

# Compte rendu Docker Compose

## Back-end

Tout d'abord j'ai initialisé le projet adonis avec npm.

Unset

```
npm init adonisjs@latest back
```

J'ai ensuite sélectionné "API Starter Kit" pour démarrer un projet pour une API REST, puis skip pour l'authentification, n'ayant pas besoin pour mon projet, puis pour la base de données j'ai décidé de partir sur postgresql, étant la technologie la plus sûre avec son système de rollback et la popularité. Une fois fait, je suis allé dans le dossier "back" créé par le CLI et j'ai installé les dépendances.

Unset

```
npm install
```

Enfin, j'ai lancé le projet pour savoir si tout fonctionnait correctement.

Unset

```
npm run dev
```

Lors du développement de l'application j'ai utilisé les commandes Ace (<https://docs.adonisjs.com/guides/ace/introduction>) pour me générer des contrôleurs, des modèles ainsi que des migrations pour ma base de données.

## Front-End

# Compte rendu Docker Compose

Pour ce qui est du front-end, j'ai tout d'abord générer un projet vite:

Unset

```
npm create vite@latest front
```

Puis sélectionner React, puis Typescript pour avoir un typage supplémentaire. Une fois fait je suis aller dans le dossier créé puis j'ai fais l'installation des dépendances:

Unset

```
cd front  
npm install
```

Et enfin j'ai lancé le projet pour savoir si tout était bon:

Unset

```
npm run dev
```

## Dockerfile

Une fois les deux projets initier, j'ai fait le Dockerfile pour chacun des services (front et back). Pour la back, je me suis simplement basé sur une image NodeJS de base, que j'ai retouché, en changeant simplement le "CMD" avec une commande ace:

Unset

```
node ace serve --watch
```

# Compte rendu Docker Compose

J'ai ensuite rajouté un `.dockerignore` pour ne pas avoir de problème lors du copy. J'ai fait de même pour le front en reprenant ce que j'avais déjà fait avant.

Pour exposer mon application au network docker, sur mon back j'ai simplement modifier l'adresse du host `0.0.0.0`, dans le `.env`. Pour ce qui est du front, j'ai modifié le host ainsi que le port dans le fichier `vite.setup.ts`.

Même chose pour le front, j'ai simplement changé le CMD a la fin ainsi que le port exposer:

Unset

```
npm run dev
```

## Docker compose

Pour le `compose.yaml`, j'ai créé trois services:

- Un service pour le backend (`adonisjs-app`)
- Un service pour le frontend (`react-app`)
- Un service pour la base de données (`postgresql`)

Les variables d'environnement sont récupérées depuis le `.env` à la racine du projet. J'ai aussi créé des volumes pour le back ainsi que le front. Cela permet de relancer automatiquement les images lorsqu'il y a un changement dans le code. Pour le back, j'ai mis une dépendance avec le service `postgresql`, afin que la base de données se lance avant le back. Enfin, j'ai ensuite créé un volume pour la base de données afin de pouvoir sauvegarder les données de la base de données et lancer un script sql pour initialiser les tables.