INSTITUT DE FORMATION AUX NORMES ET TECHNOLOGIES DE L'INFORMATIQUE
SOKODE

Docker

UE Libre

KONDI Abdoul malik
ADJANAYO Simone [1]

IFNTI

November 5, 2024

[1] Inspiré de la documentation docker et du cours d'openclassroom sur docker

Table des matières

1. Concept général
2. Docker préambule
3. Docker
4. Les Conteneurs
5. Les Volumes
6. Les Images
   - Le Dockerfile
   - Manipuler les images
   - Résumé
7. Les Couches (Layers)
8. Le cache
9. Les Réseaux
10. Docker Compose
11. Docker API
12. Environnement développeur

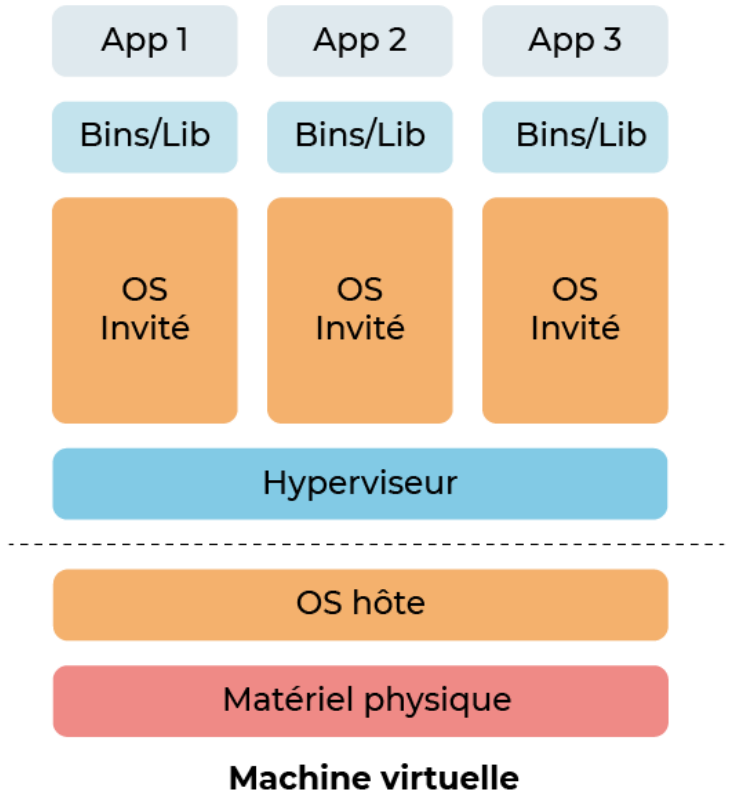
Concept général

Quelques Notions

Notions

- Machine virtuelle
- Conteneur

Machine virtuelle 1/2



Machine virtuelle

