



docker

Introduction à Docker



- Outil qui permet
 - Créer
 - gérer
 - déployer
- Facilite le déploiement
- Assure que l'application fonctionne de la même manière sur toutes les machines.

Qu'est-ce qu'un conteneur



- Contient tout ce dont ton application a besoin pour fonctionner
 - code
 - bibliothèque
 - etc

Initialisation d'un conteneur



- Docker init

```
Welcome to the Docker Init CLI!
```

```
This utility will walk you through creating the following files with sensible defaults for your project:
```

- .dockerignore
- Dockerfile
- compose.yaml
- README.Docker.md

```
Let's get started!
```

```
? What application platform does your project use? [Use arrows to move, type to filter]
```

- Go - suitable for a Go server application
- Python - suitable for a Python server application
- Node - suitable for a Node server application
- Rust - suitable for a Rust server application
- ASP.NET Core - suitable for an ASP.NET Core application
- PHP with Apache - suitable for a PHP web application
- Java - suitable for a Java application that uses Maven and packages as an uber jar

```
> Other - general purpose starting point for containerizing your application
```

```
Don't see something you need? Let us know!
```

```
Quit
```

```
? What application platform does your project use? PHP with Apache
? What version of PHP do you want to use? 8.3.0
? Please enter the relative directory (with a leading .) for your app: ./ (current directory)
? What port do you want to use to access your app? (9000) 
```

```
✓ Created → .dockerignore
✓ Created → Dockerfile
✓ Created → compose.yaml
✓ Created → README.Docker.md
```

→ **Your Docker files are ready!**

Review your Docker files and tailor them to your application.

Consult README.Docker.md for information about using the generated files.

What's next?

Start your application by running → `docker compose up --build`

Your application will be available at `http://localhost:5000`

Bien configurer son conteneur



- Dans compose.yaml

```
services:
  server:
    build:
      context: .
    ports:
      - 9000:80
    volumes:
      - ../var/www/html/
    depends_on:
      - db
  db:
    image: mysql:5.7
    container_name: mysql
    restart: always
    environment:
      MYSQL_ROOT_PASSWORD: rootpassword
      MYSQL_DATABASE: mydatabase
      MYSQL_USER: myuser
      MYSQL_PASSWORD: mypassword
    ports:
      - "3306:3306"
    volumes:
      - db_data:/var/lib/mysql
  phpmyadmin:
    image: phpmyadmin/phpmyadmin
    container_name: phpmyadmin
    restart: always
    ports:
      - "8081:80"
    environment:
      PMA_HOST: db
      MYSQL_ROOT_PASSWORD:

volumes:
  db_data:
```

Nom du service

Démarre le service db

paramètre de la db

sauvegarde les données de la db

Indique ou se trouve la db

build



- > Se rendre dans le dossier qui contient le fichier compose.yaml
- > Exécuter la commande "**docker-compose up**"
- > Accéder à l'application
 - application web : <http://localhost:9000>
 - phpMyAdmin : <http://localhost:8081>



Merci beaucoup