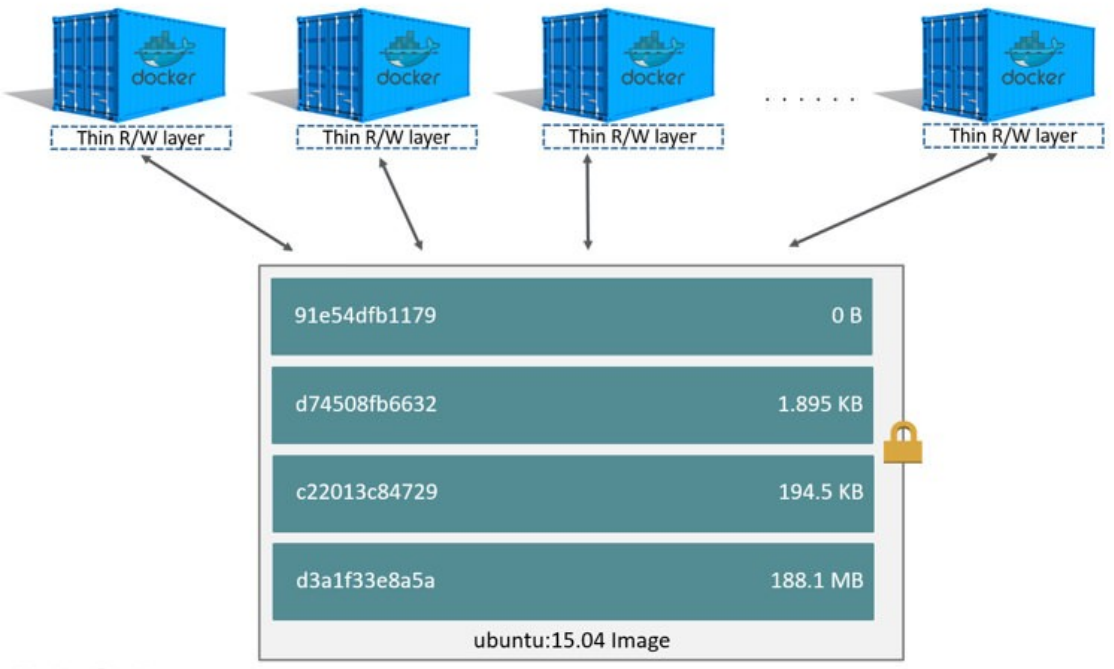
Docker – Go deeper

Arnaud Tillieux

Docker Rappels



Docker Image

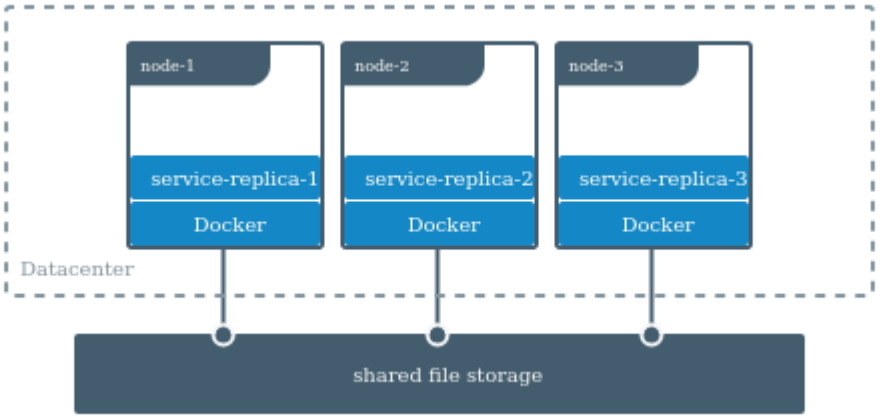
- Une image est « fixe »
- Un conteneur doit être stateless et facilement remplaçable
  - Pas de données générée par l’application dans le conteneur
- Comment faire des « backups » et migrer un conteneur existant ?
  - Exemple base de donnée : Comment la migrer ?
- => Utilisation de volumes (bind mounts)
  - Attention il existe différents « types » volumes !

