

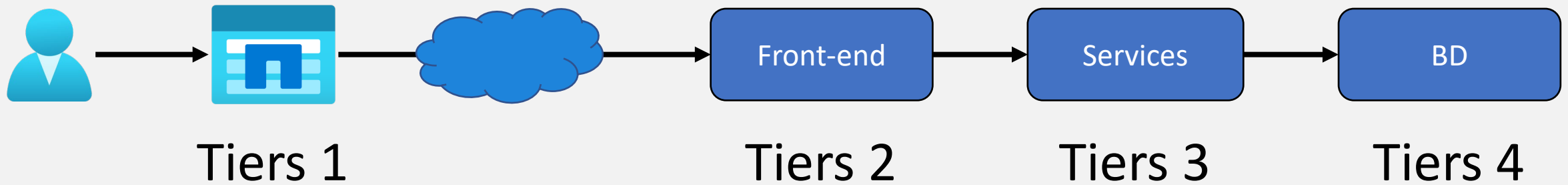
Conteneur



Objectifs

- Modèles de déploiement
- Comprendre l'idée des conteneurs
- Introduction à Docker

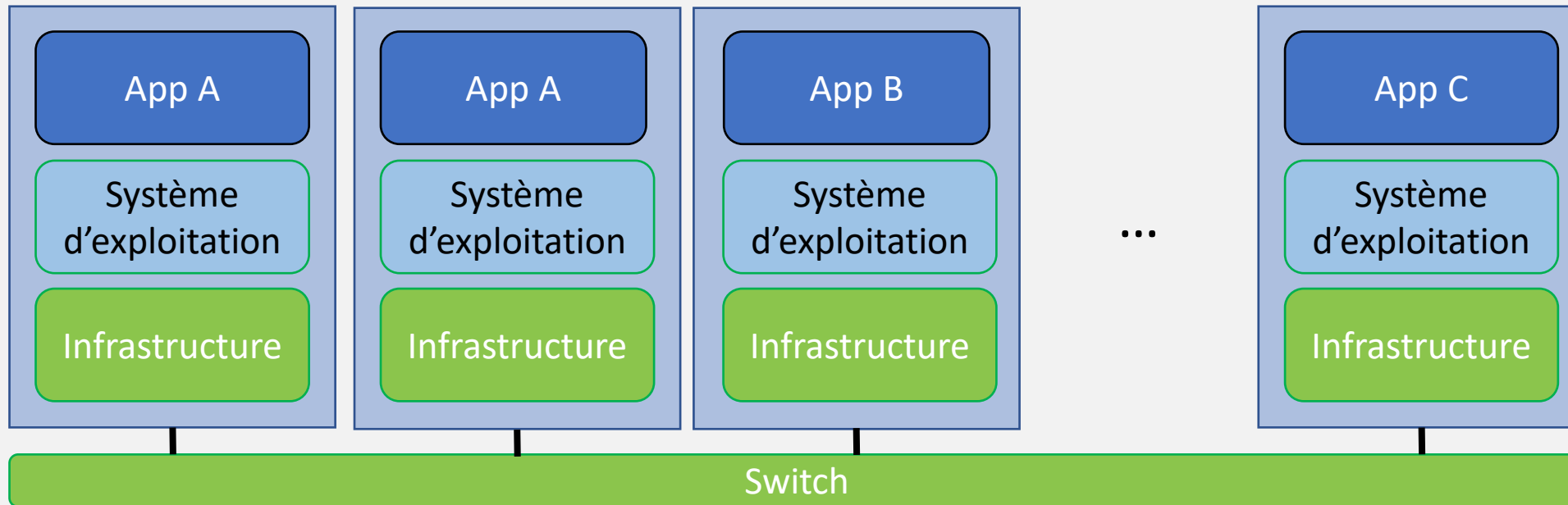
Modèle d'application utilisée en exemple (4-tiers)



