Mise en production

Déploiement sur Mogenius

FRITZ Florian, n° étudiant : 31810971

02/12/2022

Sommaire

1. BACK END

Dans un premier temps l'idée était de déployer un Docker de l'API backend sur Mogenius. Pour cela on build déjà un docker avec « docker build -t covid.api . ». Le soucis est qu'on ne peut pas accéder à la base de données PostgresSQL au sein du container.

Dockerfile
```
FROM eclipse-temurin:17-jre
COPY build/libs/covid-api-0.0.1-SNAPSHOT.jar /app/app.jar
WORKDIR /app
EXPOSE 8080
CMD ["java", "-jar", "app.jar"]
```

Figure 1: Dockerfile

On obtient donc une erreur légitime au moment du build. Une première idée aurait pu être d'essayer de créer un environnement postgre et le définir dans le Dockerfile mais cela ne fonctionnait pas.

On peut aussi créer un docker-compose.

docker-compose.yml
```
version: '3.8'
services:
  db:
    image: postgres:14.1-alpine
    restart: always
    environment:
      - POSTGRES_USER=${POSTGRES_USER}
      - POSTGRES_PASSWORD=${POSTGRES_PASSWORD}
    ports:
      - '5432:5432'
    volumes:
      - db:/var/lib/postgresql/data
      - ./db/init.sql:/docker-entrypoint-initdb.d/create_tables.sql
  api:
    container_name: covid-api
    build:
      context: ./
```

Figure 2: docker-compose.yml

Nous spécifions des variables d'environnement pour que le code fonctionne à la fois en local et sur le code déployé.

Par exemple, dans la configuration .yaml.

```yaml
spring:
  datasource:
    url: jdbc:postgresql://${ENDPOINT}:${PORT}/${POSTGRES_DB}
    username: ${POSTGRES_USER}
    password: ${POSTGRES_PASSWORD}
  jpa:
    database-platform: org.hibernate.dialect.PostgreSQLDialect
    open-in-view: false
    show-sql: true
    hibernate.ddl-auto: create
  liquibase:
    enabled: false
```

Figure 3: application.yaml

Cela fonctionne bien en local, malheureusement nous ne pouvons pas vérifier le déploiement puisque le docker build échoue.

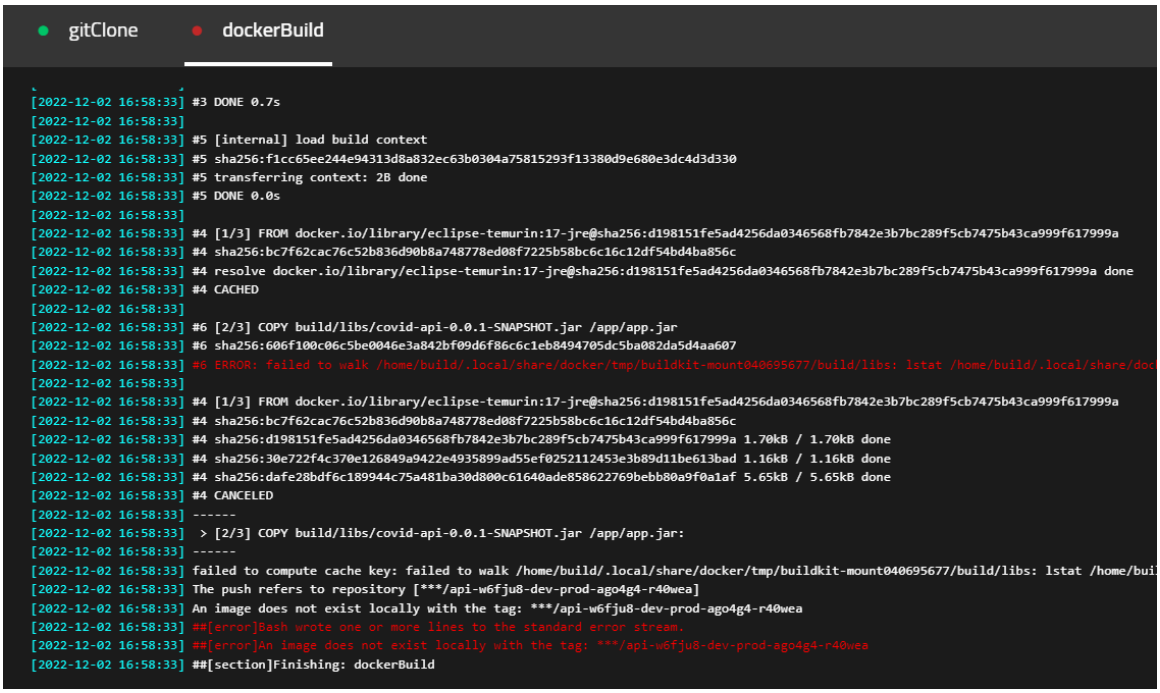
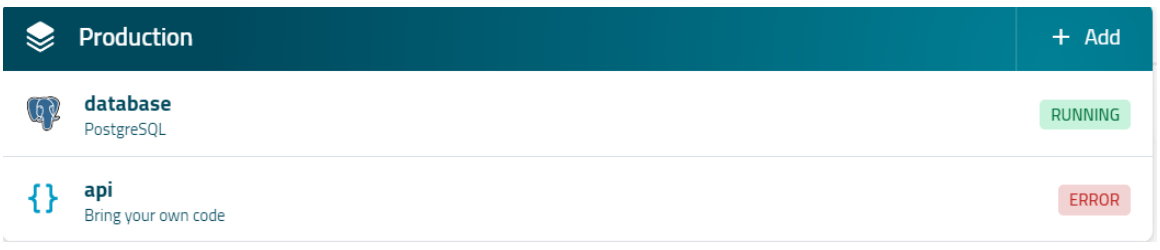


