

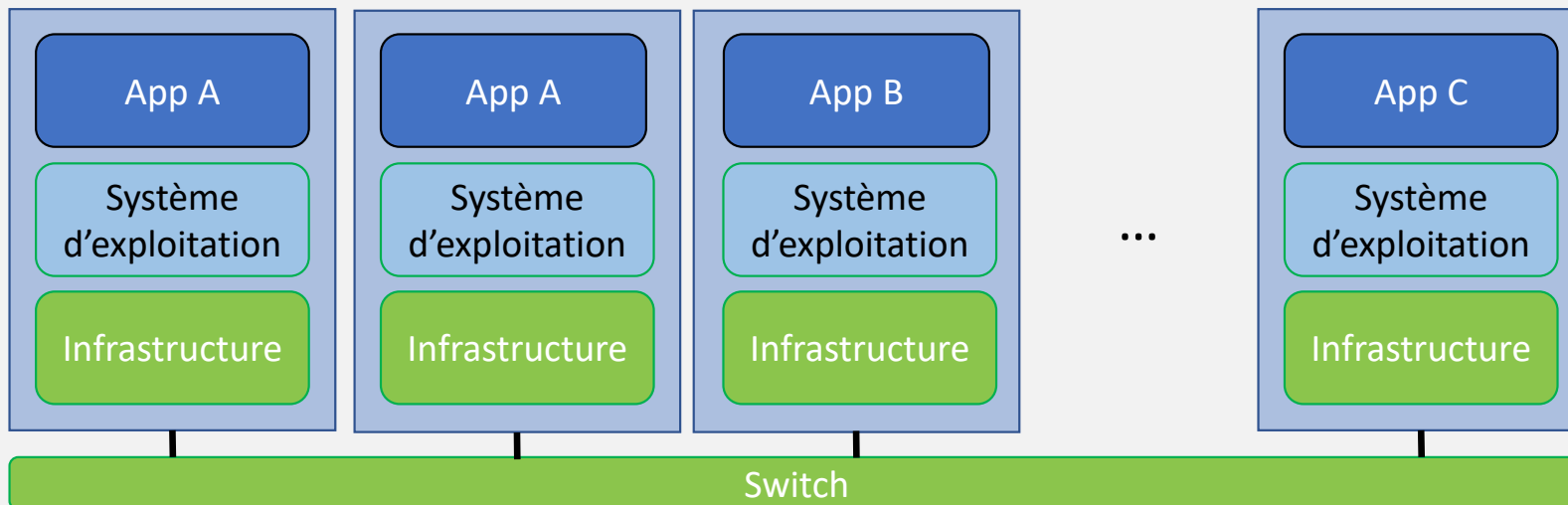
Conteneur



Modèles de déploiement applicatif

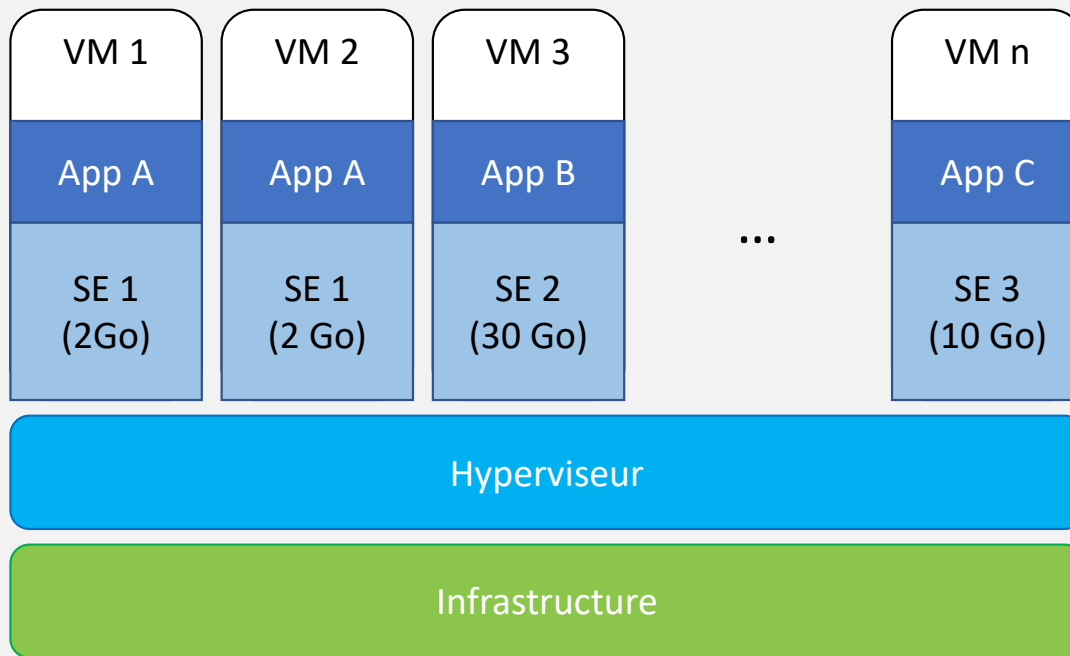
- Modèle 1 : Physique
 - Les applications et les services s'exécutent sur une machine physique
- Modèle 2 : Virtualisation
 - Les applications et les services s'exécutent sur des machines virtuelles
- Modèle 3: Conteneurs
