PRÉSENTATION DOCKER

CATI DIISCICO

Julien Cufi

04/02/2020

PRÉSENTATION DOCKER - CATI DIISCICO

1. Docker introduction
2. Docker pour l'utilisateur
3. Docker pour le développeur
4. Retours d'expérience

DOCKER INTRODUCTION

Qu'est-ce que c'est ?

Docker est une technologie permettant d'exécuter une application dans un environnement isolé, comprenant

l'application mais également l'ensemble des dépendances nécessaires a son fonctionnement.

DOCKER INTRODUCTION

Quelques mots de vocabulaire...

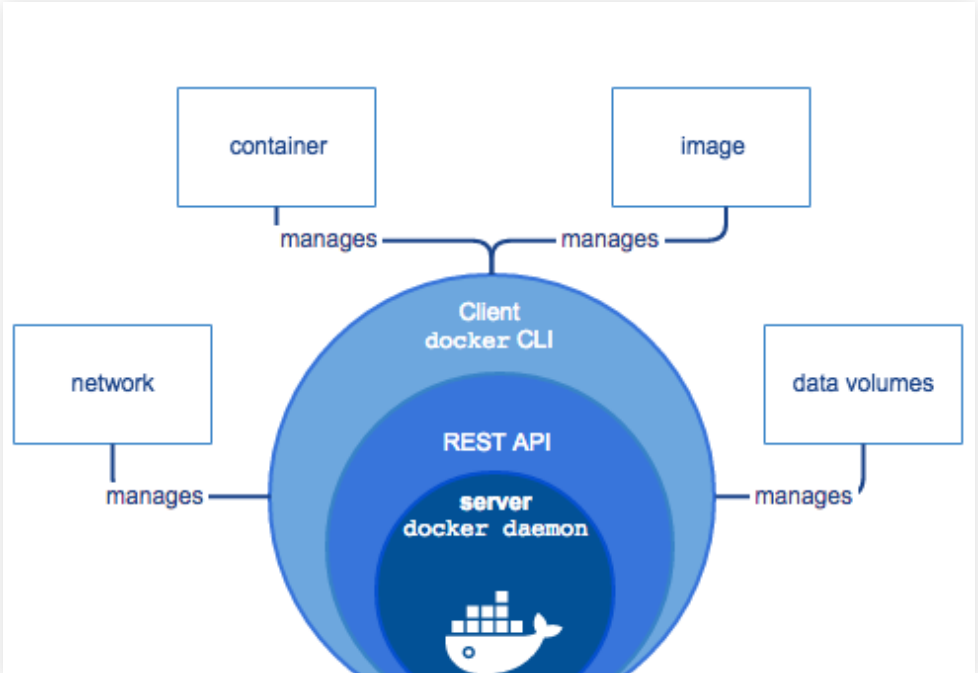
- L'environnement dans lequel s'exécute l'application est appelé un ***conteneur Docker***.
- La matrice servant a définir ce qui est présent dans le conteneur est appelé une ***image Docker***.
- Un catalogue public d'images accessible sur le web

DOCKER INTRODUCTION

Qui l'a crée et pourquoi ?

- Docker est un projet OpenSource sous licence Apache 2.0 crée en 2013 par Solomon Hykes.
- Le projet est supporté par la communauté et par l'entreprise Docker Inc.
- Crée a l'origine pour la société dotCloud (PaaS)

DOCKER INTRODUCTION



DOCKER INTRODUCTION

Comment ça marche ?

A l'origine Docker s'appuie sur LXC (LinuX container) et les fonctionnalités d'isolation du noyau Linux (cgroup,

namespaces, ...) pour fournir un environnement d'exécution étanche.

⇒ Problème de portabilité pour Docker Inc.