Modèles de déploiement applicatif

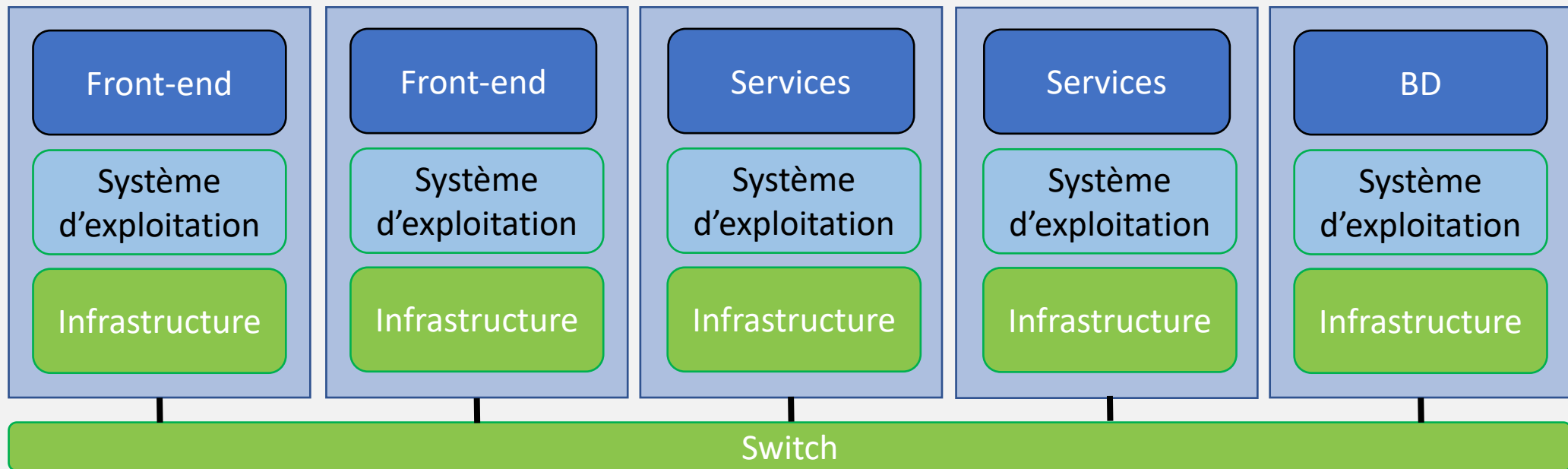
- Modèle 1 : Physique
 - Les applications s'exécutent sur une machine physique
- Modèle 2 : Virtualisation
 - Les applications s'exécutent sur des machines virtuelles
- Modèle 3: Conteneurs
 - Les applications s'exécutent sur un moteur de conteneurs

