

Guide d'Installation et de Déploiement

Ce projet regroupe trois composants essentiels :

- PostgreSQL pour la base de données
- Backend Node.js/Express pour la logique serveur
- Frontend React.js pour l'interface utilisateur

Ce guide vous aidera à installer et lancer l'ensemble via Docker et Docker Compose.

Prérequis:

- Un système Windows ou Mac
- Une connexion internet stable
- Un accès administrateur sur votre machine
- Docker Desktop installé

1. Installer Docker Desktop

Pour Windows / Mac:

1. Téléchargement:

Rendez-vous sur le site officiel de Docker Desktop et téléchargez la version adaptée à votre système.

2. Installation:

- Exécutez l'installateur et suivez les instructions.
- Sur Windows : Activez WSL2 si demandé.
- Redémarrez votre ordinateur après l'installation.

3. Vérification:

Exécutez "docker --version" pour vérifier l'installation.

2. Structure du Projet:

HubUniversity/

■■■■ docker-compose.yml

■■■■ database/

■ ■■■■ init.sql

■■■■ backend/

■ ■■■■ Dockerfile

■ ■■■■ (vos fichiers backend : server.js, etc.)

■■■■ frontend/

■■■■ package.json

■■■■ public/

■■■■ src/

■■■■ App.js

■■■■ firebase.js

3. Fichiers Clés:

3.1. docker-compose.yml:

Ce fichier définit trois services: PostgreSQL, Backend et Frontend.

version: '3.8'

services:

db:

image: postgres:17

container_name: projet_universite_db

environment:

POSTGRES_USER: postgres

POSTGRES_HOST_AUTH_METHOD: trust

POSTGRES_DB: projet_universite

ports:

- "5432:5432"

volumes:

- ./database/init.sql:/docker-entrypoint-initdb.d/init.sql

healthcheck:

```
test: ["CMD-SHELL", "pg_isready -U postgres"]
interval: 5s
timeout: 5s
retries: 5
backend:
build: ./backend
container_name: projet_universite_backend
environment:
- DATABASE_URL=postgres://postgres@db:5432/projet_universite
- NODE_ENV=production
depends_on:
- db
ports:
- "5001:5001"
volumes:
- ./backend:/app
- /app/node_modules
frontend:
image: node:18
container_name: projet_universite_frontend
working_dir: /app
environment:
- CHOKIDAR_USEPOLLING=true
- HOST=0.0.0.0
volumes:
- ./frontend:/app
- /app/node_modules
ports:
- "3000:3000"
command: sh -c "npm install && npm start"
```

3.2. database/init.sql:

Contenu pour creer la table users et inserer des donnees.

3.3. Backend:

Exemple de Dockerfile dans backend/:

```
-----
FROM node:18
WORKDIR /app
COPY package*.json ./
RUN npm install
COPY . .
EXPOSE 5001
CMD ["node", "server.js"]
-----
```

3.4. Frontend:

Pour initialiser et installer le frontend React.js:

- npx create-react-app .
- npm install axios react-router-dom firebase

4. Lancer le Projet:

Dans la racine du projet, lancez:

```
docker-compose up --build -d
```

Acces:

- Backend: <http://localhost:5001/api>
- Frontend: <http://localhost:3000>

5. Connexion a PostgreSQL:

Via Docker:

```
docker exec -it projet_universite_db psql -U postgres -d projet_universite
```

6. Arret et Suppression:

```
docker-compose down
```

```
docker-compose down -v
```

7. Depannage:

- Verifier Docker avec "docker info"
- Consulter les logs via "docker logs <container_name>"