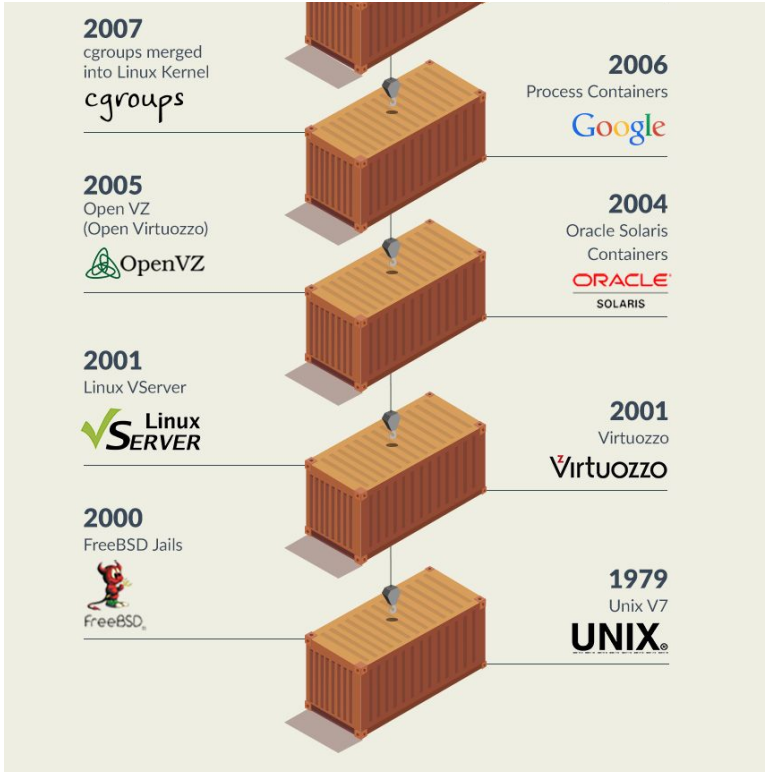
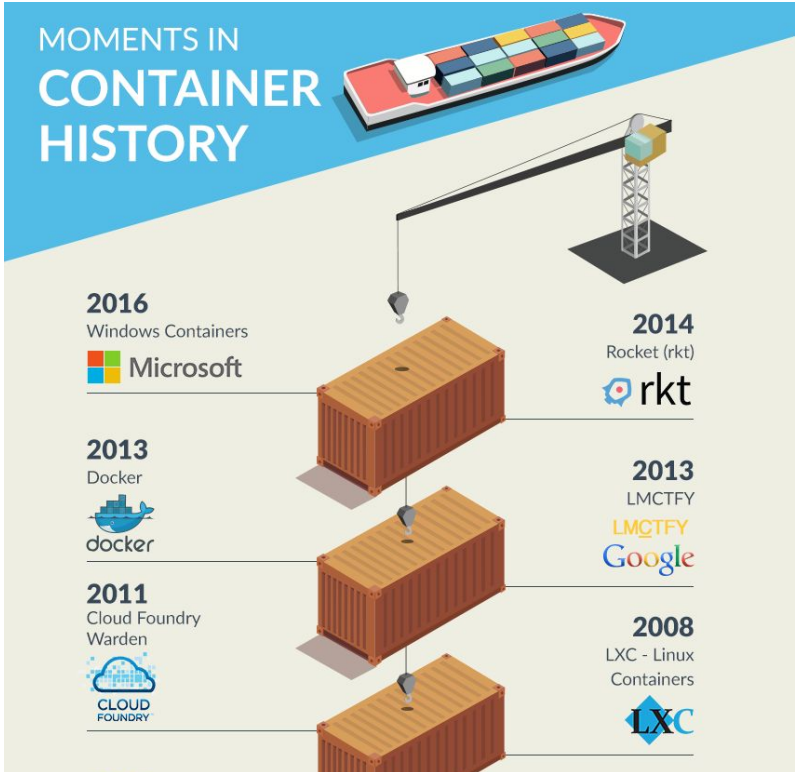
CentraleSupélec