Machine virtuelle 2/2 : Une virtualisation lourd

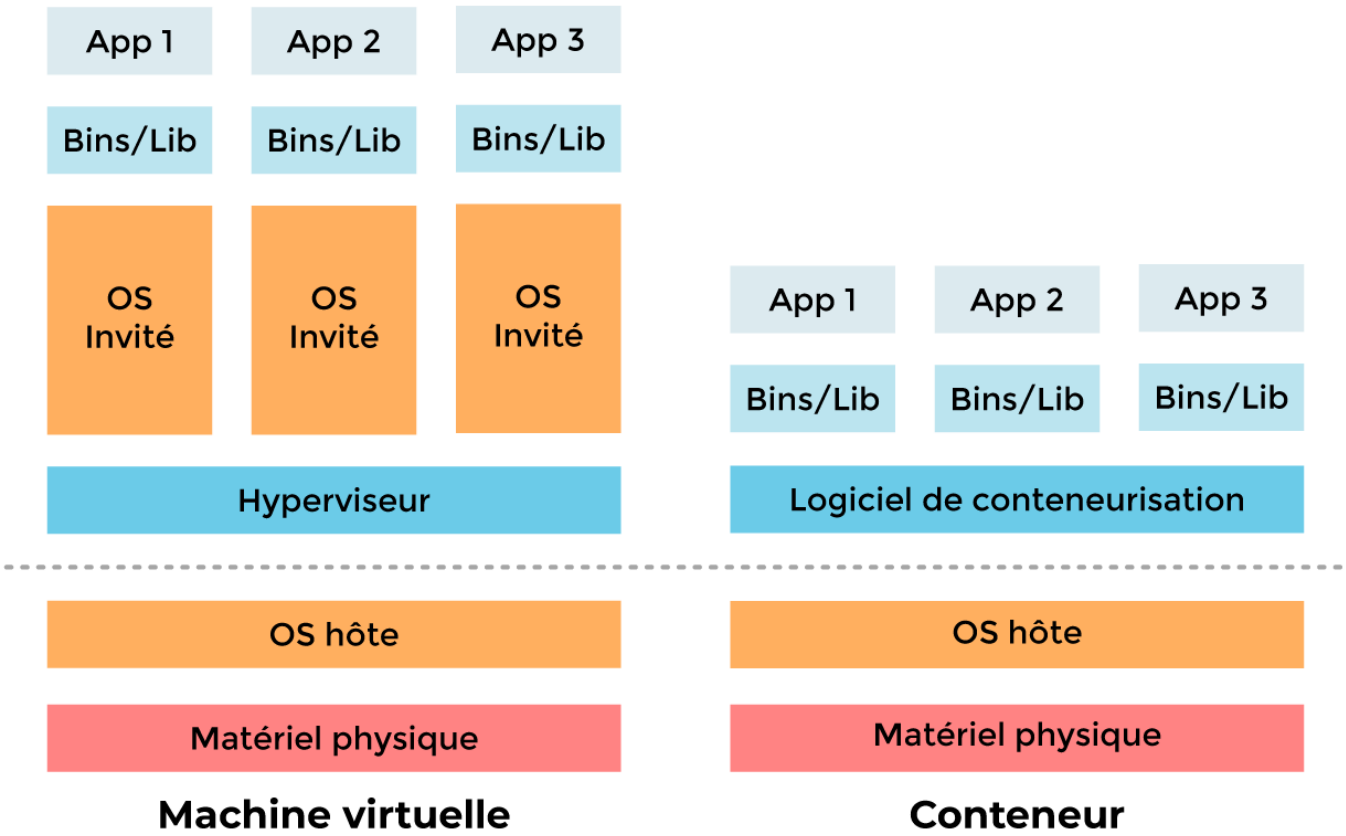
Avantages

- Isolation total
- Réservation des ressources
- Support multiple OS

Inconvénients

- Démarrage lent
- Réservation des ressources

Conteneur 1/3



Conteneur 2/3 : Une virtualisation légère

Avantages

- Ne réservez que les ressources nécessaires
- Démarre plus rapidement
- Donne plus d'autonomie aux développeurs

Inconvénients

- Pas d'isolation total

Conteneur 3/3

Attention

Les conteneurs existait bien avant docker. On peut citez OpenVZ et LXC.

Docker vs LXC ou OpenVZ

- Docker : Logiciel de conteneurisation
- LXC ou OpenVZ : Solutions de virtualisation d'OS

Pourquoi utiliser des conteneurs ?

