



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

Docker

UE Libre

KONDI Abdoul malik
ADJANAYO Simone ¹

IFNTI L3

May 29, 2023

¹Inspiré 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 Réseaux
- 8 Docker Compose

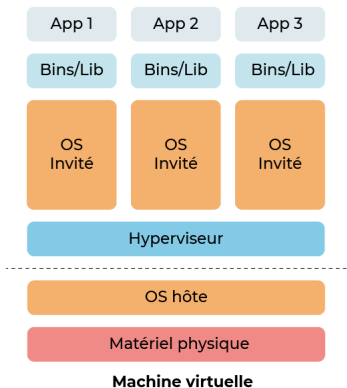
Concept général

Quelques Notions

Notions

- Machine virtuelle
- Conteneur

Machine virtuelle 1/2



Machine virtuelle 2/2 : Une virtualisation lourde

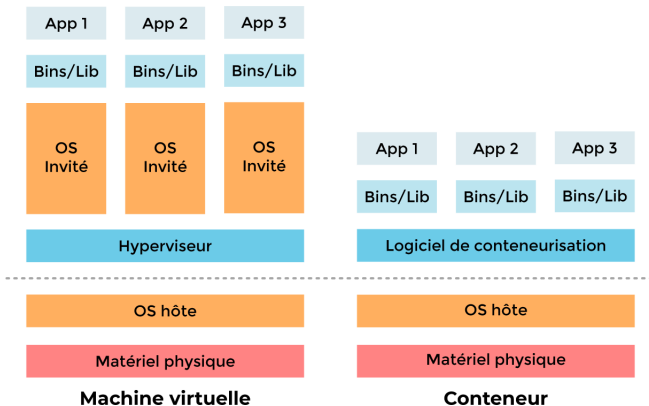
Avantages

- Isolation totale
- Réserveation des ressources
- Support multi OS

Inconvénients

- Démarrage lent
- Réserveation des ressources

Conteneur 1/3



Conteneur 2/3 : Une virtualisation légère

Avantages

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

Inconvénients

- Pas d'isolation totale

Conteneur 3/3

Attention

Les conteneurs existaient bien avant docker. On peut citer 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
- Modularité de l'infrastructure

Un nouveau métier 1/2

Auparavant

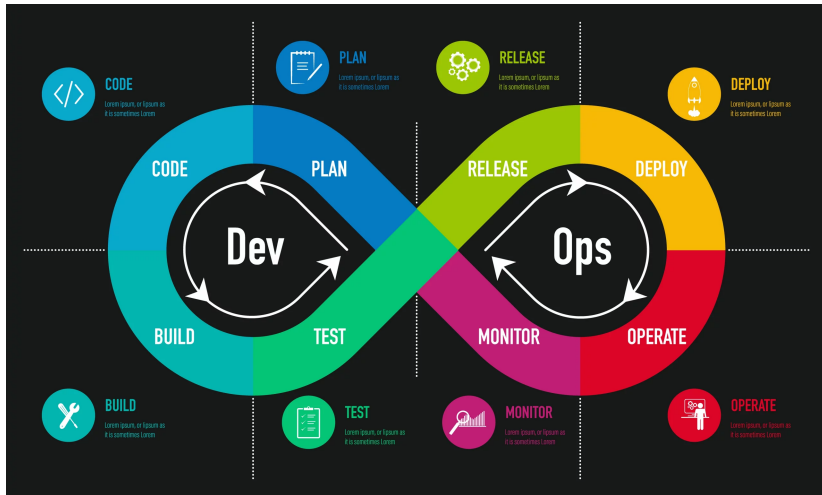
- Les administrateurs système (SysAdmin)
- Les développeurs (Dev)

Exemple d'équation entre ces deux acteurs

Administrateur système = Garant de la stabilité de la sécurité des systèmes informatiques

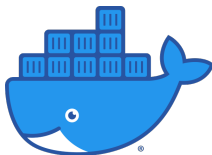
Développeurs = Créateurs de nouvelles fonctionnalités et applications

Un nouveau métier 2/2



Docker préambule

Qu'est ce que docker ?



Docker

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

Objectif

Ojectif

Docker permet de faciliter principalement le déploiement des applications web.

Pourquoi docker ?

Raisons

- Éviter les difficultés liées 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 : Fichier de configuration permettant de créer et de gérer plusieurs conteneurs.
- Registry : C'est un dépôt d'images. La registry officielle de docker s'appelle **docker hub**
- Stack : C'est un ensemble de conteneurs.

Docker

Installer docker 1/2

Trois versions proposées par docker Inc

- Docker Community Edition (Linux seulement)
- Docker Desktop (Mac ou Windows)
- Docker Enterprise (Linux seulement et il est propriétaire)

Installer docker 2/2

Lien des commandes linux

Lien des commandes windows & mac

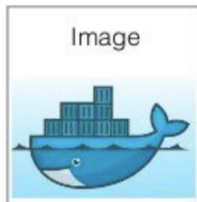
Attention

Pour ceux qui sont sous windows, il est obligatoire de créer un compte docker Hub. Voici le lien <https://hub.docker.com/>

Les Conteneurs

D'abord ...