TP DOCKER

Initiation aux conteneurs et à leur déploiement

Historique containers



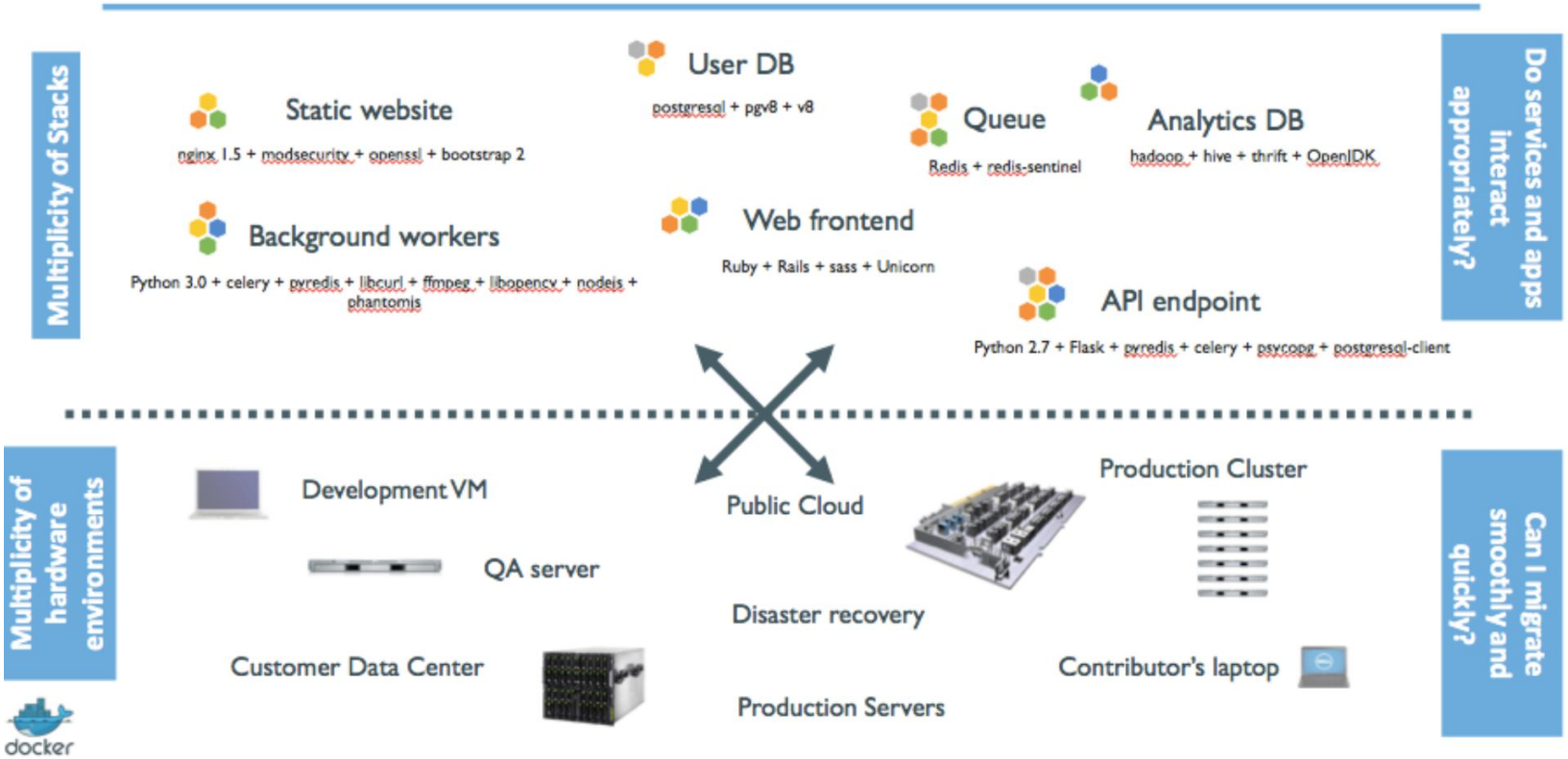


https://www.plesk.com/blog/business-industry/infographic-brief-history-linux-containerization/

Pourquoi Docker ?

