

DOCKER

Une formation pratique



Un peu de théorie : VMs vs conteneurs

VOUS PENSIEZ QUE J'ALLAIS PARLER DE
DOCKER ? RATÉ
(PROMIS, CE SERA RAPIDE)

Les VMs

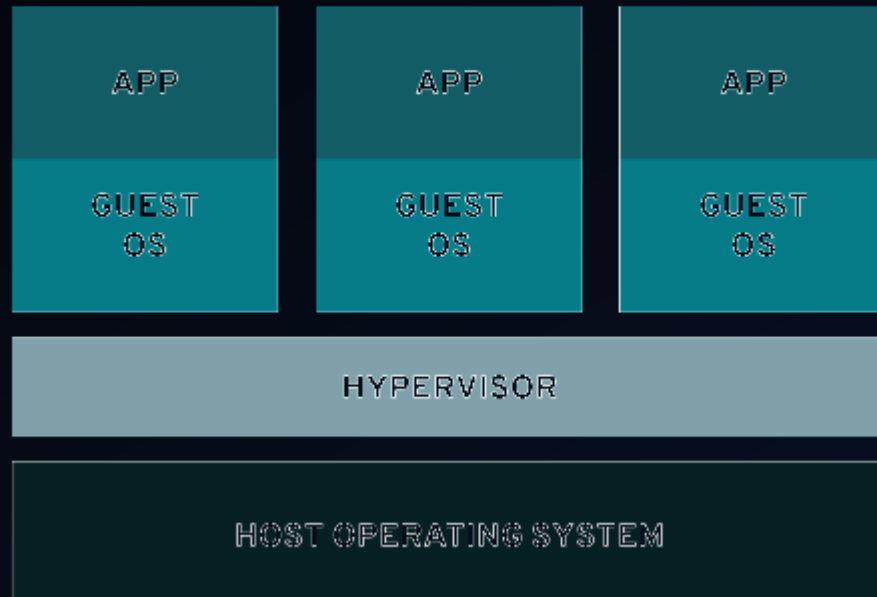
- Principe (simplifié) : faire tourner un OS invité complet au-dessus d'un autre OS hôte
- Avantage : abstraction de l'OS hôte, isolation complète, accès à toutes les capacités d'un OS complet
- Inconvénient : lourdeur en termes de déploiement et de performances

Les conteneurs

- Principe (simplifié) : faire tourner juste ce qu'il faut au-dessus de l'OS hôte pour faire tourner son projet
- Avantage : abstraction de l'OS hôte, légèreté
- Inconvénient : pas d'isolation totale, certaines ressources sont difficiles à accéder

VMs vs Docker

VIRTUALIZATION



VS.