 - Les applications et les services s'exécutent sur un moteur de conteneurs

Modèle 1 – Physique



- Une application est installée par machine physique
- Les applications sont isolées
- Coût élevé / perte de ressource
- Complexité de gestion des dépendances
- Long à déployer

Modèle 2 – Virtualisation des machines physiques



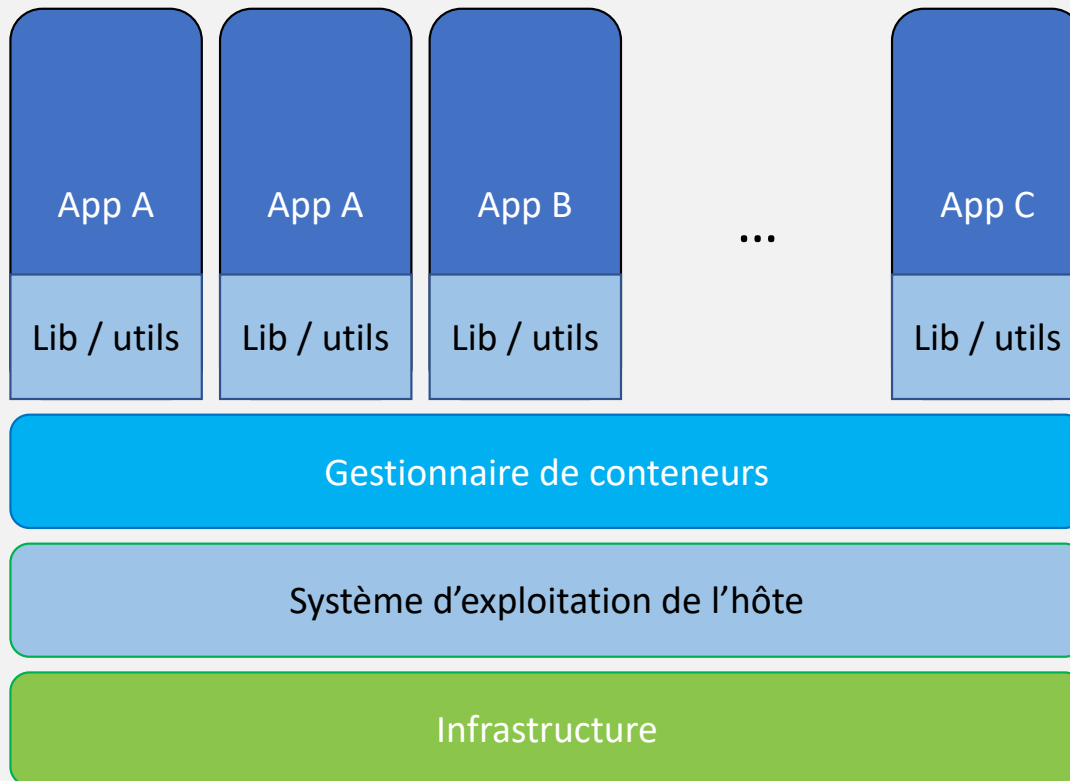
1 hyperviseur par hôte :