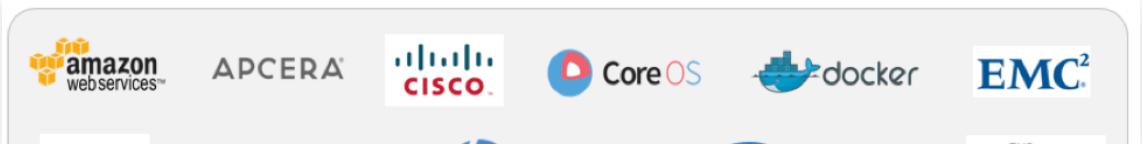
DOCKER INTRODUCTION

- Remplacement de LXC par libcontainer
- Modularisation de la partie serveur
  - création d'outils spécifiques (creation container, gestion du cycle de vie...)
  - implémentation de référence des spécifications émise par Open Container Initiative...

DOCKER INTRODUCTION

Open Container Initiative

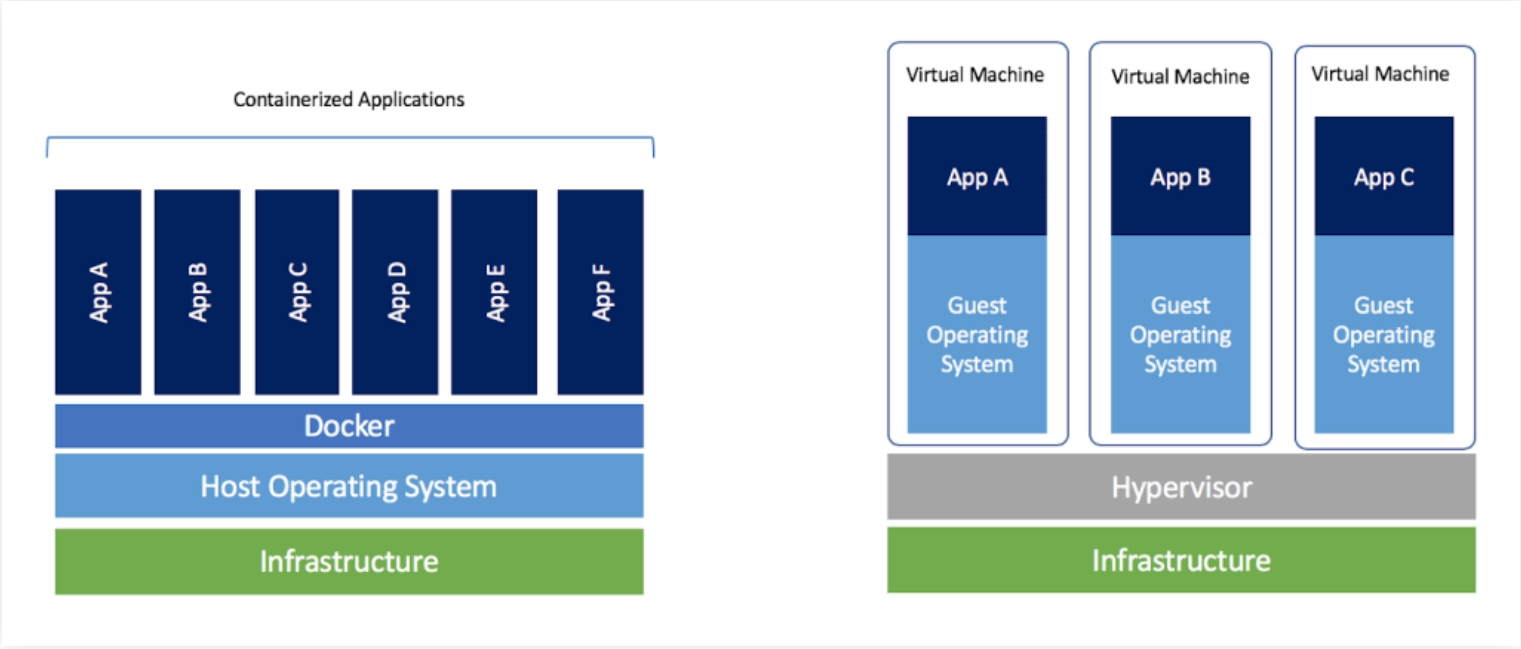
- Projet crée en 2015 et supporté par la Fondation Linux
- Objectif : la normalisation des conteneurs
  - environnement d'exécution des conteneurs
  - images
- Crée par Docker Inc. avec quelques petites sociétés...