Figure 4: Code erreur Mogenius

Nous avons également créé un deuxième service, pour la base de données postgresSQL. Il nous faut pouvoir établir un lien entre ce service et le service de l'api.



Les variables d'environnements ont également été spécifié, mais le lien à établir n'a pas été trouvé.

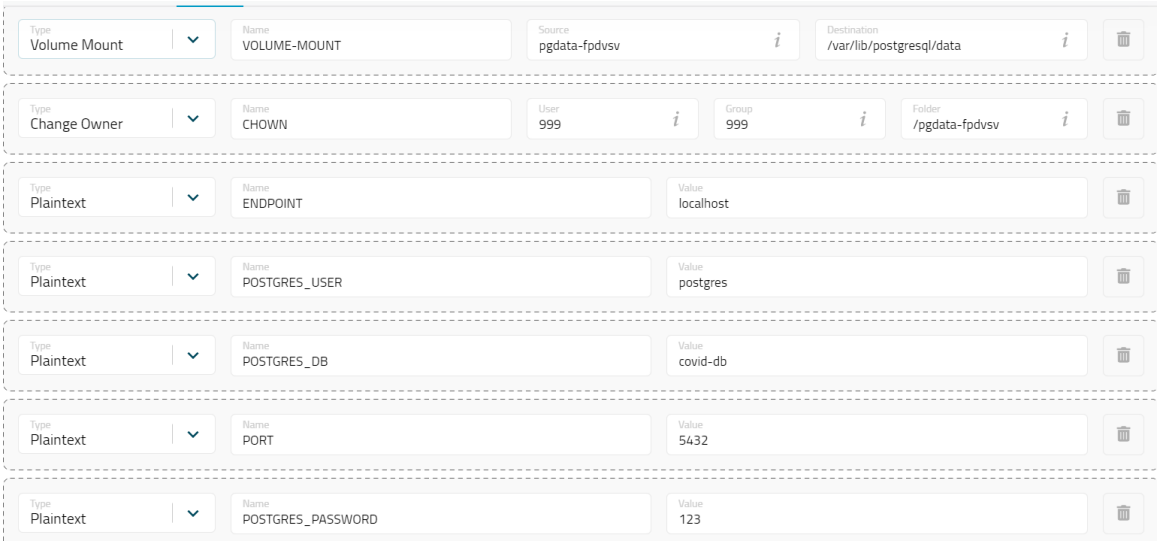


Figure 5: Variables d'environnement

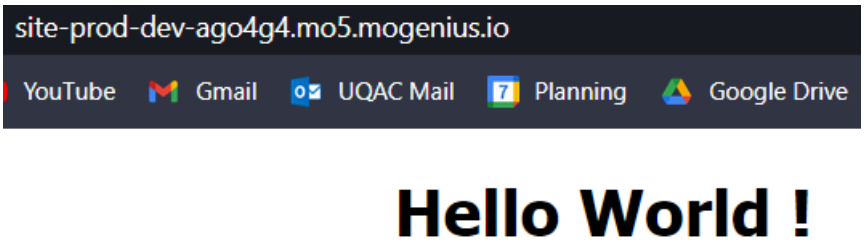
Le dépôt de l'api utilisé est disponible à l'adresse : https://github.com/Aeranduils/covid-api.

2. NGINX

De manière plus simpliste, il est possible d'utiliser les template de mogenius accessible à la fois en interne et en externe.

Pour ajouter un service :

https://site-prod-dev-ago4g4.mo5.mogenius.io/



Disponible sur le repository : https://github.com/Aeranduils/site

Avec un Dockerfile qui créer une image NGINX.

```
1  FROM nginxinc/nginx-unprivileged:stable-alpine
2
3  COPY html /usr/share/nginx/html
4
5  EXPOSE 8080
6
7  USER 101
8
9  CMD nginx -g 'daemon off;'
```

Figure 6: Dockerfile