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

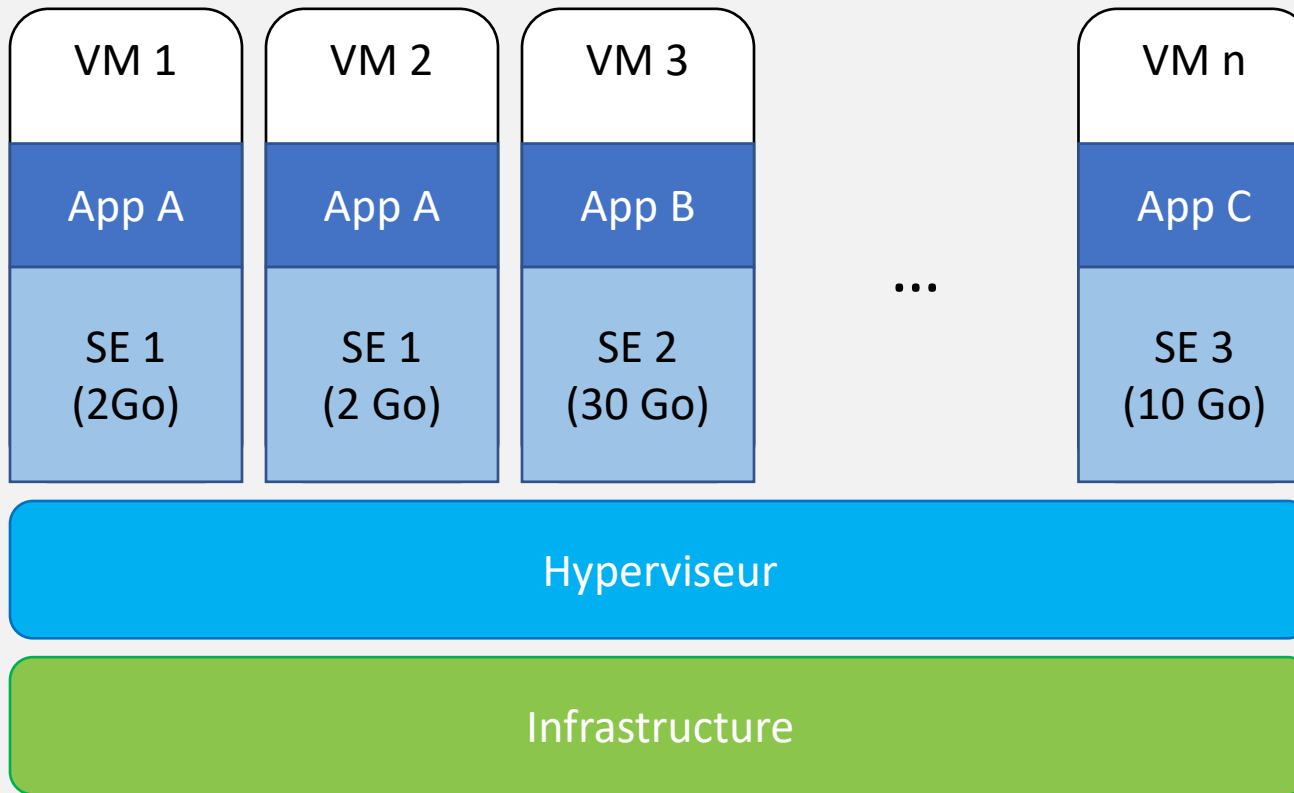
Modèle 1 – Physique



Automatisation :

- Scripts (Puppet, chef, ansible)
- Packages
- Etc.

Modèle 2 – Virtualisation des machines physiques



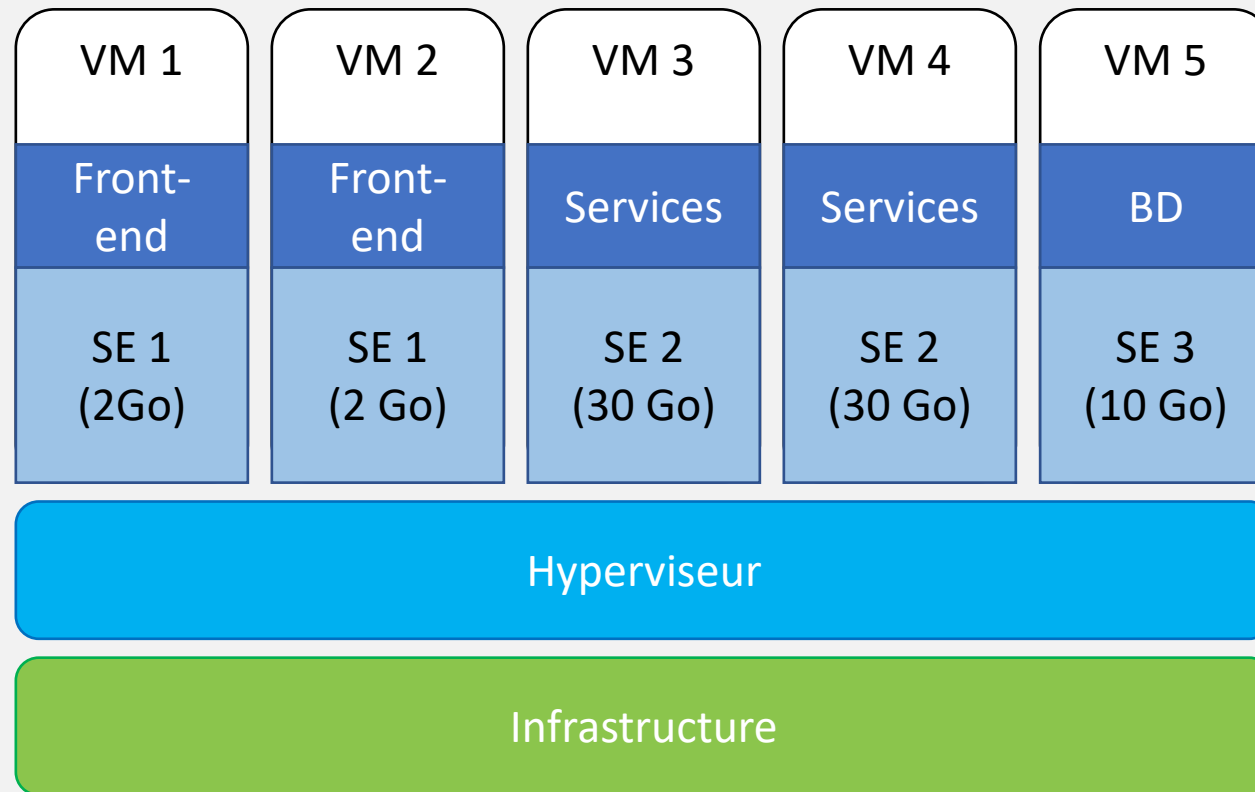
1 hyperviseur par hôte :