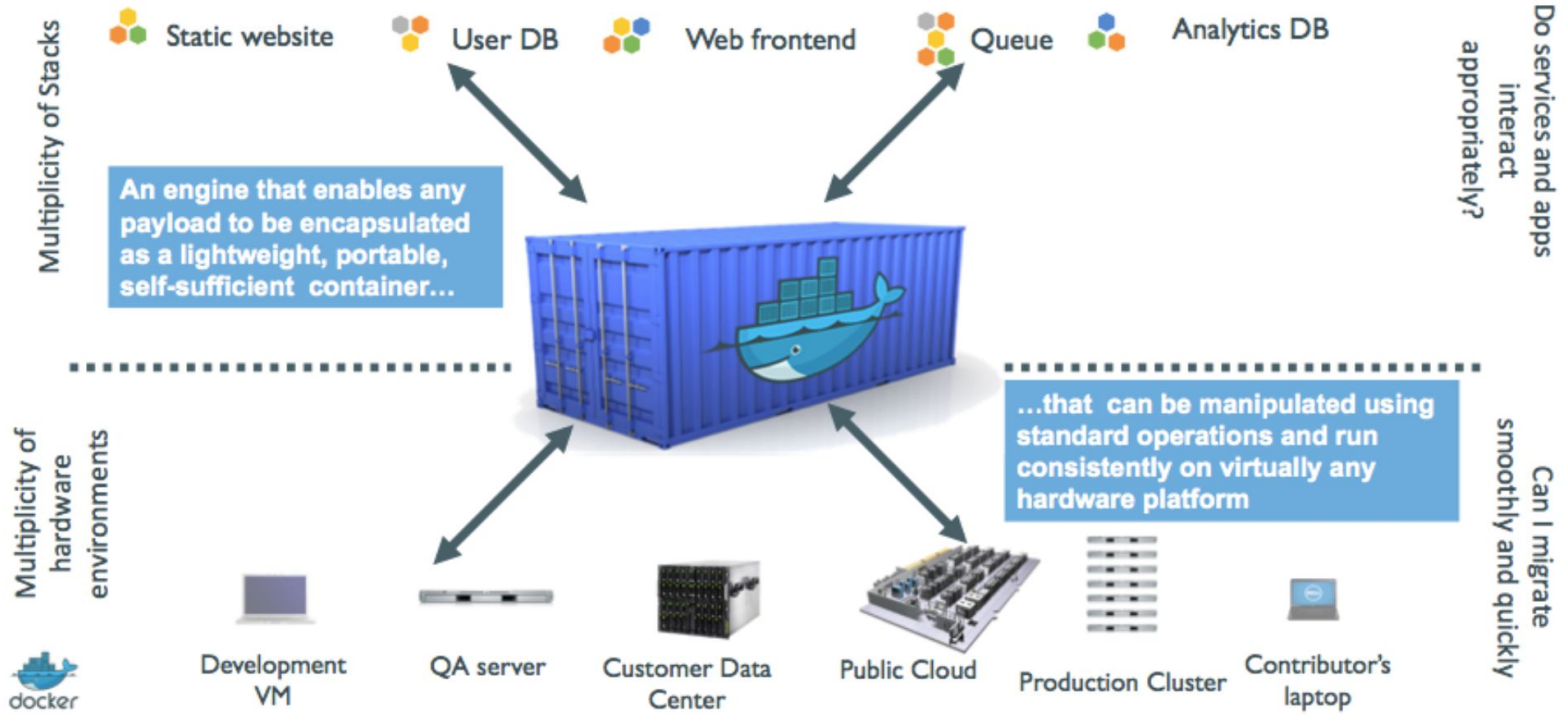
- L’industrie du logiciel a évolué
- Avant :
  - Applications monolithes
  - Cycle de développement long
  - Environnement unique
  - évolution de la charge prédictible et lente
- Maintenant :
  - Services découplés (micro services)
  - évolutions itératives et rapides
  - environnements multiples
  - évolution de la charge très rapide
- Evolution des déploiements
  - Langages, frameworks, bases de données : multiplicité importante
  - Cibles multiples : poste de DEV, pre-prod, QA, staging, production (onPrem, Cloud, Hybrid)