- Réduction des coûts
- Augmentation de la densité de l'infrastructure
- Scalabilité de l'infrastructure

Un nouveau métier 1/2

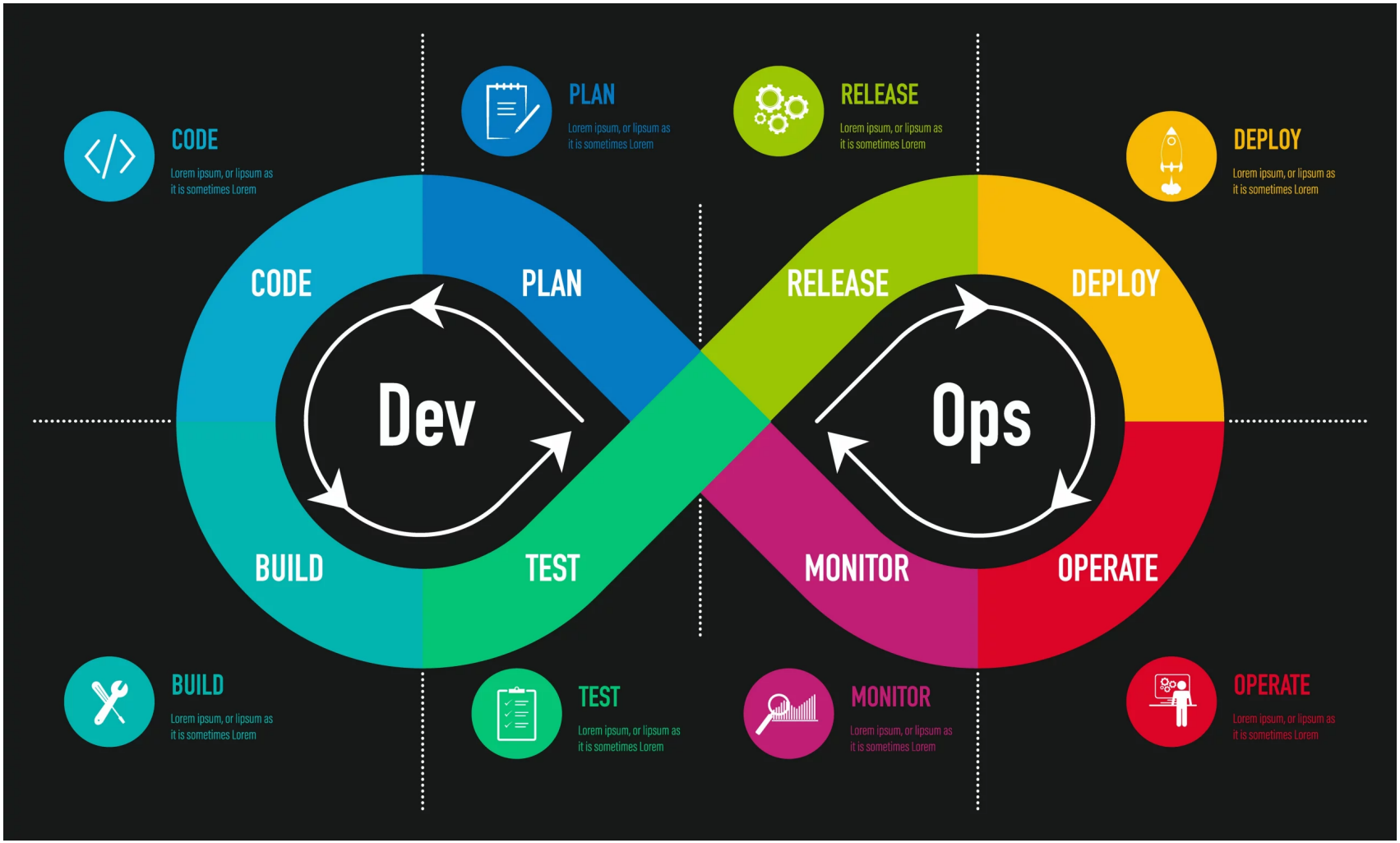
Au paravent

- Les administrateurs système (SysAdmin)
- Les développeurs (Dev)

Exemple d'équation entre ces deux acteurs

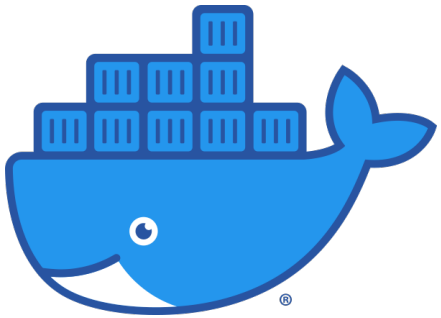
Administrateur système = Garant de la stabilité de la sécurité des systèmes informatique
Développeurs = Créateurs de nouvelles fonctionnalité et applications

Un nouveau métier 2/2



Docker préambule

Qu'est ce que docker ?