Qu'est ce que c'est ?



Docker Image

run



Docker Container

Télécharger une image : hello-world

Que fait chaque commande ?

- docker image ls
- docker pull hello-world[:latest]
- docker image ls

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 dites ce qu'elle faites ?
- 2- docker run hello-world
 - Lancez cette commande et dites ce qu'elle faites ?
- 3- docker run nginx
 - Lancez cette commande et dites ce qu'elle faites ?
- 4- Quelle remarque fait vous ?
- 5- docker run -d nginx
 - Lancez cette commande et dites ce qu'elle faites ?

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

Accéder aux conteneurs

- 1- De quelle(s) information(s) a-t-on besoin pour accéder à un conteneur ?
- 2- `docker inspect ID_CONTENEUR`
 - Lancez cette commande et dites ce qu'elle fait ?
 - Récupérez les informations demandées précédemment.
- 3- Accédez au conteneur via votre navigateur web.
- 5- `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 dites ce qu'elle fait ?

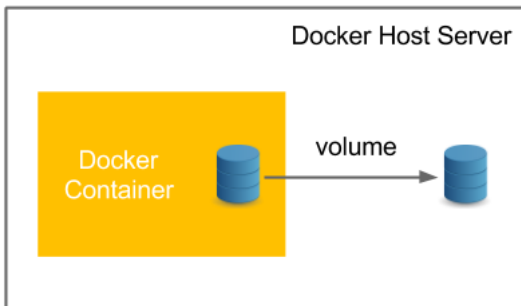
En résumé

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'option `-f`
- `docker inspect`
- `docker system prune`

Les Volumes

Introduction



Utilité

- Rendre un conteneur "state full"

Types de volumes

Deux types

- Les volumes persistants
- Docker Volume

TP 2 Les volumes persistants (1/2)

Utilisez un volume

- 1- `docker run -d --name serveur_web nginx`
 - Lancez cette commande et dites ce qu'elle fait.
 - Affichez la liste de tous les conteneurs et accédez au conteneur via le navigateur.
- 2- Quel est le chemin du fichier `index.html` du serveur `nginx` ?
Modifiez le fichier `index.html` et réactualisez la page.
- 3- Supprimez le conteneur
- 4- Recréez le conteneur `serveur_web`. vérifiez via le navigateur si le contenu de votre fichier *index.html* a toujours les modifications appliquées. Est ce le cas ?
- 5- Créez un répertoire `html`. Créez un fichier `index.html` mettez à l'intérieur ce que vous voulez.

TP 2 Les volumes persistants (2/2)

Utilisez un volume

- 6- `docker run -d --name serveur_web -v $(pwd)/html:/usr/share/nginx/html nginx`
- Visitez le contenu du fichier index.html via le navigateur.
 - Modifiez le contenu du fichier index.html et revisitez la page html.
 - Que fait l'option **-v** ?

TP 2 Docker Volume (1/2)

Utilisez docker volume

1- docker volume ls

- Que fait cette commande ?

2- docker volume create mon_volume.

- Que fait cette commande ?

3- docker inspect mon_volume.

- Que fait cette commande ?
- Dans quel répertoire le volume a-t-il été créé ?

4- docker run -d --name serveur_web --mount source=mon_volume,target=/usr/share/nginx/html nginx

- Que fait cette commande ?
- Modifiez le contenu du fichier index.html et revisitez la page html.

TP 2 Docker Volume (2/2)

Utilisez docker volume

- 5- Supprimez le conteneur
- 6- Recréez le conteneur `serveur_web` (ref N-4). Vérifiez via le navigateur si le contenu de votre fichier `index.html` a toujours les modifications appliquées. Est ce le cas ?

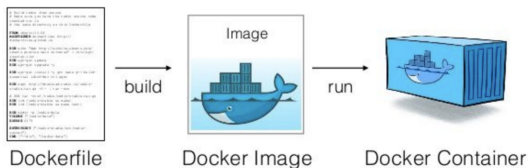
En résumé

Quelques commandes

- `docker volumes`
- `docker run -d --name nom_conteneur -v chemin_volume_hôte:chemin_volume_conteneur nom_image`
- `docker run -d --name nom_conteneur -v nom_volume:chemin_volume_conteneur nom_image`
- `docker run -d --name nom_conteneur --mount source=nom_volume,target=chemin_volume_conteneur nom_image`

Les Images

Le Dockerfile



C'est quoi ?

Un Dockerfile est un fichier de configuration.

Attention

Chaque ligne du dockerfile créer une layer (couche).

Quelques clauses d'un Dockerfile

- FROM *image:version* : Définir l'image de base.
- RUN *commande* : Exécuter une commande système.
- ENV *var=valeur* : Définir une variable d'environnement.
- EXPOSE *port* : Exposer un port d'écoute.
- VOLUME *nom_volume* : Définir un volume.
- COPY *src dest* : Copier des données de l'hôte vers le conteneur.
- CMD [*"commande"*, ...] : Définir le processus maître.