Pourquoi Docker ?
Le problème du déploiement



https://container.training/intro-selfpaced.yml.html#24

Pourquoi Docker ?
Un container pour packager les applciations

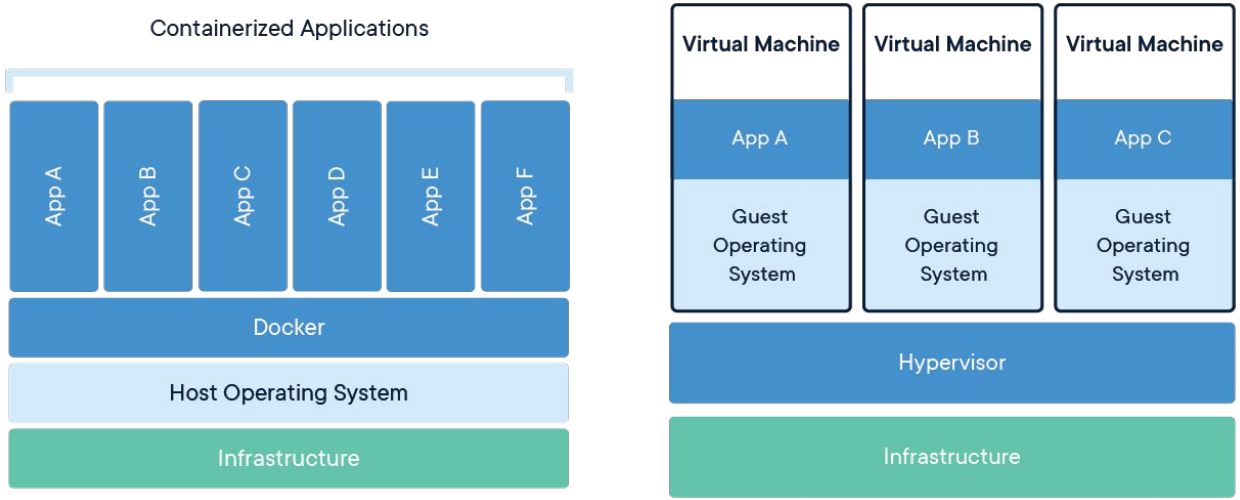


https://container.training/intro-selfpaced.yml.html#24

CentraleSupélec

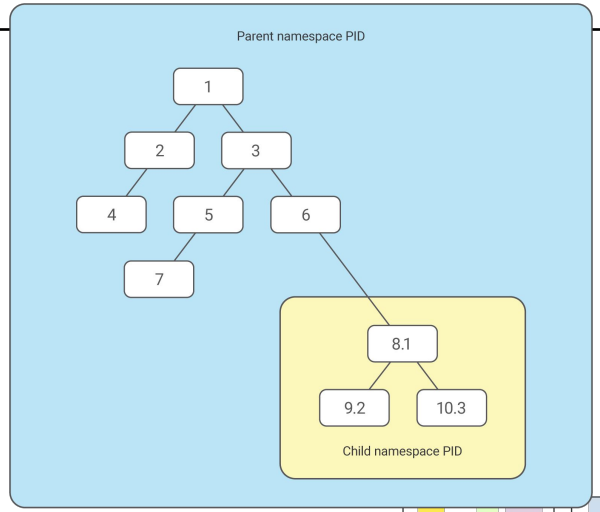
DOCKER - PRINCIPES DE FONCTIONNEMENT

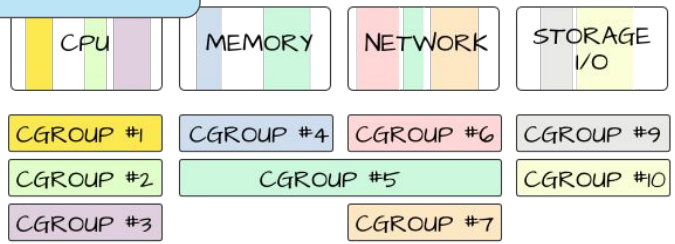
les containers vs VMs

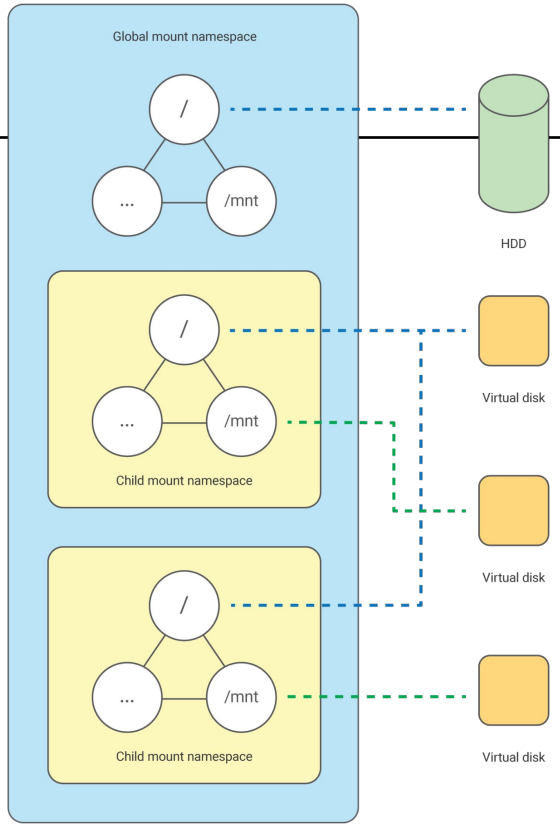


- Un Hyperviseur fait croire à des machines virtuelles qu'elles s'exécutent sur du vrai matériel
- Chaque Machine Virtuelle démarre avec un Système d'exploitation qui lui est propre (BSD/Linux, Windows, etc.)
- VMs: plus lourd (temps de boot OS, …) à l'installation, au scaling à la maintenance (update)
- **CONTAINER : Partage du noyau Linux** de l'hôte