Docker

- Une plateforme de conteneurisation.
- Créer en 2013 par la société dotCloud.

Objectifs

Ojectif

Docker permet de faciliter principalement le déploiement des applications web.

Pourquoi docker ?

Raisons

- Évité les difficultés lier au déploiement.
- Avoir un environnement unifié et fonctionnel.

Comprendre la terminologie de Docker ? 1/2

Avant de se lancer dans l'utilisation de Docker, il est important de comprendre la terminologie suivante :

Quelques termes

- Image : programme permettant de créer un conteneur.
- Conteneur : instance d'une image.
- Dockerfile : Fichier de configuration permettant de créer une image.
- Docker compose : Outils qui permet de gérer un ensemble de conteneurs.
- Registry : C'est un dépôt d'image. La registry officielle de docker s'appelle docker hub.
- Stack : C'est un ensemble de conteneur.

Docker

Installer docker 1/2

Trois versions proposer par docker Inc

- Docker Community Edition (Linux seulement)
- Docker Desktop (Mac ou Windows)
- Docker Enterprise (Linux seulement)

Installer docker 2/2

Commande linux

sudo apt-get install docker-ce docker-ce-cli container.io

Lien windows & mac

https://www.docker.com/products/docker-desktop/

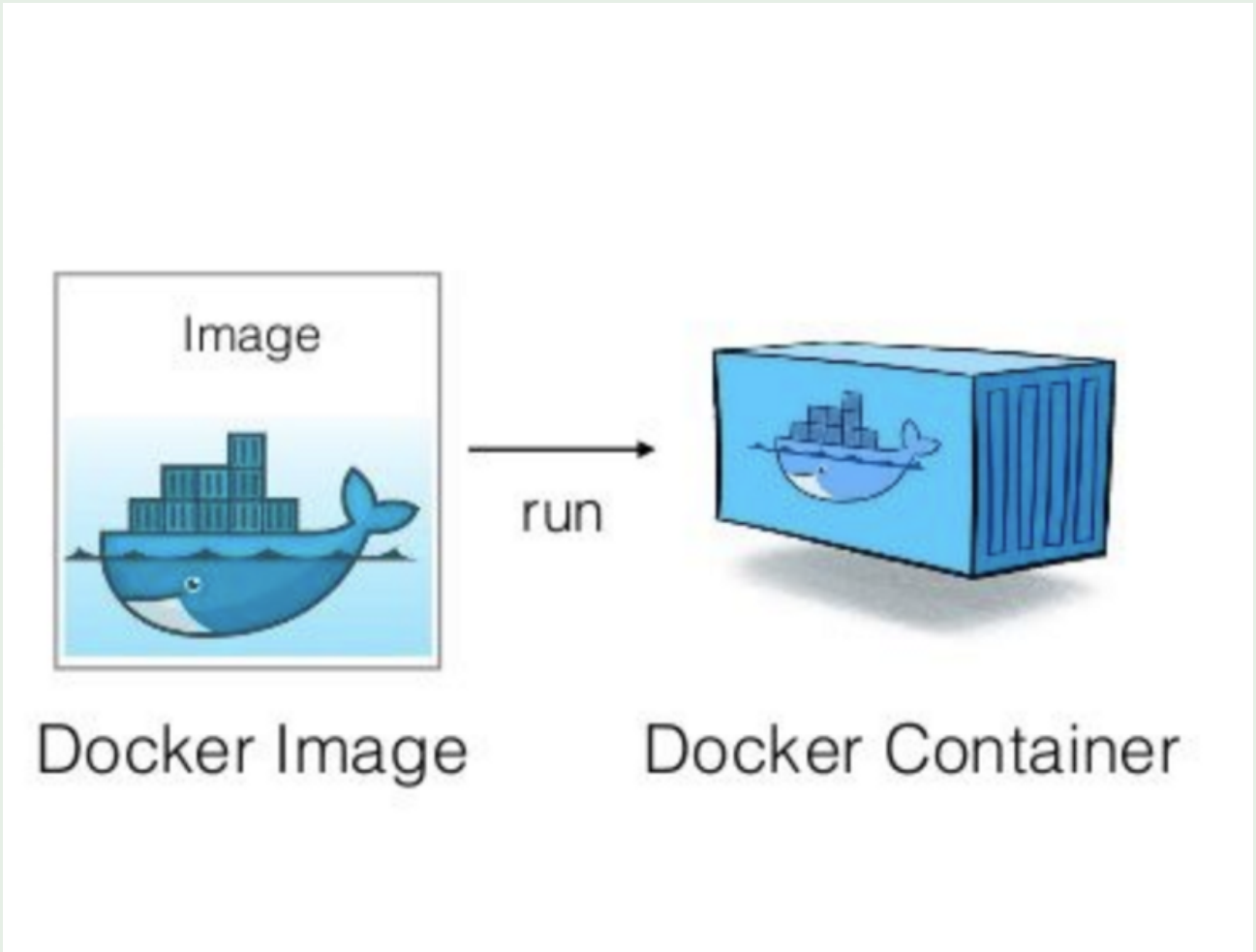
Attentions

Pour ceux qui sont sous windows, il est obligatoire de créer un compte docker Hub. Voici le lien

Les Conteneurs

D'abord ...

C'est quoi un conteneur ?



Utilisons l'image : hello-world

Déduction logique

Exécuter et dite successivement ce que chaque commande fait.

