



Open source - Lecture 2

Création de conteneurs Docker



Rappels

- Virtualisation d'applications
 - Virtualisation « légère »
- Utilisation du kernel Linux host
- Isoler des applications
- Mise en production aisée du conteneur



Image Docker

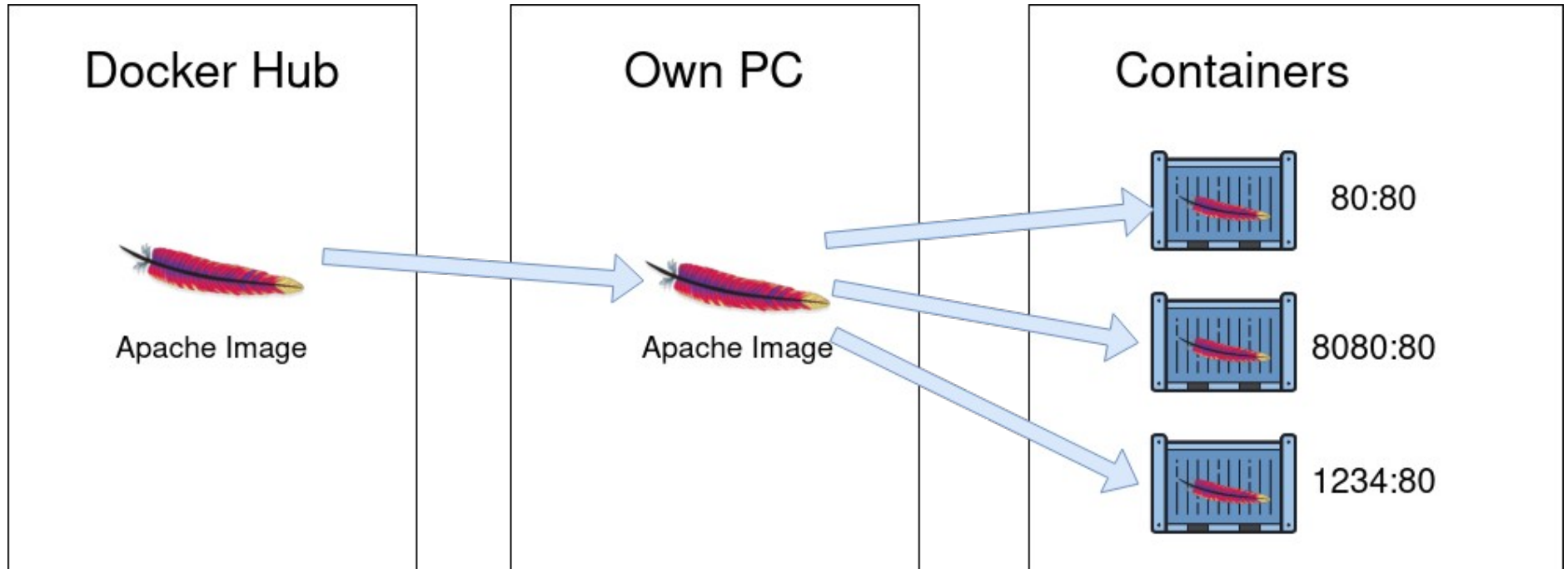
- Une image rassemble des
 - Bibliothèques
 - Binaires
 - File system
 - Dépendances
- => Tous les « datas » utiles pour faire tourner l'application



Container Docker

- Un container est une instance d'un image
 - Plusieurs instances peuvent tourner en même temps
 - Partagent tous la même base d'image
 - Sont indépendants l'un de l'autre
 - Spécification d'un port pour ne pas interférer avec les autres containers (8080:80) (80:80)

Image Docker



Lancer un container Docker

1) Récupérer l'image depuis les serveurs de docker

- `docker image pull Nom image`

2) Crée et lance un container

- `docker container run -p 8080:80 Nom image`
- « -p 8080:80 » permet de rediriger le port 8080 de l'host vers le port 80 du container



Informations Docker

- Docker image ls
 - Lister les images locales
- Docker container ls
 - Lister les containers existants en exécution
- Docker container ls -a
 - Lister les containers existants (tous)

Démo

- Docker image pull httpd
 - Récupère l'image httpd (apache) depuis le Docker Hub
- Docker container run -p 8080:80 httpd
 - Lance une instance de l'image httpd en redirigeant le port 8080 host vers le port 80 du container
- Localhost:8080
 - Nous avons une instance d'apache fonctionnelle !!

Démo

- Nouveau terminal
- Docker container ls
 - Liste les containers actifs
- Terminal 1 : Ctrl+C (arrêter le container)
- Docker container ls
 - Rien n'est affiché (normal car on a arrêté le container)
- Docker container ls -a
 - Affiche tous les containers, on voit bien le container arrêté

Démo

- Docker container run -p 8080:80 httpd
 - Redémarre le container arrêté précédemment ?
- Docker container ls -a
 - Nous avons maintenant deux entrées :
 - Le premier container arrêté
 - Le deuxième container démarré
- => **Docker container run lance un nouveau container**
- Docker container start *IdContainer*
 - Permet de redémarrer un container existant (background)



Docker image

Docker container ls

CONTAINER ID	IMAGE	COMMAND	CREATED	STATUS	PORTS	NAMES
988eed841885	httpd	"httpd-foreground"	25 minutes ago	Up 15 minutes	0.0.0.0:80->80/tcp	wonderful_merkle

- Container ID
 - Hash du container, permet de l'identifier de façon unique
- Image
 - Image de base
- Command
 - Commande exécutée lors du démarrage du container
- Name
 - Nom donné au container

Simplification docker run

- La commande « Docker run -p 8080:80 httpd » permet en une seule opération de :
 - Pull l'image si elle n'est pas existante sur la machine
 - Lancer un container de l'image
- Utilisez les commandes « détaillées » au début pour comprendre plus facilement !!



Démo : Container simultanés

- Docker container run -p 8080:80 httpd
 - Lance une première instance d'un container écoutant sur le port « 8080 »
- Docker container run -p 80:80 httpd
 - Lance une deuxième instance d'un container écoutant sur le port « 80 »

Docker suppression

- Docker container rm *IdContainer(s)*
 - Supprime un container (doit être arrêté)
- Docker container stop *IdContainer(s)*
 - Permet d'arrêter un container
- Docker container kill *IdContainer(s)*
 - Permet de forcer l'arrêt d'un container
- Docker **image** rm *IdImage(s)*
 - Permet de supprimer une image locale



Docker Volumes

- Utilisation des « volumes »
 - Permet de « linker » un dossier local dans le container
- Docker container run `-v localPath:ContainerPath`
 - *LocalPath* étant le path local de la machine hôte
 - *ContainerPath* étant le path dans le container

Démo : Docker Apache site peso

- « Link » du dossier contenant les fichiers HTML à Apache
- Création d'un fichier HTML dans le répertoire courant (test.html)
- Docker container run -p 8080:80 -v "\$PWD":/usr/local/apache2/htdocs/ httpd
- Localhost:8080/test.html
 - Affiche bien le fichier test.html
- Si il y a des problème, donner les permissions 777 à tout le dossiers