les containers : Isolation



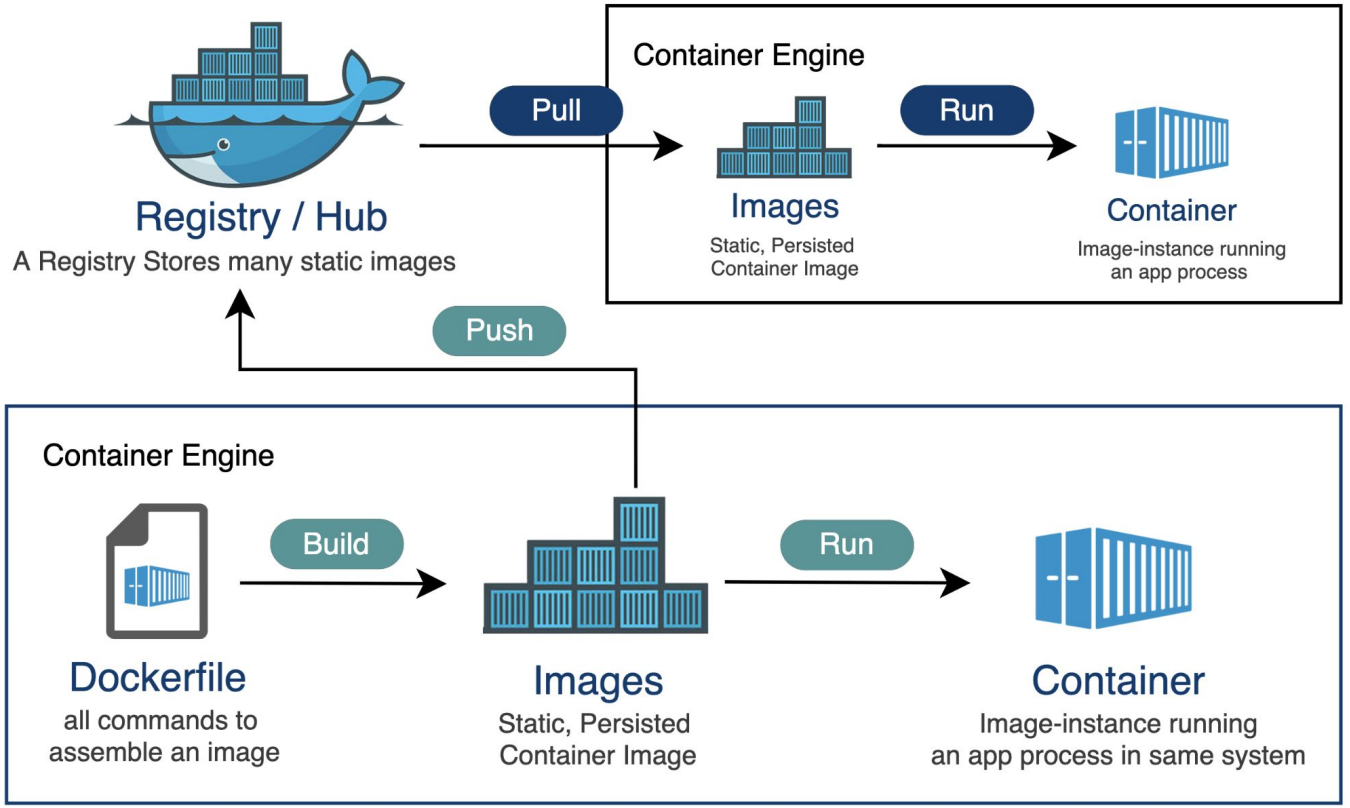




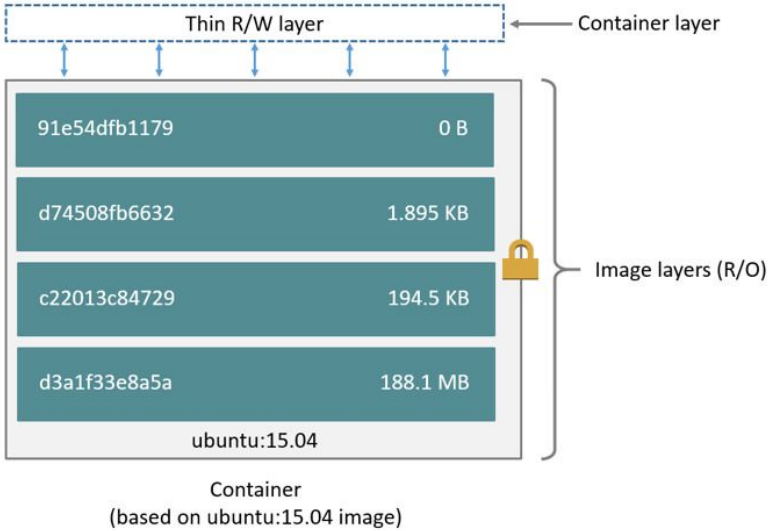
namespaces : isolation des process, du network, des File Systems (initié avec chroot)
control groups (cgroups) : permet d'allouer des quotas sur l'utilisation CPU, RAM, réseau, … pour les processus

