

# Le métier de DevOps et Docker



# Qui suis-je ?

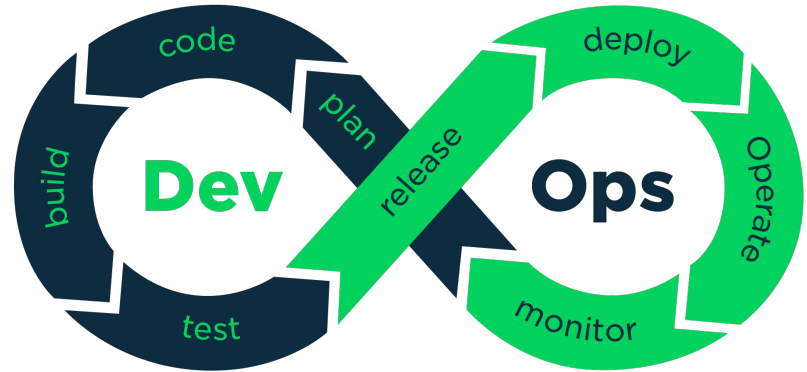


# Qu'est-ce que le DevOps ?

- Automatisation
- Qualité
- Agilité

## PARTIE 1 :

## Le Métier de DevOps



# Les missions clés du DevOps

- Code
- Continuous Integration (CI)
- Continuous Development (CD)
- Automatisation
- Monitoring
- Infrastructure as Code (IaC)
- Sécurité

# Outils et Compétences

Catégorie	Outils Populaires	Compétences Associées
Conteneurisation	Docker, Kubernetes, Helm	Gestion des conteneurs, Orchestration
CI/CD	Jenkins, Gitlab CI/CD, Github Actions, CircleCI	Création de pipelines, automatisation des builds
Versionnement	Git, Github, Gitlab, Bitbucket	Gestion de versions, collaboration en équipes
Infrastructure	Terraform, Ansible, Puppet	Infrastructure as Code (IaC)
Surveillance	Prometheus, Grafana, ELK (Elasticsearch, Logstash, Kibana)	Monitoring des performances, logs centralisés.
Cloud	AWS, Azur, Google Cloud Platform	Déploiement dans le cloud, scalabilité

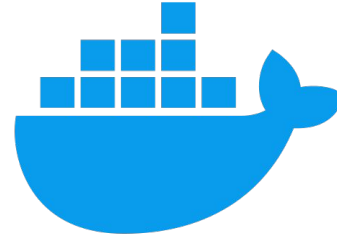
# Méthode Agile et DevOps

- **Agile** : Planification, développement, feedback utilisateur.
- **DevOps** : Intégration continue, tests, automatisation.
- Livraisons fréquentes de logiciels fonctionnels.
- Adaptation continue aux besoins changeants.
- Collaboration étroite entre les équipes.

# Qu'est-ce que Docker et la Conteneurisation

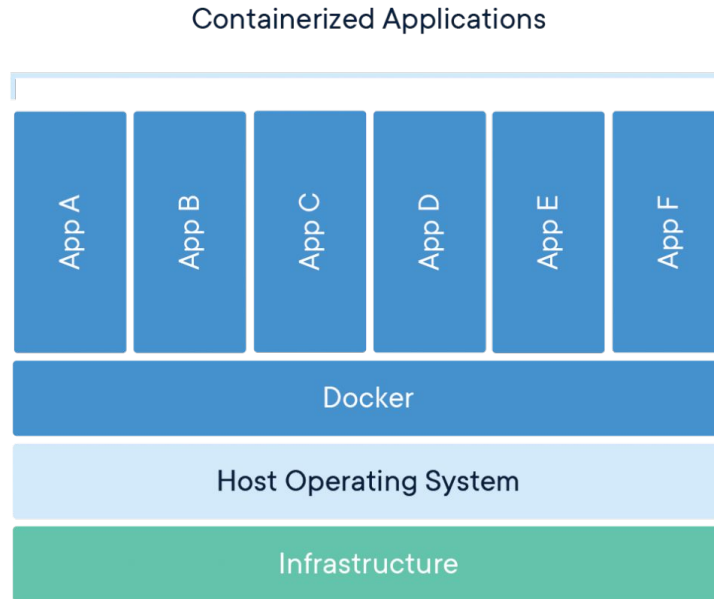
## PARTIE 2 :

