

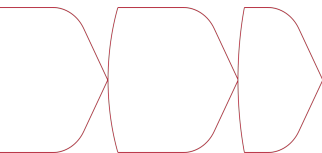


# Introduction aux Conteneurs et à Docker

## Comprendre les bases des conteneurs et de Docker

Slimani Mohamed Amine

January 31, 2025



# Sommaire

Qu'est-ce qu'un conteneur ?

Qu'est-ce que Docker ?

Concepts de base de Docker

Cycle de vie d'un conteneur

Dockerfile

Docker Compose

Bonnes pratiques

Outils pour travailler avec Docker

Exemple de commandes Docker

Pourquoi c'est important ?

# Qu'est-ce qu'un conteneur ?

- ▶ **Définition** : Un conteneur est une unité logicielle qui encapsule une application et ses dépendances.
- ▶ **Avantages** : Isolation, portabilité, légèreté.
- ▶ **Comparaison avec les machines virtuelles** : Les conteneurs partagent le noyau du système hôte, ce qui les rend plus légers et plus rapides.

# Qu'est-ce que Docker ?

- ▶ **Définition** : Docker est une plateforme open-source pour développer, expédier et exécuter des applications dans des conteneurs.
- ▶ **Composants** : Docker Engine, Docker Hub, Docker Compose.
- ▶ **Avantages** : Simplifie le déploiement, améliore la productivité, facilite la collaboration.

# Concepts de base de Docker

- ▶ **Images** : Modèles en lecture seule pour créer des conteneurs.
- ▶ **Conteneurs** : Instances d'exécution d'une image.
- ▶ **Volumes** : Persistance des données.
- ▶ **Réseaux** : Communication entre conteneurs.

# Cycle de vie d'un conteneur

- ▶ **Créer une image** : À partir d'un Dockerfile.
- ▶ **Lancer un conteneur** : À partir d'une image.
- ▶ **Arrêter et redémarrer un conteneur.**
- ▶ **Supprimer un conteneur.**

# Dockerfile

## Exemple de Dockerfile

```
FROM ubuntu:20.04
RUN apt-get update && apt-get install -y python3
COPY . /app
WORKDIR /app
CMD ["python3", "app.py"]
```

# Docker Compose

## Exemple de fichier docker-compose.yml

```
version: '3'
services:
  web:
    image: nginx
    ports:
      - "80:80"
  db:
    image: postgres
    environment:
      POSTGRES_PASSWORD: example
```



# Bonnes pratiques

- ▶ **Utiliser des images officielles** : Pour garantir la sécurité et la stabilité.
- ▶ **Minimiser la taille des images** : En utilisant des images de base légères et en supprimant les fichiers inutiles.
- ▶ **Utiliser des volumes pour les données persistantes** : Pour éviter de perdre des données lors de la suppression d'un conteneur.

# Outils pour travailler avec Docker

- ▶ **Docker CLI** : Interface en ligne de commande pour gérer les conteneurs.
- ▶ **Portainer** : Interface graphique pour gérer les conteneurs.
- ▶ **Kubernetes** : Orchestration de conteneurs à grande échelle.

# Exemple de commandes Docker

## Commandes Docker

```
# Construire une image
docker build -t mon-image .

# Lancer un conteneur
docker run -d -p 80:80 mon-image

# Lister les conteneurs en cours d'exécution
docker ps
```

# Pourquoi c'est important ?

- ▶ Les conteneurs et Docker sont essentiels pour le développement moderne et le déploiement d'applications.
- ▶ Ils améliorent la portabilité, la productivité et la collaboration.
- ▶ Comprendre leur fonctionnement est crucial pour les développeurs et les administrateurs système.

**Les conteneurs et Docker** sont des technologies clés pour le développement et le déploiement d'applications modernes. Ils offrent isolation, portabilité et légèreté.