https://selectel.ru/blog/en/2017/03/09/containerization-mechanisms-namespaces/

les containers : construction, stockage et runtime

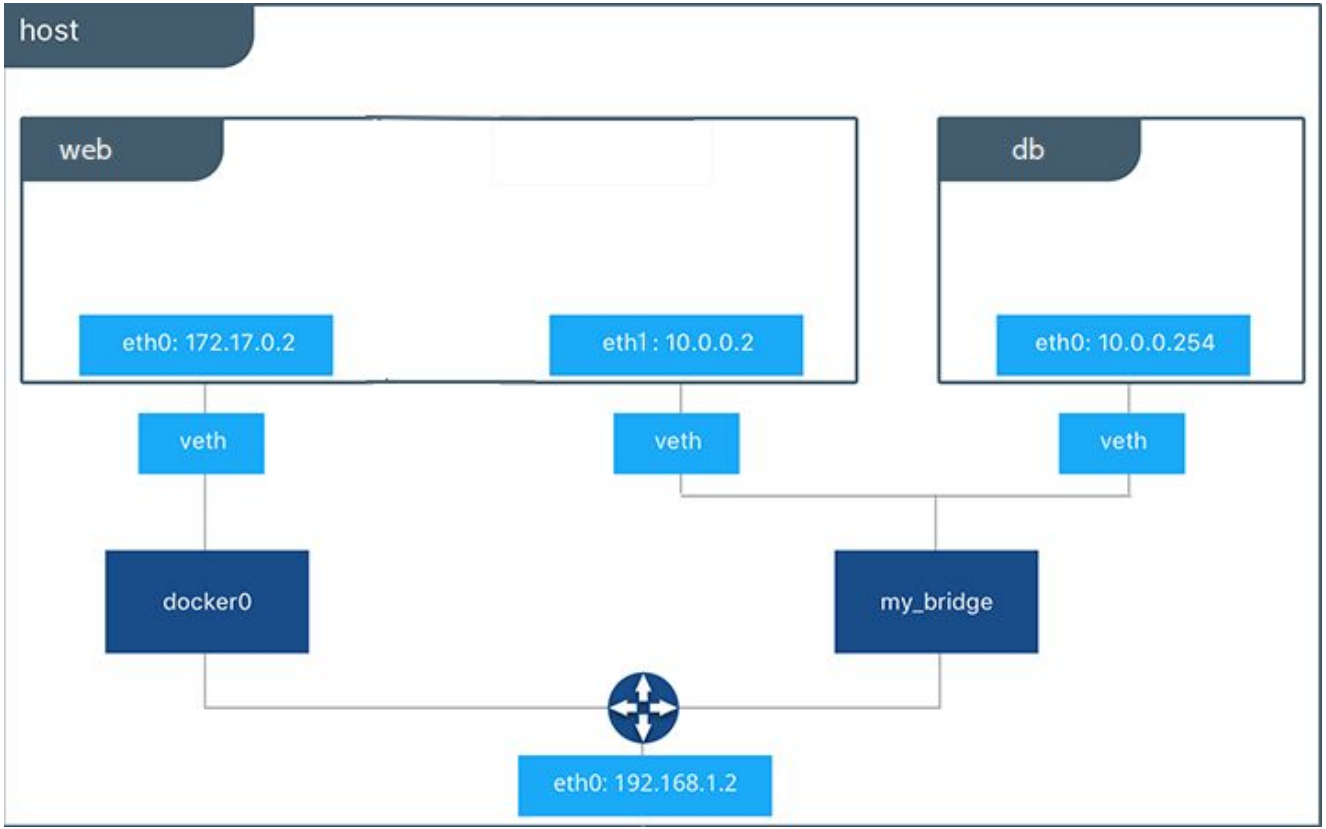


les containers : Image



Une représentation de l'environnement à exécuter (container): système de fichier et de son contenu, de paramètres, de variables d'environnement