DOCKER INTRODUCTION

Je fais déjà ça avec mes machines virtuelles !

Dans une machine virtuelle, on simule une machine (ie. toute la partie hardware) et chaque machine virtuelle a son propre système d'exploitation.



DOCKER INTRODUCTION

Comment je l'utilise ?

- Docker est disponible sur Linux / Windows / Mac (VM)
  ⇒ requiert Hyperviseur sous windows
- Deux versions EE et CE (≠ niveaux de support)
- On l'utilisera au travers de lignes de commandes

passons à la pratique...

DOCKER : 1ER CAS D'UTILISATION

Quelques commandes de base

```
# Télécharger une image
$docker image pull <image>

# Démarrer un conteneur
$docker container run <image>

# Lister les conteneurs démarrés
$docker container ps

# Stopper un conteneur
$docker container stop <nom conteneur / identifiant>

# Supprimer un conteneur
$docker container rm <nom conteneur / identifiant>
```

DOCKER : 1ER CAS D'UTILISATION

Je souhaite démarrer une base postgres v12 pour effectuer quelques tests.

- Recherche d'une image existante sur DockerHub (https://hub.docker.com/)
- Le DockerHub contient
  - des images officielles : vérifiées par Docker et à jour
  - des images non-officielles : le Far West

DOCKER : 1ER CAS D'UTILISATION

Démarrage d'un conteneur basé sur l'image postgres:12

```
$docker container run -it postgres:12
Unable to find image 'postgres:12' locally
12: Pulling from library/postgres
8ec398bc0356: Downloading [====>]  11.72MB/27.09MB
65a7b8e7c8f7: Download complete
b7a5676ed96c: Download complete
```

Le client demande le démarrage d'un conteneur, le serveur ne connaissant pas l'image il interroge le DockerHub et la télécharge.

DOCKER : 1ER CAS D'UTILISATION

Une fois l'image téléchargé le conteneur est démarré, la base est prête a être utilisée

```
PostgreSQL init process complete; ready for start up.
listening on IPv4 address "0.0.0.0", port 5432
database system is ready to accept connections
```

⇒ Youpi ?

DOCKER : 1ER CAS D'UTILISATION

Le conteneur est un environnement isolé de l'hôte donc *par défaut* :

- Pas de communication réseau
- Pas de partage de données

⇒ Association de port hôte/conteneur

⇒ Montage d'un volume partagé

DOCKER INTRODUCTION

On recommence

```
# Création d'un volume
$docker volume create pgdata
pgdata
# Lancement du conteneur
$docker container run -it
                      -p 5432:5432
                      -v pgdata:/var/lib/postgresql/data postgres:12
```

-p 5432:5432

⇒ Association de port hôte/conteneur

-v pgdata:/var/lib/postgresql/data postgres:12

⇒ Montage d'un volume partagé

DOCKER

Pour les curieux

```
$docker volume inspect pgdata
[
    {
        "CreatedAt": "2019-02-26T17:12:59+01:00",
        "Driver": "local",
        "Labels": {},
        "Mountpoint": "/var/lib/docker/volumes/pgdata/_data",
        "Name": "pgdata",
        "Options": {},
        "Scope": "local"
    }
]
$ls /var/lib/docker/volumes/pgdata/_data
postgresql.conf base     pg_commit_ts  pg_ident.conf  pg_notify
...
```

DOCKER

Quelques particularités sur les conteneurs

- Ephémères
- Isolés de l'hôte
- Tant qu'ils ne sont pas supprimés les fichiers présents dans le conteneur existent