Docker Volumes

- Permet de « monter » un endroit du système de fichier host vers le conteneur
  - Docker container run -v HostMountPoint:ContainerMountPoint
  - Souvent utilisé dans des env de développement
- Permet de créer un espace persistant entre les recréations de conteneurs
  - Docker container run -v name:ContainerMountPoint …
  - Docker container rm .. -v (suppression volume et container)

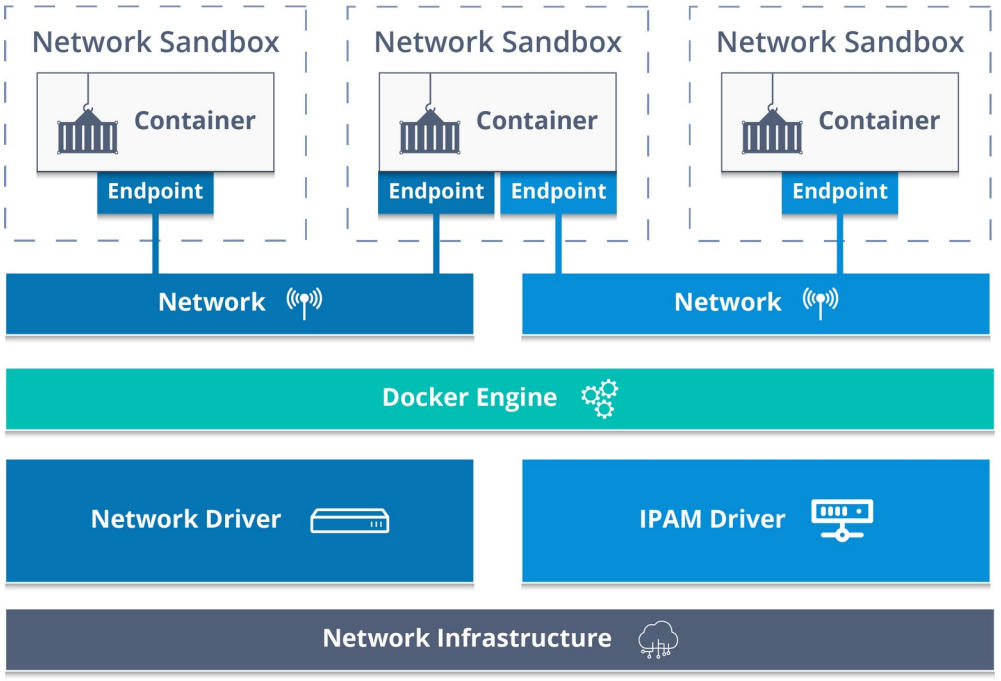
Docker Volumes

- Partage du même volume entre conteneurs



Docker Networks

- Permet de faire communiquer plusieurs conteneurs entre eux