1. docker image ls
2. docker pull hello-world[:latest]
3. docker image ls

NB

Par défaut la version de l'image sera **latest**

Introduction aux images 3/3

Quelques commandes

- Lister toute les images : docker images ou docker image ls
- Telecharger une images (depuis le docker hub) : docker pull image:version
- Supprimer une image : docker rm ID_IMAGE avec l' options -f
- Inspecter une image : docker inspect ID_IMAGE

Types de conteneurs

Deux types

- Conteneur persistant ou avec service
- Conteneur éphémère ou jetable

TP 1 : Manipulez vos premiers conteneurs (1/2)

Créer des conteneurs

1. docker container ls
   - Lancez cette commande et dite ce qu'elle fait ?
2. docker run hello-world
   - Lancez cette commande et dite ce qu'elle fait ?
3. docker run nginx
   - Lancez cette commande et dite ce qu'elle fait ?
4. Quelle remarque fait vous ?
5. docker run -d nginx
   - Lancez cette commande et dite ce qu'elle fait ?

TP 1 : Manipulez vos premiers conteneurs (2/2)

Accéder aux conteneurs

1. De quelle(s) information à t'on besoin pour accéder à un conteneur ?
2. docker inspect ID CONTENEUR
   - Lancez cette commande et dite ce qu'elle fait ?
   - Récupérez les informations demandé précédemment.
3. Accéder au conteneur via votre navigateur web.
4. docker run -d -p 8080:80 nginx
   - Lancez cette commande et allez à l'adresse suivante : localhost:8080.
   - Pouvez vous dire ce que fait cette commande ?
5. docker exec -it ID CONTENEUR bash
   - Lancez cette commande et dite ce qu'elle fait ?

