

Guide Complet : Configuration et Déploiement de Docker

Par : Raphael Ferreira

Introduction

Docker est une plateforme permettant de créer, déployer et exécuter des applications en conteneurs de manière efficace et portable. Ce guide vous fournira toutes les étapes nécessaires pour installer, configurer et gérer Docker sur différents systèmes d'exploitation, ainsi que des conseils avancés sur l'optimisation des performances.

1. Installation de Docker

1.1 Installation sur Windows

1. Téléchargez Docker Desktop depuis le site officiel : <https://www.docker.com/get-started>
2. Exécutez l'installateur et suivez les instructions.
3. Activez la virtualisation dans le BIOS si nécessaire.
4. Vérifiez que WSL 2 est installé et activé pour une meilleure compatibilité.
5. Redémarrez votre ordinateur et testez l'installation avec :

```
docker --version
```

1.2 Installation sur Linux

1. Mettez à jour vos paquets :

```
sudo apt update && sudo apt upgrade -y
```

2. Installez Docker :

```
sudo apt install docker.io -y
```

3. Démarrez et activez Docker au démarrage :

```
sudo systemctl start docker
```

```
sudo systemctl enable docker
```

4. Vérifiez que Docker fonctionne avec :

```
docker run hello-world
```

1.3 Installation sur macOS

1. Téléchargez Docker Desktop pour Mac depuis le site officiel.
2. Installez l'application et ouvrez-la.
3. Vérifiez l'installation avec :

```
docker --version
```

2. Configuration de Docker

Une fois Docker installé, il est essentiel de bien le configurer pour optimiser ses performances et éviter les problèmes courants.

2.1 Configuration des ressources

Sur Docker Desktop (Windows/Mac), ajustez les paramètres CPU et RAM dans les préférences.

Sur Linux, vous pouvez limiter les ressources d'un conteneur avec :

```
docker run --memory=512m --cpus=1 my_container
```

2.2 Configuration des réseaux

Docker propose plusieurs modes réseau : bridge, host, none et overlay.

Créer un réseau personnalisé :

```
docker network create mon_reseau
```

3. Gestion des Conteneurs

Lister tous les conteneurs en cours d'exécution :

```
docker ps
```

Démarrer un conteneur :

```
docker start mon_conteneur
```

Arrêter un conteneur :

```
docker stop mon_conteneur
```

4. Déploiement avec Docker Compose

Docker Compose permet de gérer plusieurs conteneurs avec un fichier YAML.

Exemple de fichier `docker-compose.yml` pour un serveur web :

```
version: '3'
services:
  web:
    image: nginx
    ports:
      - "80:80"
```

5. Bonnes Pratiques et Optimisation

- Utiliser des images légères comme Alpine Linux.
- Nettoyer régulièrement les conteneurs et images inutilisés avec ``docker system prune``.
- Configurer des volumes pour persister les données : ``docker volume create mon_volume``.

6. Dépannage et Résolution des Problèmes

- Vérifier les logs d'un conteneur avec ``docker logs mon_conteneur``.
- Vérifier l'espace disque utilisé par Docker : ``docker system df``.