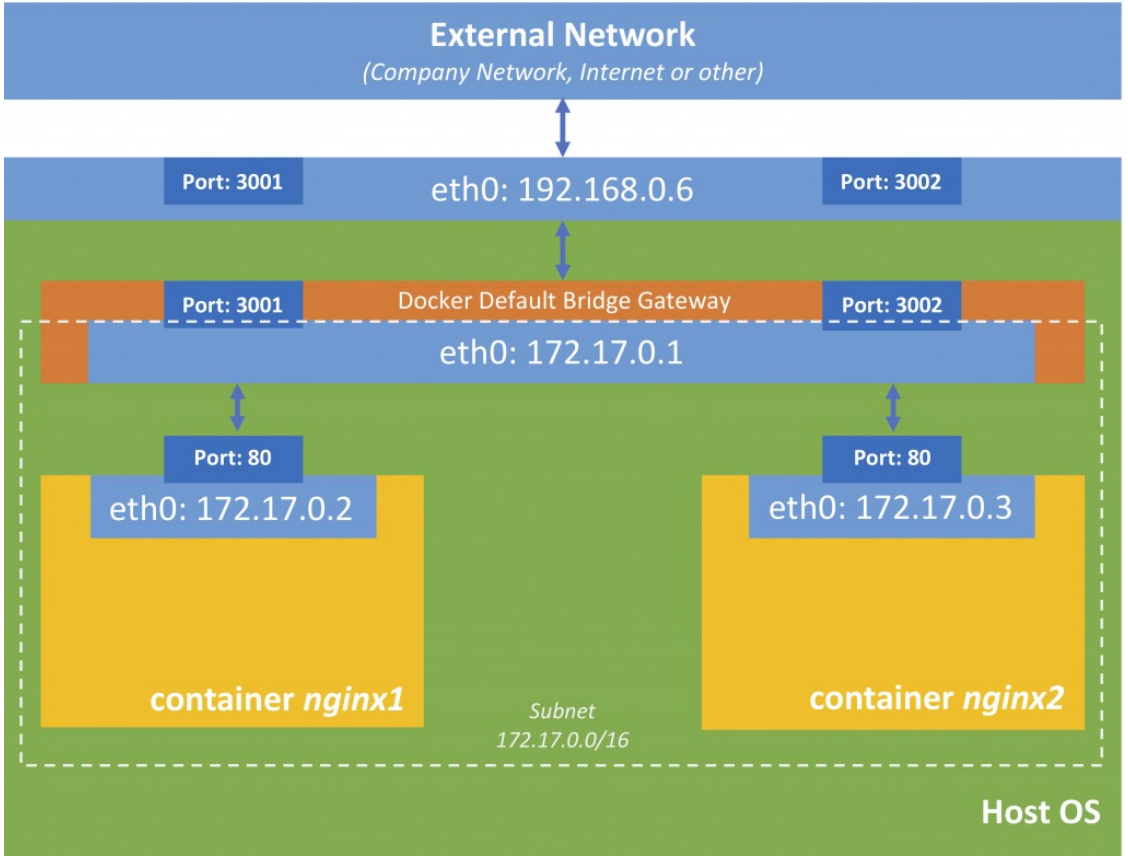
Docker Networks

- Il existe différents « types » de networks
  - Bridge (network par défaut)
  - Host (Directement sur la machine, pas d’isolation)
  - Overlay (Utile pour les swarms)
  - Macvlan (Assigner une MAC à Docker)
  - None (Désactive les networks)
  - Via plugins (Pour avoir d’autres options)

Docker Networks

- docker container run -d --name nginx1 -p3001:80 nginx:alpine
- docker container run -d --name nginx2 -p3002:80 nginx:alpine
- docker network inspect bridge