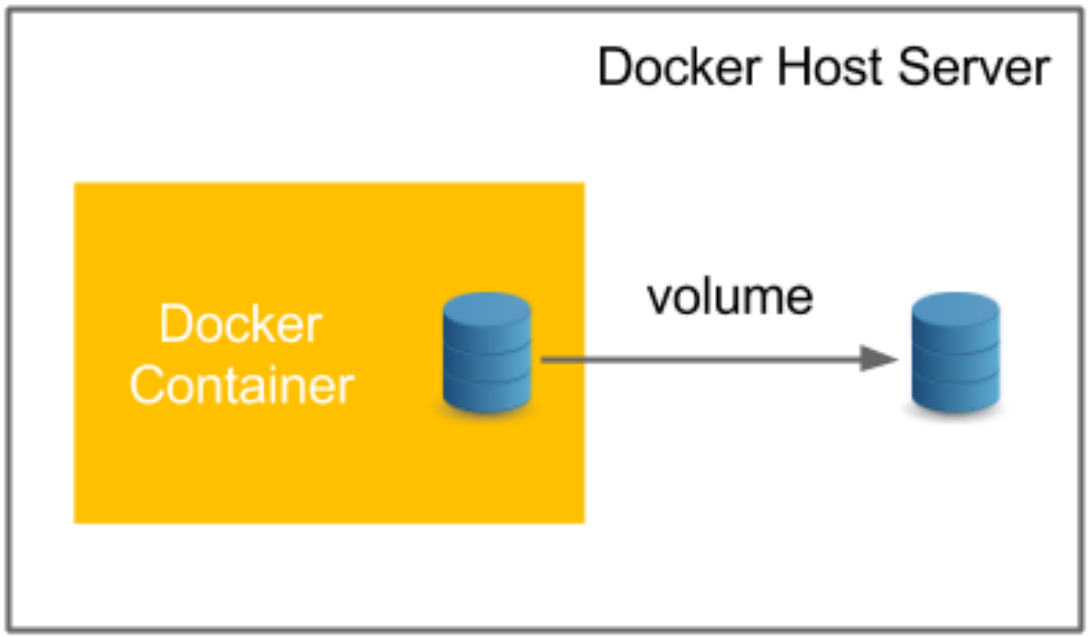
En résumé

Quelques commandes

- docker ps ou docker container ls
- docker run image:version avec les options -d et -p
- docker stop ID_CONTENEUR
- docker start ID_CONTENEUR
- docker exec -it ID_CONTENEUR
- docker rm ID_CONTENEUR avec l' options -f
- docker inspect
- docker system prune

Les Volumes

Introduction



Utilité

- Rendre un conteneur state full

Type de volume

Deux types

- Les volumes persistant
- Docker Volume

TP 2 partie 1 : Les volumes persistant

TP 2 partie 2 : Docker Volume

En résumé

Quelques commandes

- docker volumes
- docker run -d –name [nom_serveur] -v [chemin_volume_hôte]:[chemin_volume_conteneur] [nom_conteneur]
- docker run -d –name [nom_serveur] -v [nom_volume]:[chemin_volume_conteneur] [nom_conteneur]
- docker run -d –name [nom_conteneur] –mount source=[nom_volume] ,target=[chemin_volume_conteneur] nginx

Les Images

Le Dockerfile

C'est quoi ?

Un Dockerfile est un fichier de configuration.

Attention

Chaque ligne du docker file créer une layer (couche).

Quelques closes d'un Dockerfile

- RUN
- ENV
- EXPOSE
- VOLUME
- COPY
- ENTRYPOINT

Les actions possible sur une image.

Quelques commandes

- Tag
- Pull
- Push

TP 3

TP 3 partie 1

Création d'un fichier dockerfile

TP 3 partie 2

Tag, pull et push

En résumé

Quelques commandes

- docker build -t [nom_image]:[tag] .
- docker build -t [nom_image][:[tag]] -f nom_dockerfile] .
- docker tag [nom_image]:[tag_actuel] [nom_image]:[nouveau_tag]
- docker pull [nom_image]:[tag]
- docker push [nom_image]:[tag]

Les Couches (Layers)

