

# Docker Compose

## Introduction

**Docker Compose** est une commande docker permettant de définir et de gérer des applications multi-conteneurs.

La commande utilise un fichier YAML (`docker-compose.yml`) pour décrire les services, les réseaux, les volumes, et les dépendances entre les conteneurs.

Ci-dessous l'exemple d'un fichier `docker-compose.yml`. Il s'agit d'un projet web avec une API Node.js, une base de données PostgreSQL, et un reverse proxy Nginx.

```
services:
  nginx:
    image: nginx
    ports:
      - "80:80"
    volumes:
      - ./nginx.conf:/etc/nginx/nginx.conf
    depends_on:
      - web

  web:
    build: ./web
    environment:
      - NODE_ENV=production
    depends_on:
      - db

  db:
    image: postgres
    environment:
      - POSTGRES_PASSWORD=secret
    volumes:
      - pgdata:/var/lib/postgresql/data
```

```
volumes:  
  pgdata:
```

La "traduction" en ligne de commandes `docker` ressemblerait à ceci :

```
# création du volume pour PostgreSQL  
docker volume create pgdata  
  
# création de l'image pour le serveur web  
docker build -t web-image ./web  
  
# lancement des conteneurs  
docker run -d --name db -e POSTGRES_PASSWORD=secret -v pgdata:/var/lib/postgresql/data  
postgres  
docker run -d --name web -e NODE_ENV=production web-image  
docker run -d --name nginx --network frontnet -p 80:80 -v ./nginx.conf:/etc/nginx/nginx.conf  
nginx
```

## Orchestration

L'orchestration d'une stack docker compose s'effectue avec la commande `docker compose`.

```
docker compose up # démarre les conteneurs  
docker compose down # arrête et supprime les conteneurs, réseaux et volumes  
docker compose build # construit les images des conteneurs  
docker compose logs # affiche les logs des conteneurs
```

## Syntaxe

### Services

La clé `services` définit les conteneurs à lancer. Chaque service représente un conteneur Docker.

```
services:  
  web:  
    image: nginx:latest
```

Pour chaque service, les instructions suivantes sont disponibles.

Instruction	Description	Exemple
<code>image</code>	Image de départ	<code>image: ubuntu:24:04</code>
<code>container_name</code>	Nom du conteneur	<code>container_name: mon_web</code>
<code>ports</code>	Mappe les ports de la machine hôte vers le conteneur	<pre>ports:   - 8080:80</pre>
<code>volumes</code>	Monte des volumes dans le conteneur. Volume mappés ou managés.	<pre>volumes:   - ./mon_dossier:/app</pre>
<code>environment</code>	Variables d'environnement	<pre>environment:   - DEBUG=1</pre>
<code>command</code>	Commande au démarrage du conteneur	<code>command: ["python", "app.py"]</code>
<code>depends_on</code>	Ordre de création des conteneurs	<code>depends_on: ["db"]</code>
<code>restart</code>	Politique de redémarrage ( <code>no</code> , <code>always</code> , <code>on-failure</code> , <code>unless-stopped</code> )	<code>restart: always</code>
<code>networks</code>	Réseaux du conteneur	<pre>networks:   - frontend</pre>

## Volumes

La clé `volumes` permet d'automatiser la création et le montage des volumes managés.

```
volumes:
  db_data # volume sans paramètres spécifiques
  external_volume:
```

```
external: true
name: mon_volume_externe
```

Instruction	Description	Exemple
<code>external</code>	Indique si le volume a été créé en dehors de la stack docker compose.	<code>external: true</code>
<code>name</code>	Nom du volume, utile avec <code>external</code>	<code>name: mon_volume_externe</code>

## Networks

La clé `networks` définit les différents réseaux docker de la stack.

```
networks:
  reseau1:
    driver: bridge
  reseau2:
    driver: bridge
  ipam:
    config:
      - subnet: "172.16.0.0/16"
  reseau_externe:
    external: true
    name: mon_reseau_externe
```

Instruction	Description	Exemple
<code>driver</code>	Type de réseau ( <code>bridge</code> , <code>host</code> , etc.)	<code>driver: bridge</code>
<code>external</code>	Indique si le réseau a été créé en dehors de la stack docker compose.	<code>external: true</code>
<code>name</code>	Nom du réseau, utile avec <code>external</code>	<code>name: mon_reseau_externe</code>
<code>ipam</code>	Configuration IP du réseau	Voir exemple ci-dessus.

Revision #22

Created 16 July 2025 09:35:33 by Thibaud FRICHET

Updated 19 September 2025 08:23:19 by Thibaud FRICHET