## Docker



- Conteneurs
- Portabilité
- Isolation

# Fonctionnement de Docker





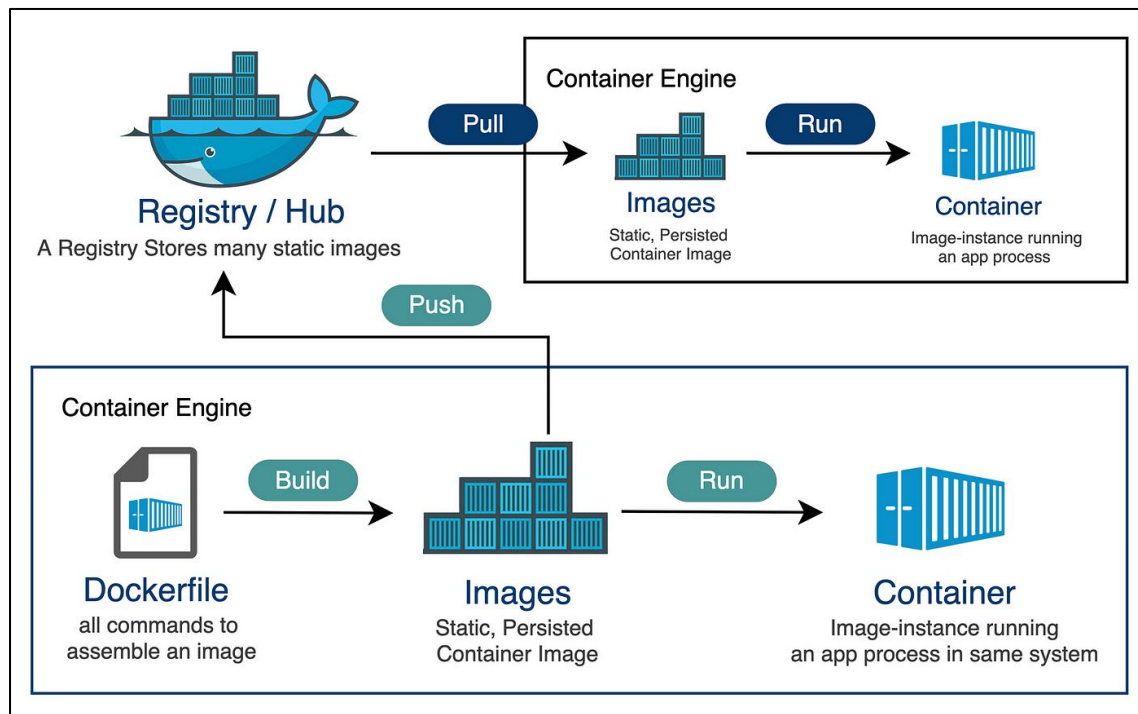
# Fonctionnement de Docker et Dockerfile

- Images
- Conteneurs
- Docker Engine

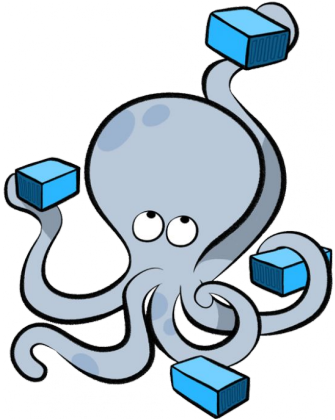
```
1 FROM node:alpine
2 WORKDIR /app
3 COPY . .
4 RUN npm install
5 CMD ["node", "server.js"]
```