Types & propriétés

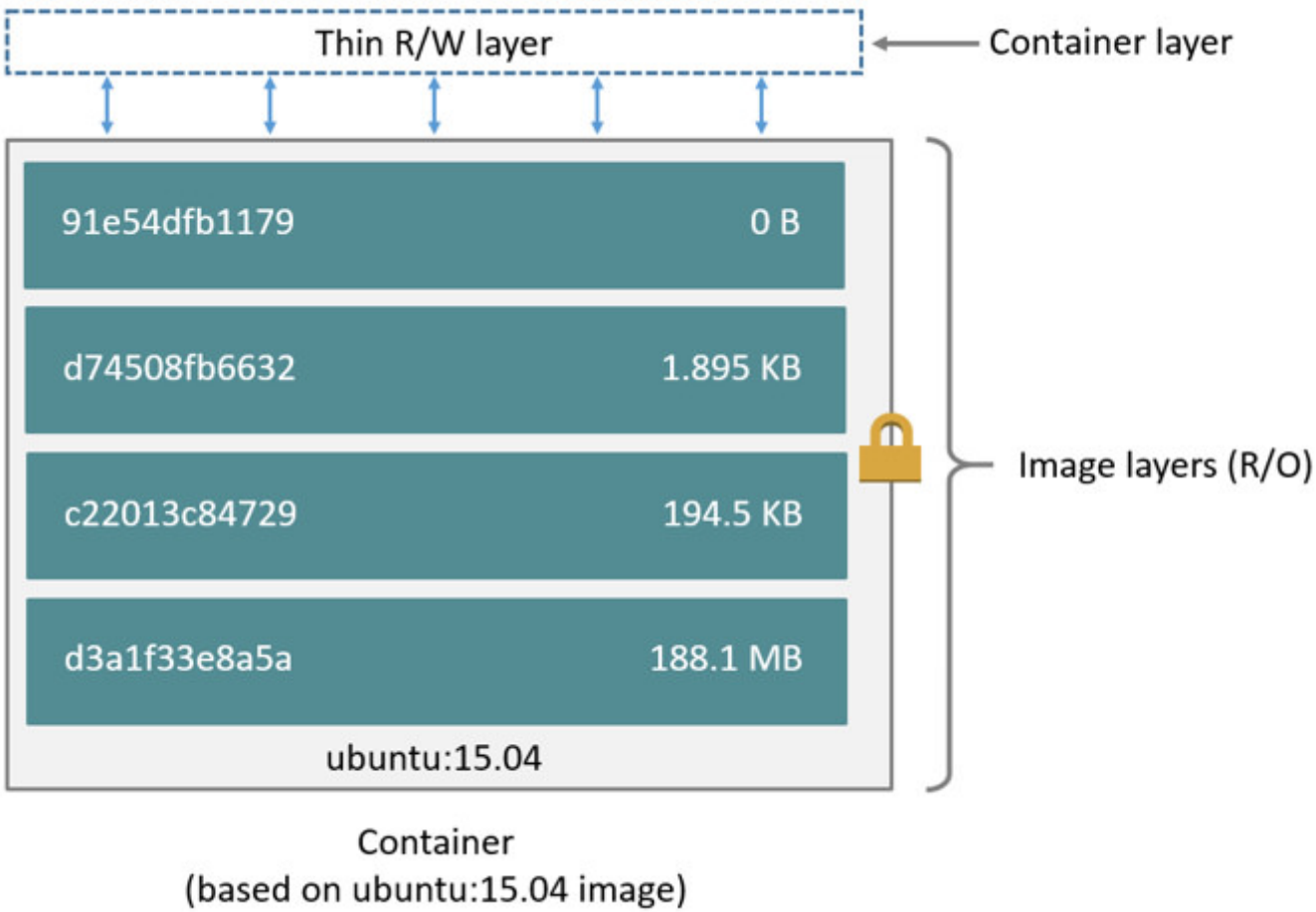
Les types de couches

- Lecture
- Lecture & écriture

Les propriétés

- partage de couche.