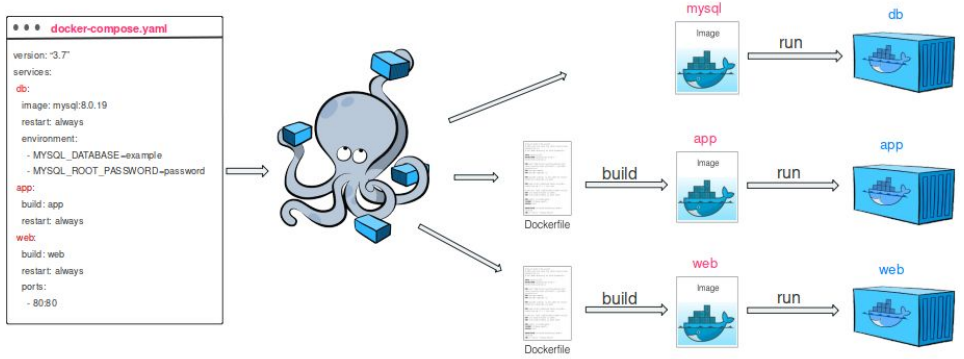
Network



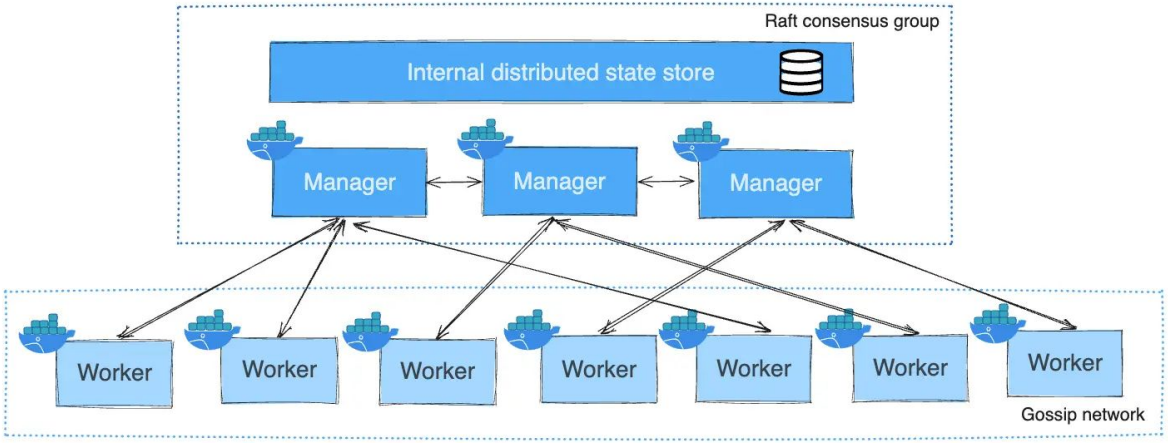
https://docs.docker.com/engine/tutorials/networkingcontainers/

Docker compose / Docker Swarm

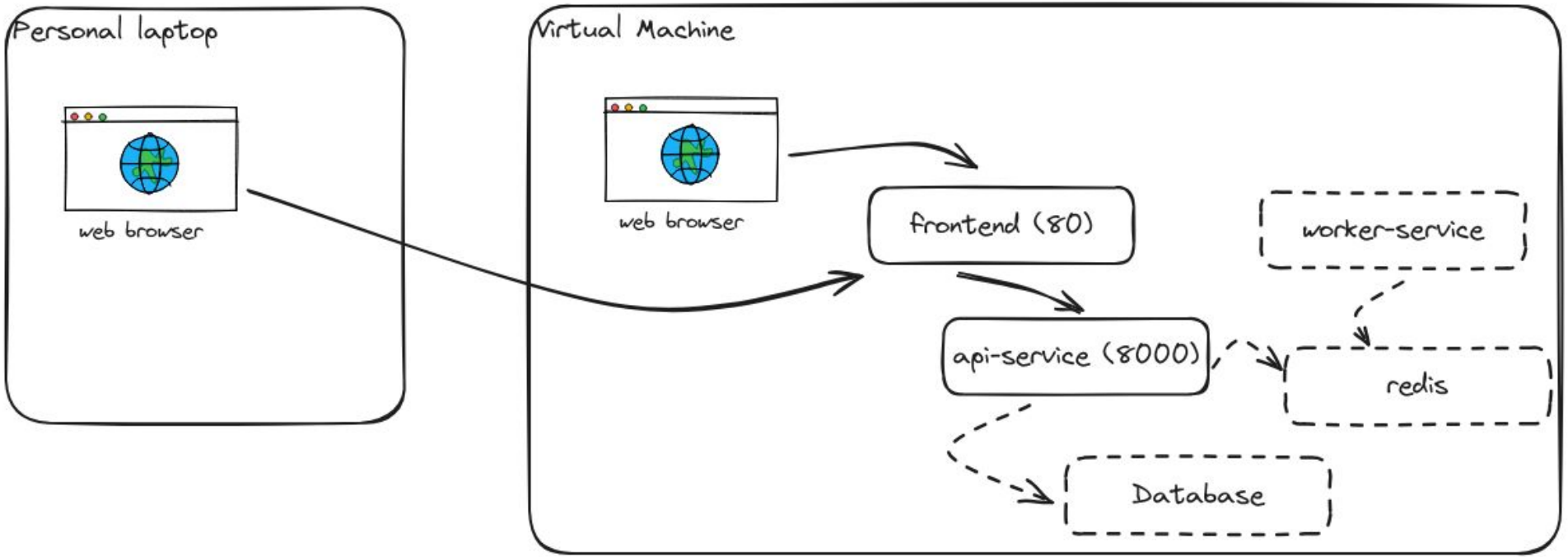
Compose : pour du prototypage (limitation à un seul noeud - pas de HA)



SWARM - préférer Kubernetes désormais



Questions


Personal laptop
web browser
Virtual Machine
web browser
frontend (80)
worker-service
api-service (8000)
redis
Database