Docker Networks



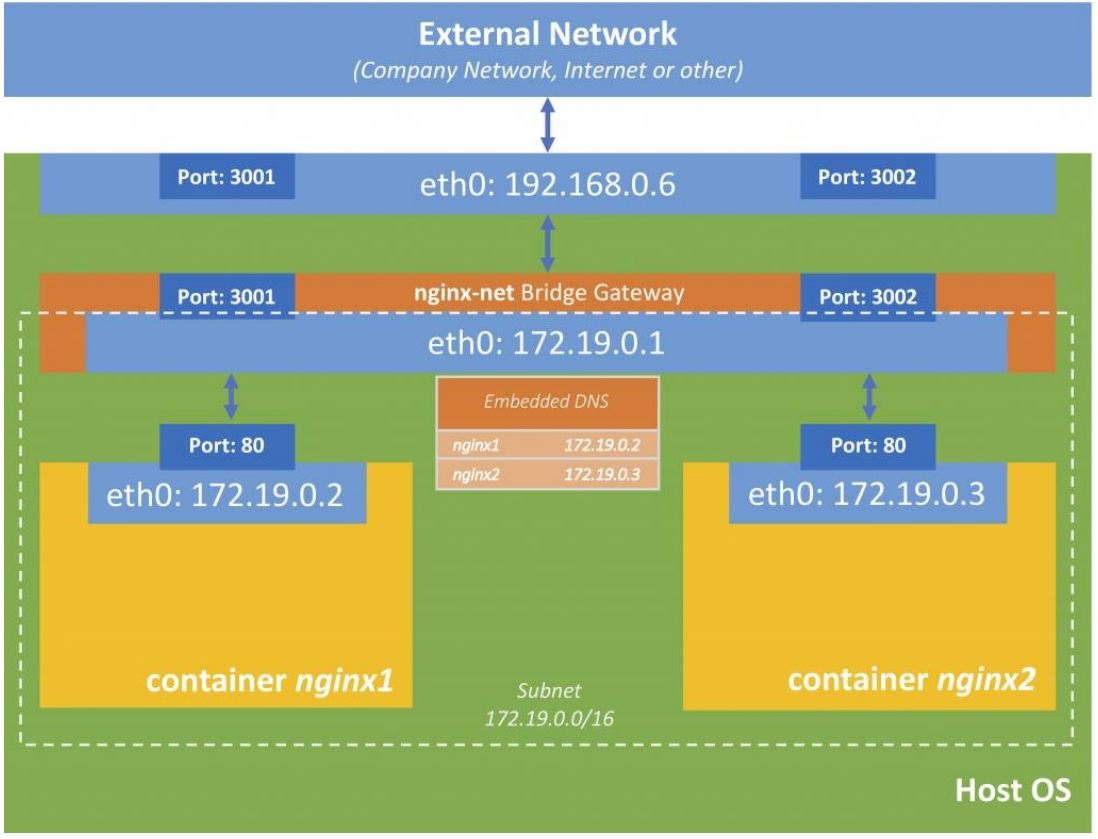
Docker Networks

- Avec le network par défaut impossible de ping avec le nom du conteneur
  - docker container exec -it nginx1 ping nginx2
- Pas pratique dans un projet pour connaître l'adresse de l'autre serveur…
- => Création d'un nouveau network
  - Permet « d'isoler » nos services
  - Permet de contacter les services via un domaine (DNS intégré au network)

Docker Networks

- Création d’un nouveau network type « bridge »
  - docker network create --driver bridge nginx-net
- Lister les networks existants
  - Docker network ls
- Connecter les conteneurs existants au network
  - docker network connect nginx-net nginx1
  - docker network connect nginx-net nginx2
- Inspecter le network créé
  - docker network inspect nginx-net

Docker Network



Docker Networks

- Créer un conteneur et directement le lier à un network existant
  - docker container run -d –network=nginx-net --name nginx1 -p3001:80 nginx:alpine

Docker - Docker-Compose

- Permet d’organiser la création de conteneurs dans un ficher yml
- Lien pour le télécharger : https://docs.docker.com/compose/install/
- TRÈS UTILE pour lancer une architecture type « microservice »
- Le YML contient toute la configuration des conteneurs
  - Crée un network type « bridge » par défaut

Docker - Docker-Compose

- Exemple fichier docker-compose.yml

Version du docker-compose

Liste des services à implémenter

Service Web

Path pour build l’image

Ports à exposer

Service redis

Image du conteneur

```yaml
version: "3.8"
services:
  web:
    build: .
    ports:
      - "5000:5000"
  redis:
    image: "redis:alpine"
```

Docker – Docker-Compose

- Ces deux services seront créés dans un network à part
  - Possibilité de contacter les services via le DNS interne
  - Nom DNS = nom du service
- Démarrage des conteneurs
  - docker-compose up -d (fond de tâche)
- Arrêt et suppression des conteneurs
  - docker-compose down