- n machines virtuelles (VM) avec des systèmes d'exploitation différents
- Par VM :
 - Un système d'exploitation complet (x Go)
 - Une instance de l'application / service

Idées :

- Déplacer les VMs d'un hôte à un autre
 - Répartir la charge
 - Reprise en cas de défaillances matérielles
- Isoler les applications
 - Si dépassement de ressource : peu d'impacts sur les autres applications
 - Si piratage : reste consigné à la VM
- Mise à l'échelle verticale et horizontale facile

Modèle 3 – Virtualisation des applications / services



1 hôte :

- 1 système d'exploitation
- Plusieurs conteneurs / hôte
- 1 application / conteneur (= instance d'une image)

Idées :

- Déplacer les conteneurs d'un hôte à un autre
 - Répartir la charge
 - Reprise en cas de défaillances matérielles
- Isoler les applications
 - Si dépassement de ressource : peu d'impacts sur les autres applications
 - Si piratage : reste consigné au conteneur
- Mise à l'échelle verticale et horizontale facile
- Léger : le conteneur ne contient que les dépendances de l'application et l'application (quelques Mo), le conteneur ne contient pas le système d'exploitation

Technologies de containerisation

- [Docker](#) est une surcouche qui rend le développement et le déploiement des conteneurs beaucoup plus simple tout en les standardisant. La première version Docker 1.0 date de 2014, mais son adoption par les plus grands comptes de l'informatique en fait déjà un outil de référence.
- [Kubernetes \(K8s\)](#) `hvw#q#p rwxu#grufkhvwdwlrq#gh#
frqwhqhxu#rshq#rxufh#shup hwdqwg#dxwrp dwvhu#h#
gýs#rhp hqw#ol#p lvh#ú#ýfkho#hwd#hvwlrq#ghv#dssdfdwlrqv#
frqwhqhxu#v#ýhv1`

Source :

- [Conteneur \(informatique\) — Wikipédia \(wikipedia.org\)](#)
- [Documentation de Kubernetes | Kubernetes](#)

