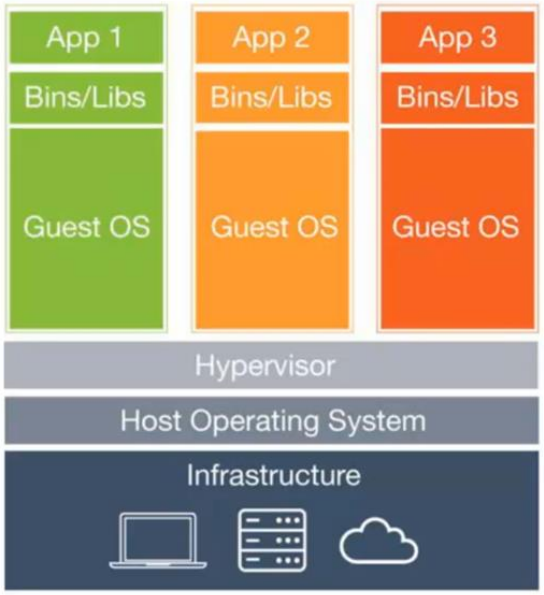
# Les conteneurs

## 1. Représentation de la virtualisation

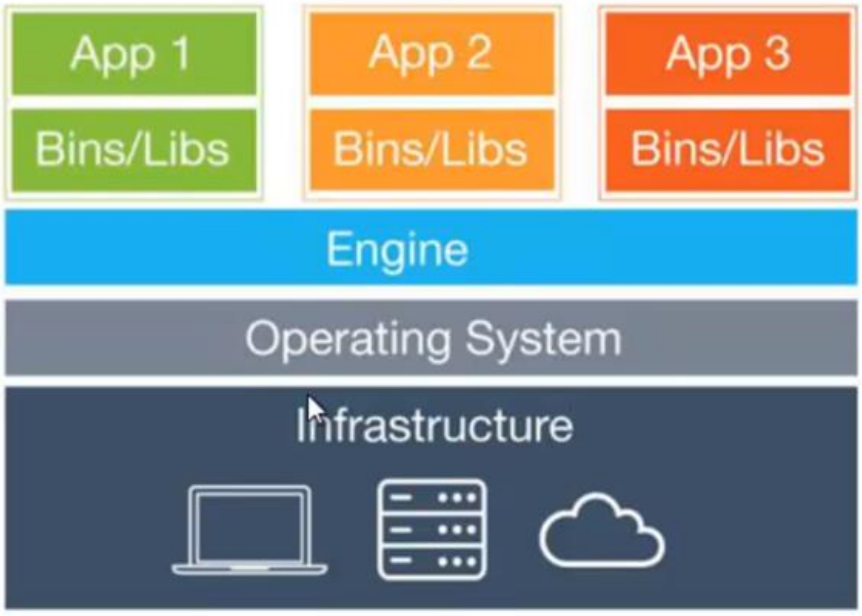



Rappel : dans chaque OS nous allons trouver 2 parties :

- Le kernel (le système en lieu même, le kernel space)
- Les applications (espace utilisateur, user space)

Un des avantages de la virtualisation : possibilité d'installer n'importe quel OS invité quelque soit le système hôte.

## 2. Le modèle des conteneurs



Les conteneurs ne sont pas des machines virtuelles, et n'ont pas de rapport direct avec la virtualisation. Nous parlerons plutôt d'isolation

Nous retrouvons ici notre infra physique, notre OS hôte (Linux en l'occurrence) et un moteur de conteneurisation (nommé engine), typiquement Docker pour ce qui nous intéresse

A partir de ce moteur nous allons pouvoir créer des conteneurs, comparable aux machines virtuelles, la grande différence est qu'il n'est pas nécessaire ici d'installer d'OS invité, mais uniquement ce qui sera indispensable au user space.

Ces conteneurs vont partager le kernel space de l'OS hôte tout en étant totalement isolés.

Ils vont donc utiliser directement le matériel physique et non pas un hardware virtualisé.

Cela possède les avantages suivants :

- Une optimisation maximum : seuls les binaires et librairies indispensables à l'application sont déployés
- Une plus grande légèreté : pas besoin d'installer et de faire tourner un OS invité comme pour la virtualisation
- Une sécurité entre les applications : tout est isolé
- Une rapidité de déploiement : il est très simple de déployer des milliers de conteneurs, ce sera beaucoup moins évident pour des machines virtuelles

Schématiquement, nous installerons des miniOS les uns à coté des autres, isolés et partageant le même kernel space

Seules les cartes réseaux sont virtualisées, nous y reviendrons.

Nous trouverons 2 types de conteneurs :

- Ceux contenant des OS (debian, ubuntu, …)
- Ceux contenant des applications (MySQL, Glpi, Wordpress, …) : ils contiennent déjà des miniOS (quelques centaines de MB) y incluant les applications désirées

## 3. Les namespaces

Le noyau Linux possède des fonctionnalités qui permet d'isoler :

- Process Namespace : Chaque processus d'un conteneur possède un numéro de process qui sera différent des process des autres conteneurs et du système hôte. C'est ainsi que se déroule l'isolation
- Network Namespace : Les cartes réseaux pourront être isolées, chaque conteneur aura sa propre carte réseau eth0, eth1, … qui sera différente des cartes réseaux des autres conteneurs
- Mount Namespace : Chaque conteneur aura son propre filesystem, chacun aura son arborescence / différente des autres conteneurs
- UTS Namespace : Permettant d'avoir des noms d'hôte et des noms de domaine différents pour chaque conteneur
- User Namespace : les root de chaque conteneur seront tout à fait différents les uns des autres et isolés du root de la machine hôte.

Les namespaces sont une fonctionnalité du noyau linux, et non pas de docker. Ce dernier les exploite simplement

### 4. Les groupes de contrôle

Ceux-ci permettent d'allouer et de surveiller des ressources. Il sera par exemple possible à un conteneur de refuser l'utilisation d'une carte réseau physique

Ils permettent de plus de réguler les ressources en limitant par exemple la bande passante sur un des conteneurs

Ils permettront aussi d'allouer tant ou tant de mémoire à tel conteneur

Ils permettent enfin de surveiller et mesurer les ressources utilisées (facturation dans le cas d'un cloud par exemple)