- 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

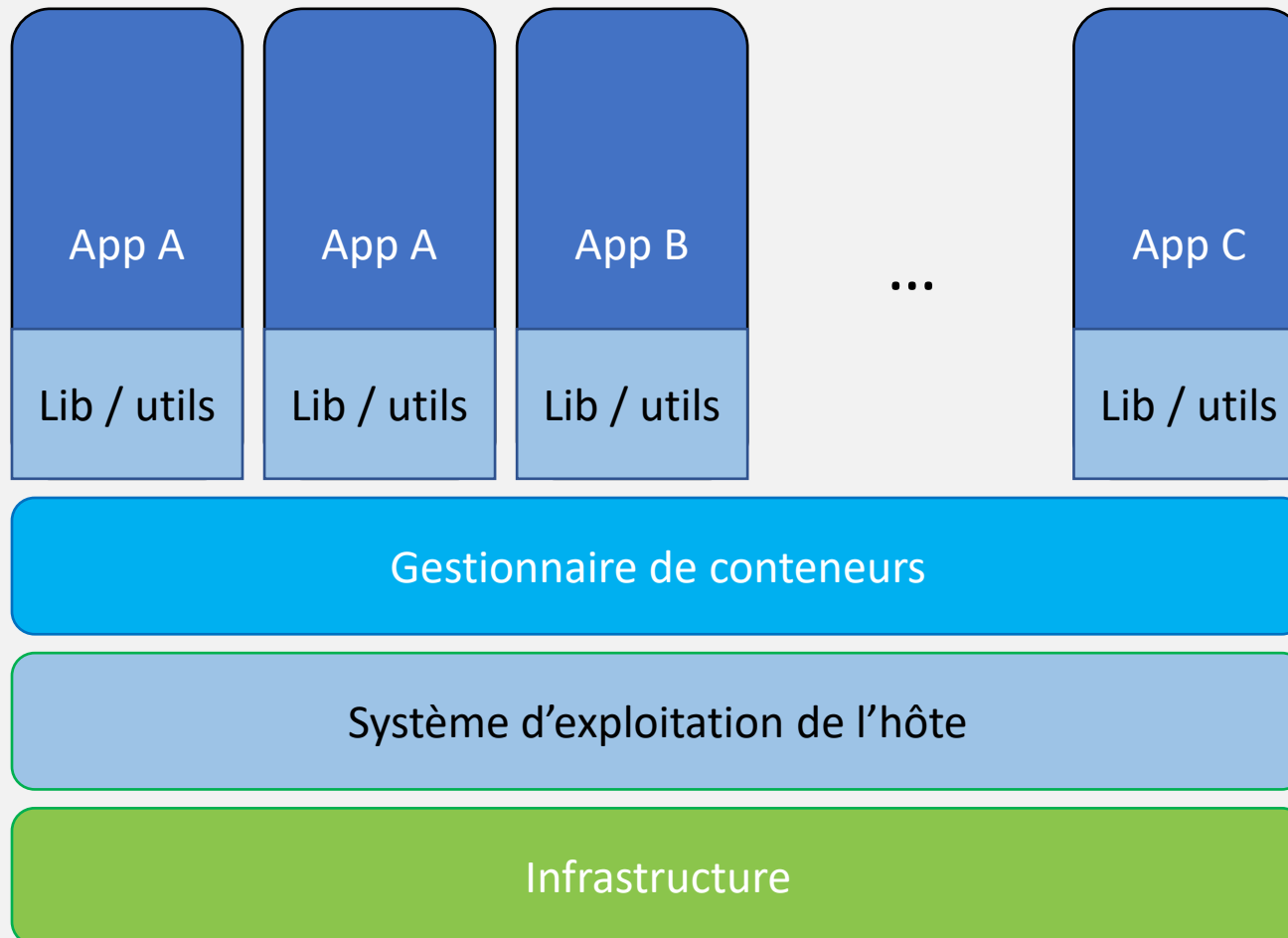
Idées :

- 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 2 – Virtualisation des machines physiques