Docker

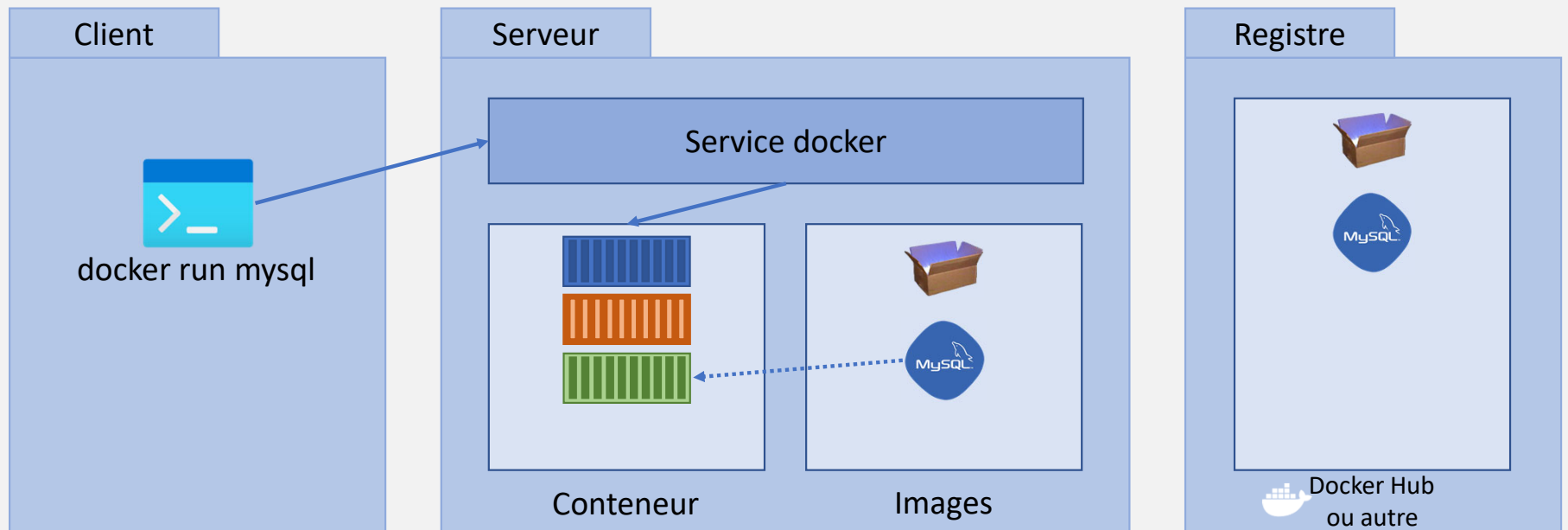
- Les conteneurs de docker s'exécute dans le moteur docker :
 - Docker est très utilisé dans l'industrie
 - Ils sont légers
 - Par défaut, les conteneurs sont isolés de l'hôte
 - Ils ne peuvent pas accéder aux ressources de l'hôte
 - Ils ont tous des adresses IP
- Les images sont organisées en couches d'images :
 - La couche de base est nommée « scratch »
 - Les autres couches sont des images qui se basent sur scratch ou sur la couche précédente, un peu comme de l'héritage en POO

Utilisations

- Déploiement :
 - Léger : ne contient que l'application et ses dépendances
 - Autonome : contient toutes les dépendances
 - Sécurité : isolement du système de fichiers et du réseau
- Développement :
 - Accéder rapidement à un service sans l'installer sur sa machine et risquer de la déstabiliser
 - Dans un environnement complexe, par exemple plusieurs applications qui communiquent ensemble, on peut les déployer et se concentrer sur l'application que l'on développe sans passer beaucoup de temps à installer des pré-requis

Docker – run

```
Docker run [-rm] [autres options] <nom_image> [arguments]
```



Docker – Quelques commandes

- `Docker run [-rm] [autres options] <nom_image> [arguments]`
 - Télécharge l'image si elle n'existe pas en local
 - Créer un conteneur à partir de l'image et l'exécute
- `docker ps [-a]` : affiche les conteneurs actifs (-a: ou tous)
- `docker image <cmd>` (gestion des images):
 - `ls` : liste les images locales
 - `rm <nom_image>` : supprime une image locale si possible
 - `history <nom_image>` : affiche l'historique d'une image (commandes)

Pourquoi plusieurs éditions ?

- Donc Docker n'est pas qu'un logiciel permettant de lancer des applications dans des conteneurs.
- Il ne s'agit pas de virtualisation, mais de conteneurisation.
- C'est un environnement complet, qui évolue TRÈS TRÈS vite, il faut donc s'assurer d'utiliser la version la plus récente existante.
- S'utilise en mode console dans Linux, MacOS, Windows, WSL. Ou encore dans Docker Desktop.
- Et Docker Desktop ?
 - Docker Desktop est une application pour les machines MacOS et Windows permettant de créer et de partager des applications et des microservices conteneurisés.
 - Gestion graphique des images, volumes.
 - comprend des outils pour les développeurs, Kubernetes et la synchronisation des versions avec les moteurs Docker de production.

Docker Hub c'est quoi ?

- [Docker Hub - Docker](#)

- Selon Docker : service pour trouver et partager des images de conteneurs avec votre équipe et la communauté Docker.
- Les utilisateurs ont accès à des dépôts publics gratuits pour stocker et partager des images ou peuvent choisir un plan d'abonnement pour des dépôts privés.
- Vous devez avoir un compte.
- Utilisé pour faire des pull.