Docker

Definition
(10 Slides)

Docker et Containerisation

Container :
Une boite, un container, pour votre app + ses dépendances
(requests, flask, etc)

Docker :
Outil pour containeriser et exécuter des applications

La containerisation / Docker permettent de résoudre
le problème classique du développeur :

« mon code marche chez moi mais pas ailleurs »

Le container isole l'app et ses dépendances

Docker et Containerisation

Sur mon PC : ça marche

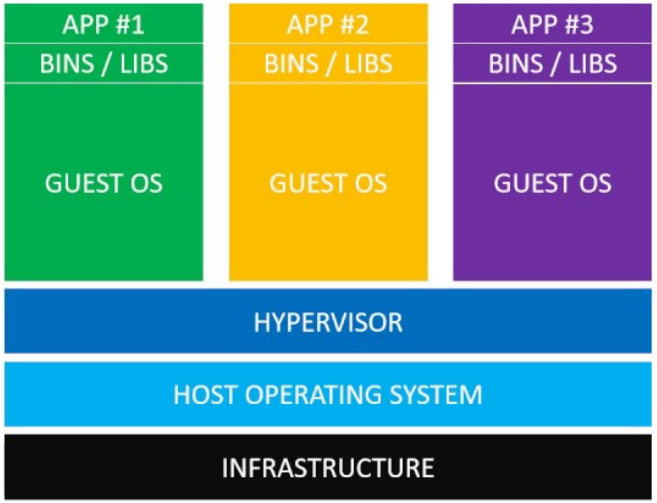
Sur le PC du voisin : ça casse

Causes : versions de Python, librairies, OS

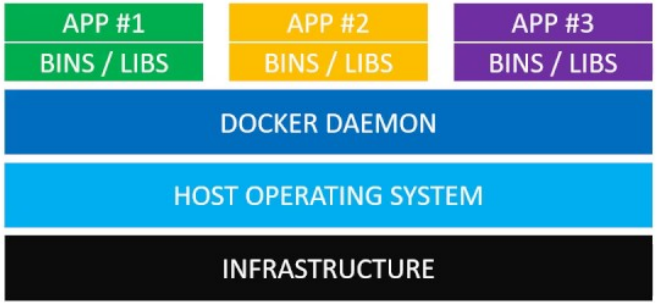
Même code,
même machine virtuelle,
mais OS ou librairies différentes = bugs

C'est le cauchemar du dev,
et surtout de l'ops !

VM vs Containers



Virtual Machines



Docker Containers

Notions d’image

Image Docker :
Code de l’application + dépendances

Container Docker :
Instance de l’image en fonctionnement

Concept clé :
Isolement et portabilité

Image Docker est comme un plan / modèle

Container Docker est l’instance de votre plan que vous pouvez utiliser partout

Dockerfile

Script pour construire une image

Instructions principales :

- FROM = choix image de base
- COPY = copier fichiers
- RUN = lancer des commandes shell
- CMD = lancer l'app

Le Dockerfile est comme un mode d'emploi pour fabriquer votre image.

```
FROM python:3.13.7-alpine

RUN adduser --disabled-password --gecos "" dailycatfacts_user
USER dailycatfacts_user

WORKDIR /app

COPY requirements.txt .

RUN pip install --no-cache-dir -r requirements.txt

COPY . .

CMD ["python3", "app/main.py"]
```

Commandes Docker essentielles

docker build -t nom_image .
→ créer une image

docker run -p 5000:5000 nom_image
→ lancer un container

docker ps
→ lister containers

docker stop <nom_ou_id_container>
→ arrêter un container

Un container est éphemère

- Container = jetable
- Les données disparaissent si non sauvegardées
- Precisez l'option -v /chemin/hote:/chemin/container → persistance

- Sans volume, vos données disparaissent à chaque container
- Avec Docker Volume, elles sont stockées sur votre disque

Pourquoi faire du Docker dans notre module ?

- Tester toutes les applis de la même manière
- Partager un projet → tout le monde exécute pareil
- Préparer le passage à CI/CD

Exercice 1 : Containeriser votre application web avec Docker

- Objectif :
  - Créer un Dockerfile
  - Lancer votre application containerisée

Exercice 2 : Persistance des données (volumes)

- Objectif :
  - Comprendre volumes
  - Sauvegarder pet.json

- Livrable :
  - Commande docker run avec volume

Exercice 3 : Reproductibilité

- Objectif :
  - Upload votre Dockerfile sur Github
  - Lancer l'application web d'un camarade / voisin, depuis son Github
- Règle :
  - Aucun pip install sur la machine hôte (desactivez vos venv)
  - L'application de votre voisin devrait s'executer sans installation de librairie

Exercice 4 : Bonus (rapides)

- Ajouter .dockerignore
- Changer le port
- Changer le nom de l’image
- Relancer votre application web