Modèle 3 – Virtualisation des applications / services



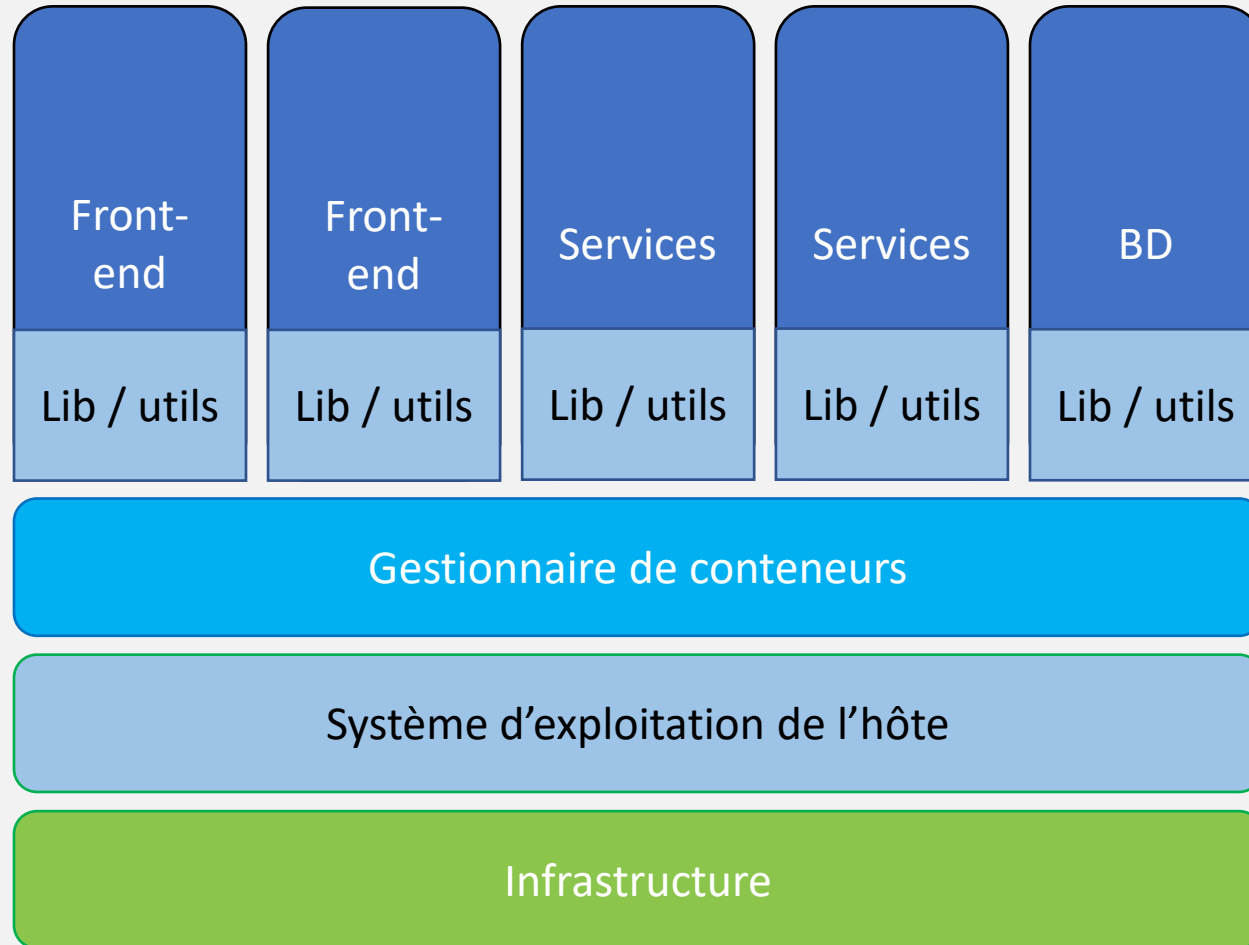
1 hôte :

- 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

Modèle 3 – Virtualisation des applications / services



Conteneur

- Une image de conteneur contient une application avec toutes ses dépendances
- Une image de conteneur est donc une façon de distribuer une application
- Un conteneur est une instance de conteneur
- Un conteneur est un tout (unité) indépendant (applicatif et système d'exploitation)
- Un conteneur démarre rapidement

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

Docker – commandes de bases

- `docker --version`
 - Permet de vérifier le bon fonctionnement
 - Avoir les information de version
 - A utiliser avec ou sans `--` (donne un résultat différent)
- `docker INFO` : affiche plus de détails concernant le serveur Docker (Docker Engine)
 - Nombre de conteneur actuellement en fonctionnement
 - Le nombre d'images stockées dans Docker
- `docker` : affiche l'aide avec toutes le commandes
 - Deux section Management Commands et Commands
 - `-- help` aussi disponible

Docker – Quelques commandes

- `docker run [-rm] [autres options] <nom_image> [arguments]`
 - Télécharge l'image si elle n'existe pas en local
 - Crée 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 local 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.
- 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 ?

- Selon Docker : service pour trouver et partager des images de conteneurs avec votre équipe et la communauté Docker.
- Vous devez avoir un compte.
- Utilisé pour faire des pull.