Principe du Copy & Write



TPs

Deux TPs

- TP 1 lecture & écriture
- TP 2 le copie and write

Le cache

Objectif

But :

- construire plus vite les images
- démarrer plus vite les conteneurs
- stocker des images légères
- partager des couches/layers

TP

Les Réseaux

Les types de réseaux

Les voici

- Bridge
- Host
- none

Le réseaux par défaut

Nom : Bridge ou Docker 0

Adresse IP : 172.17.0.1/16

Comment définir le réseau d'un conteneur ?

Il existe deux moyens

- - -network
- - -link

Faisons du réseau avec Docker

TP 1

Network

Docker Compose

C'est quoi une stack docker-compose ?

Définition

- Gestionnaire de conteneurs

Exemple

Comment déployer un environnement Wordpress ?

Notion de service

Définition

- 1 service = 1 processus = 1 conteneur

Solution

- Identifier les différents services.

DD

Docker compose : installation

Linux

sudo apt-get install docker compose –version

Windows

sudo apt-get install

Mac

sudo apt-get

TP : Utilisons docker compose

Le CLI Docker Compose

Quelques commandes

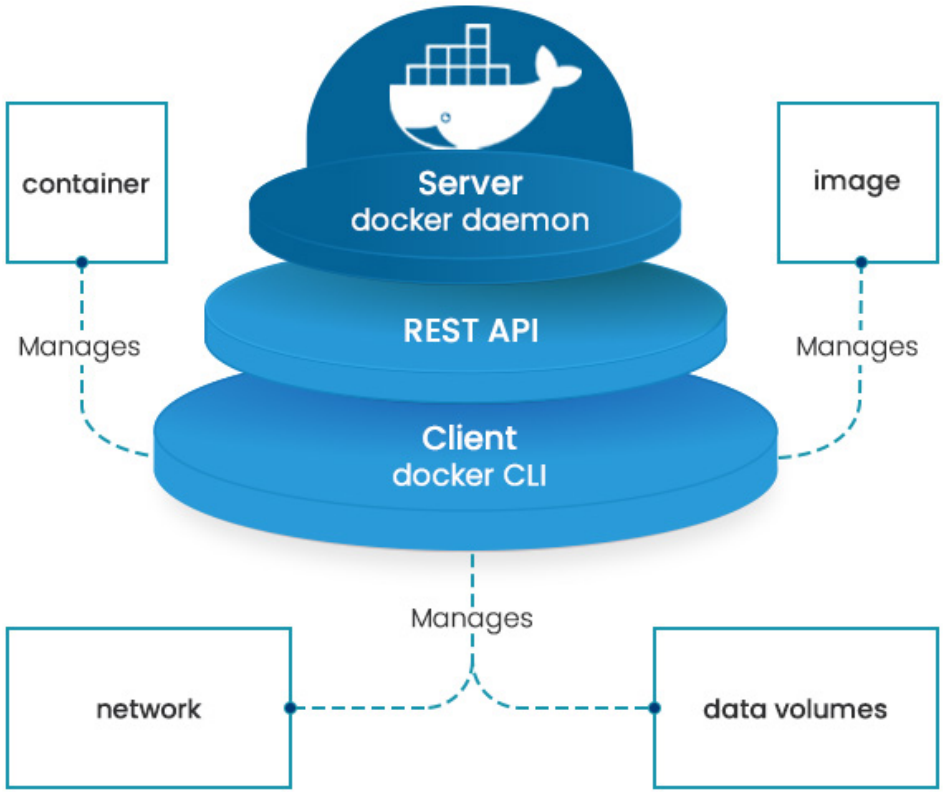
- docker-compose config
- docker-compose up (options: **-d** ...)
- docker-compose ps
- docker-compose logs (options: **-f, -tail**...)
- docker-compose stop et docker-compose start.
- docker-compose down

Attention

Pour utiliser le CLI Docker compose nous devons avoir un fichier **docker-compose.yml**.

Docker API

Notion de socket



socket TCP/IP

socket SSH

TP

Environnement développeur