# Workflow Docker (BRS)

Build -> Run -> Share

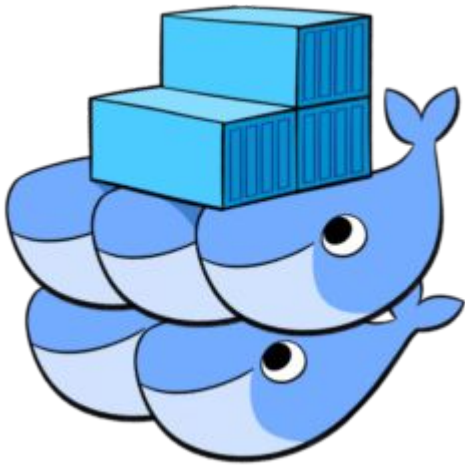


# Docker Compose



```
1 services:
2   web:
3     image: myapp
4     ports:
5       - "8080:80"
6   db:
7     image: postgres
8     environment:
9       POSTGRES_PASSWORD: example
```

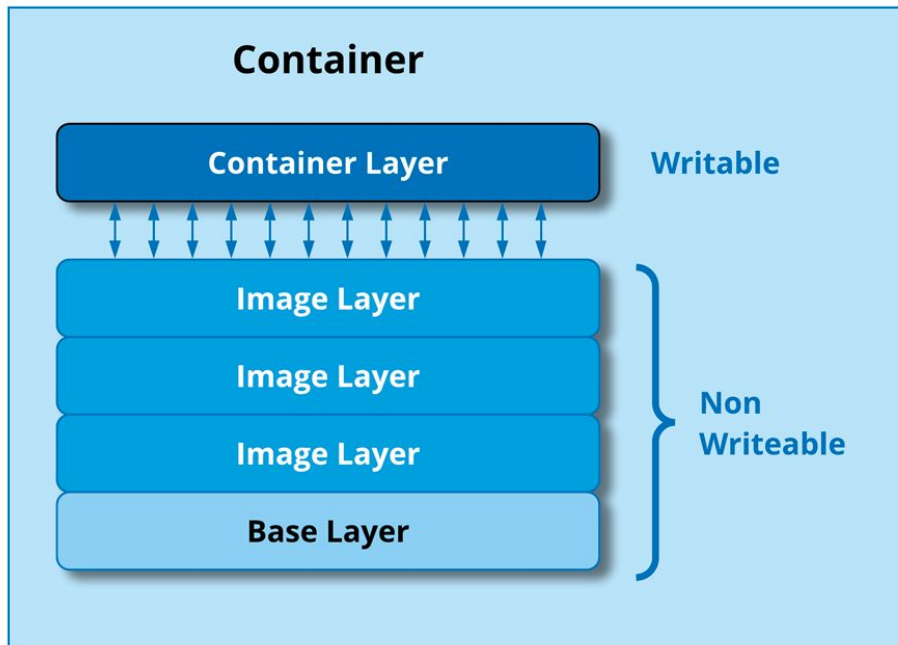
# Docker Swarm Et Kubernetes



# Sécurité et Best Practice

- Utiliser des images officielles et vérifiées
- Limiter les privilèges
- Scanner les images Docker pour des vulnérabilités
- Sécurisation du Docker Daemon
- Utiliser des réseaux Docker isolés

# Dive Into Images



# **PARTIE 3 :**

## **La Pratique**

### **Pré-Requis**

- Node
- Python
- Docker Desktop / Engine
- Git
- Dive (optionnel)

# Installation de Git



```
$ git --version  
$ sudo apt update  
$ sudo apt install git
```



# Git Post Install



```
$ git config --global user.email "youremail@example.com"  
$ git config --global user.name "Your Name"
```

# Installation Python



```
$ python3 --version  
$ sudo apt update  
$ sudo apt install python3  
$ sudo apt install python3-pip
```

# Installation Node



```
$ curl -o- https://raw.githubusercontent.com/nvm-sh/nvm/v0.40.0/install.sh | bash
$ nvm install --lts
$ node -v
$ npm -v
```

# Installation Docker Desktop



```
$ sudo apt install gnome-terminal #Si vous n'utilisez pas le DE GNOME  
$ sudo apt update  
$ sudo apt install ./docker-desktop-amd64.deb
```