CONTAINERS





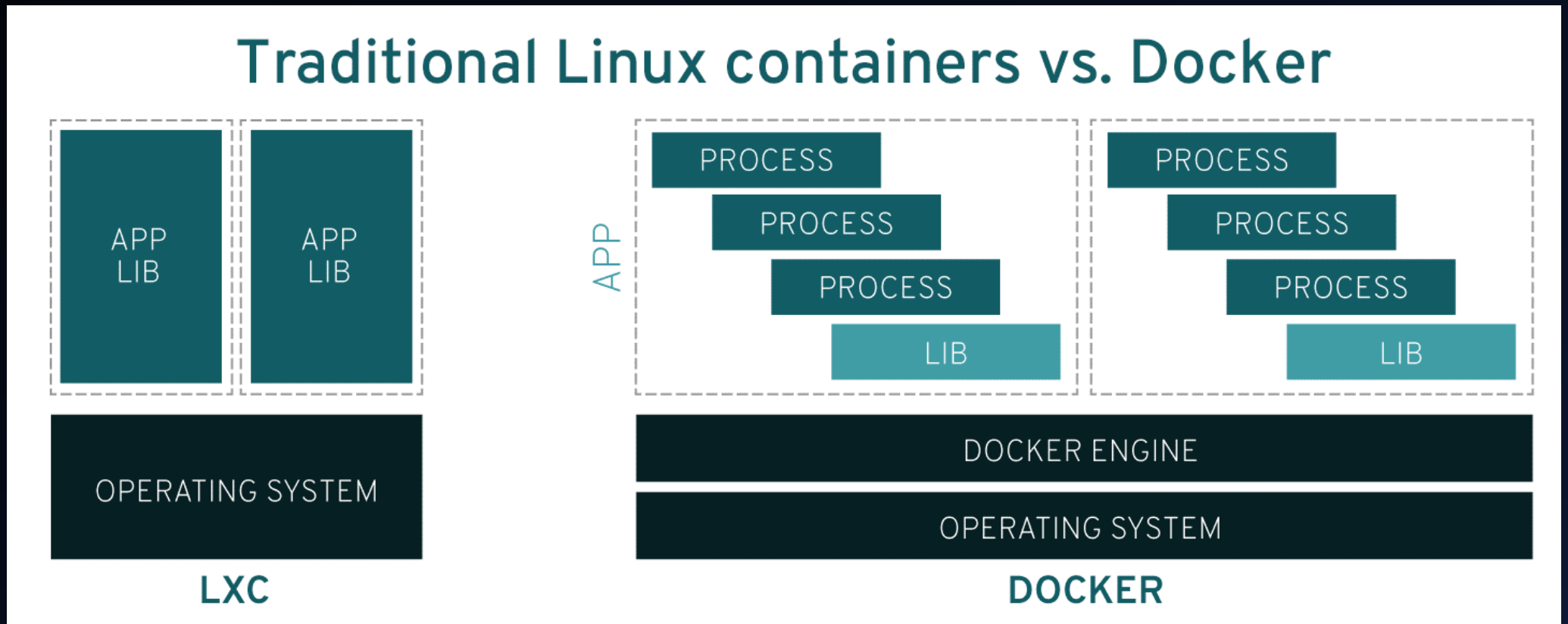
La pratique : Docker

ENFIN



Les concepts de base

- Les conteneurs Docker, pas des conteneurs classiques



Un construction par couches

- Les conteneurs Docker sont construits par couches :
 - Une pour chaque étape
- Le conteneur finale est obtenu en assemblant tous les conteneurs intermédiaires

Comment construire un conteneur Docker : les DockerFiles

- On y retrouve la construction par couche : une suite d'instructions
 - FROM : image de base
 - RUN : commande lancée lors de la construction de l'image
 - ENV : variables d'environnement
 - VOLUME : partie du système de fichiers du conteneur exposé
 - EXPOSE : ouvrir des ports du conteneurs
 - COPY
 - ADD : possibilité d'utiliser des urls et d'extraire des archives
 - CMD : commande lancée lorsque le conteneur est lancé sans argument ou commande particuliers
 - ENTRYPOINT : commande lancée lorsque le conteneur est lancé. Pas de possibilité de le remplacer mais possibilité de changer les arguments

Alpine : une distribution Linux orientée conteneurs

- Très légère : 8 Mio
- Très simple à utiliser
- Pensée pour être utilisée dans des conteneurs



alpine
Linux

Comment construire un conteneur Docker : les Dockerfile

- Exemple : l'image Docker nginx
- Exemple : construire ensemble une image Docker nginx

Lancer des conteneurs

- `docker run <options> <image_name> <command> <args...>`
- Options :
 - `-d`
 - `-p host_port:container_port`
 - `-v host_path:container_path`
 - `-e VAR=VALUE`
 - `--name container_name`
 - `--restart no/on-failure[:max-retries]/unless-stopped/always`

Interagir avec des conteneurs

- `docker exec <options> <image_name> <command>`
- Différence avec `run` :
 - Permet de passer outre le `cmd` et l'entrypoint de l'image
 - Permet aussi de lancer une commande après que le conteneur soit lancé
 - `-i` : interactive
 - `-t` : obtenir un terminal
 - `-it` : très utile lorsqu'on veut lancer un shell interactif
 - `--name container_name`

Construire des conteneurs

- Docker build <options> <path_to_dockerfile>
 - -t <image_name>
 - --build-args VAR=VALUE

D'autres actions sur les conteneurs

- `docker stop container_name`
- `docker rm container_name`
- `docker restart container_name`
- `docker ps`



Plusieurs dockers

ON PASSE DE 1 À 2, 3, BEAUCOUP



La communication entre conteneurs dockers

- Réseau privé virtuel pour la communication
- Interface virtuel pour la communication entre les conteneurs et l'hôte (et donc l'accès à Internet)
- Possibilité de créer ses propres réseaux pour isoler les conteneurs entre eux :
 - `docker network create network_name`
 - `docker run --network=network_name image_name`
 - `docker network connect network_name container_name`
 - `docker network rm network_name`

La communication entre conteneurs dockers

- Comment communiquer d'un conteneur à l'autre ?
 - IP : mauvaise idée, pas stable
 - Nom : lequel ? Celui du conteneur ? Comment retrouver l'ip du conteneur ?
- Docker fait tourner un serveur DNS
 - Le nom du conteneur est associé à son conteneur
 - Permet d'utiliser quelque chose de plus stable dans le temps que l'ip


Automatiser la construction et le lancement d'images


Docker : docker compose


- Les fichiers docker-compose.yml
 - Permet de créer et de faire interagir entre eux plusieurs conteneurs Docker
 - Permet de spécifier l'ordre de création des conteneurs

Automatiser des tests et le déploiement d'applications (CI/CD)


Test

python_check_... 


python_integr... 

python_unit... 

Build

angular_buil... 

Deploy

deploy_prod... 

deploy_prod... 