TP 3 : Déployez un site html, css et js



TP 3 : Architecture des fichiers

```
mon_site_web/  
├── gestion_ticket  
│   ├── index.html  
│   ├── index.js  
│   ├── README.md  
│   └── styles.css  
  
1 directory, 4 files
```


TP 3 : Dockerfile

Instructions

- Créez un répertoire nommé mon_site_web.
- Copiez le répertoire gestion_ticket qu'on va vous fournir.
- Créez un dockerfile dans le répertoire nommé mon_site_web.

Voici sont contenu :

```
FROM nginx
```

```
WORKDIR /usr/share/nginx/html/
```

```
COPY gestion_ticket/* .
```

```
ENTRYPOINT ["nginx", "-g", "daemon off;"]
```

Créez une image

Instructions

- Exécutez la commande : `docker image ls`.
- `docker build -t image_gestion_ticket .`
 - Que fait cette commande ? (Exécutez la).
 - Exécutez la commande : `docker image ls`.
- Créez un conteneur nommé `mon_site` à partir de l'image `image_gestion_ticket`.

Les actions possible sur une image.

Quelques commandes

- Tag
- Pull
- Push

En résumé

Quelques commandes

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

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

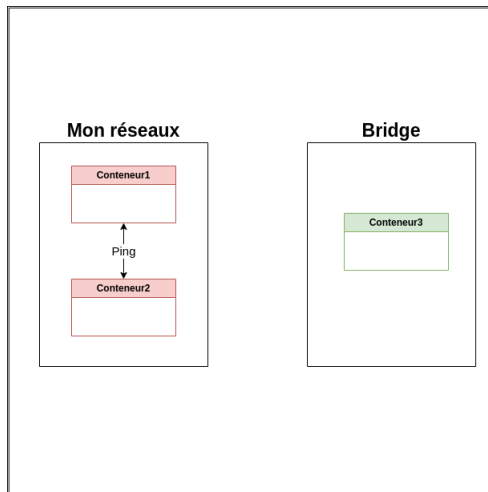
Faisons du réseau avec Docker 1/4

Ping sur le docker 0

Créer le conteneur conteneur1 qui est une instance d'une image de l'**OS** apline.

- Quelle commande avez-vous lancé ?
- Rendez-vous dans le conteneur nommé conteneur1 . Quelle commande avez-vous lancé ?
- Lancez la commande *ip a* . Que fait cette commande ?
- Quelle est le numéro de machine logique, physique de votre conteneur ?
- Que remarquez-vous pour le premier numéro ?

Faisons du réseau avec Docker 2/4



Faisons du réseau avec Docker 3/4

Ping sur le docker 0

Lancez les commandes suivantes puis dites ce quelles font.

- `docker network ls`
- `docker network create -d bridge --subnet 172.30.0.0/16 resaux_uk`
- `docker run -itd --name conteneur1 --network resaux_uk alpine`
- `docker run -itd --name conteneur2 --network resaux_uk alpine`
- `docker run -itd --name conteneur3 alpine`
- Fait un ping du conteneur1 vers conteneur2. Que constatez-vous ? Pourquoi ?

Faisons du réseau avec Docker 4/4

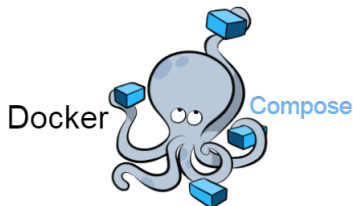
Ping sur le docker 0

Lancez les commandes suivantes puis dites ce quelles font.

- Faites un ping du conteneur2 vers conteneur1. Que constatez-vous ? Pourquoi ?
- Faites un ping du conteneur3 vers conteneur1. Que constatez-vous ? Pourquoi ?
- `docker run -itd --name conteneur4 --link conteneur3 alpine`
- Connectez-vous au container3 et affichez le contenu du fichier `/etc/hosts`. Faites la même chose pour le conteneur4

Docker Compose

C'est quoi une stack docker-compose ?



Définition

- Gestionnaire de conteneurs

Notion de service

Définition

1 service = 1 processus maître = 1 conteneur

TP

Comment déployer un environnement Wordpress ?

Solution

Identifier les différents services.

Docker compose : installation

Linux

```
sudo apt-get install docker compose --version
```

Windows

```
sudo apt-get install
```

Mac

```
sudo apt-get
```


TP : Utilisez docker compose

Déployez une application Laravel

- Créez un dossier gestion_evenement
- Copiez les fichiers dockerfile et docker-compose.yml
- Lancez la commande : `docker compose up -d`.
Que fait cette commande ?
- Rendez-vous à l'adresse IP suivante : `http://127.0.0.1:8081`

TP : Utilisez docker compose

Déployez une application Laravel

- Créez un dossier gestion_parking
- Copiez les fichiers dockerfile et docker-compose.yml
- Lancez la commande : `docker compose up -d`.
Que fait cette commande ?
- Rendez-vous à l'adresse IP suivante : `http://127.0.0.1:8081`

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**.

FIN