DOCKER : 2EME CAS D'UTILISATION

J'ai implémenté un algorithme, je souhaite le mettre a disposition

Objectif :

Faciliter la reproductibilité des résultats en minimisant les étapes d'installation* du logiciel

⇒ Nécessite de créer une image Docker propre à son logiciel

* mais il y en aura toujours

DOCKER

Focus sur une image Docker

- Une image docker est un fichier texte nommé Dockerfile respectant un language propre a Docker
- On peut "hériter" d'autres images existantes pour les étendre
- Il contient l'ensemble des instructions nécessaires a l'installation du logiciel
- Il est nécessaire de compiler le fichier Dockerfile pour

DOCKER

Exemple de fichier Dockerfile :

```
FROM ubuntu:latest
RUN apt-get update && \
    apt-get install -y cowsay
ENTRYPOINT ["/usr/games/cowsay"]
```

Compilation de l'image :

```
$docker image build -t cow .
Step 1/3 : FROM ubuntu:latest
 ---> dd6f76d9cc90
Step 2/3 : RUN apt-get update &&     apt-get install -y cowsay
 ---> Running in 9a0c163a5579
...
Step 3/3 : ENTRYPOINT ["/usr/games/cowsay"]
 ---> Running in c17aa839e8a8
Removing intermediate container c17aa839e8a8
 ---> 000e5e657c8d
Successfully built 000e5e657c8d
Successfully tagged cow:latest
```

DOCKER

Utilisation de l'image

```
$docker container run cow "Je suis une vache"

 ____________________
< Je suis une vache >
 --------------------
        \   ^__^
         \  (oo)\_______
            (__)\       )\/\
                ||----w |
                ||     ||
```

DOCKER

Retour sur l'exemple initial

```
# Le fichier Dockerfile
FROM maven:3.6.2-jdk-8
RUN mkdir -p /app/results && \
    mkdir -p /app/src && \
    mkdir -p /app/data
COPY data /app/data
COPY src /app/src/

COPY pom.xml build.xml /app/
RUN gzip -d /app/data/FoodOnAgroPortalImport2.nq.gz && \
    chmod -R 755 /app
WORKDIR /app
CMD ["mvn", "package", "exec:java", ..."]
VOLUME [ "/app/results"]
```

```
# Installation du logiciel
$docker image build -t align-tool .
# Lancement du logiciel
$docker container run --rm -it -v ${pwd}/results:/app/results align-t
```

DOCKER

DOCKER : RETOURS D'EXPÉRIENCES

Notre besoin

- Mise en place de deux plateforme (test et production)
- Automatiser l'installation des logiciel i. 8 applications web ii. Schémas d'efactor, d'atweb, de spo2q, de meatylab, capex-ee, mychoice, de fuseki, de damn = 4 BD sem (2 graphdb, 1 fuseki), 7 serveurs d'app (5 JAVA, 1 truc de capex, 1 ruby), 2 serveur R, 3 bases relationnelles (2 postgres, 1 mysql), 2 bases NoSQL = 9

DOCKER

- Complexité
  - Unix : lu, parlé, écrit
  - Problèmes liés au LOCALE
  - Impossibilité de charger des modules dans le kernel (modprobe)

DOCKER

- Bonnes pratiques (commune)
  - Bon sens : ne pas récupérer nimp sur le DockerHub
    - Etude sur les failles dans les images Docker
  - Ne pas monter la racine / dans le conteneur (\U+1F628)
  - Utilisateur dédié

DOCKER

- Bonnes pratiques (commune a l'admin sys)
  - Ne pas surcharger le conteneur avec des paquets inutiles
  - Logiciel a maintenir à jour
- Sécurité

DOCKER

LIENS

- Lien vers projet Docker OpenSource
  https://github.com/moby
- Docker security bench
  https://github.com/docker/docker-bench-security