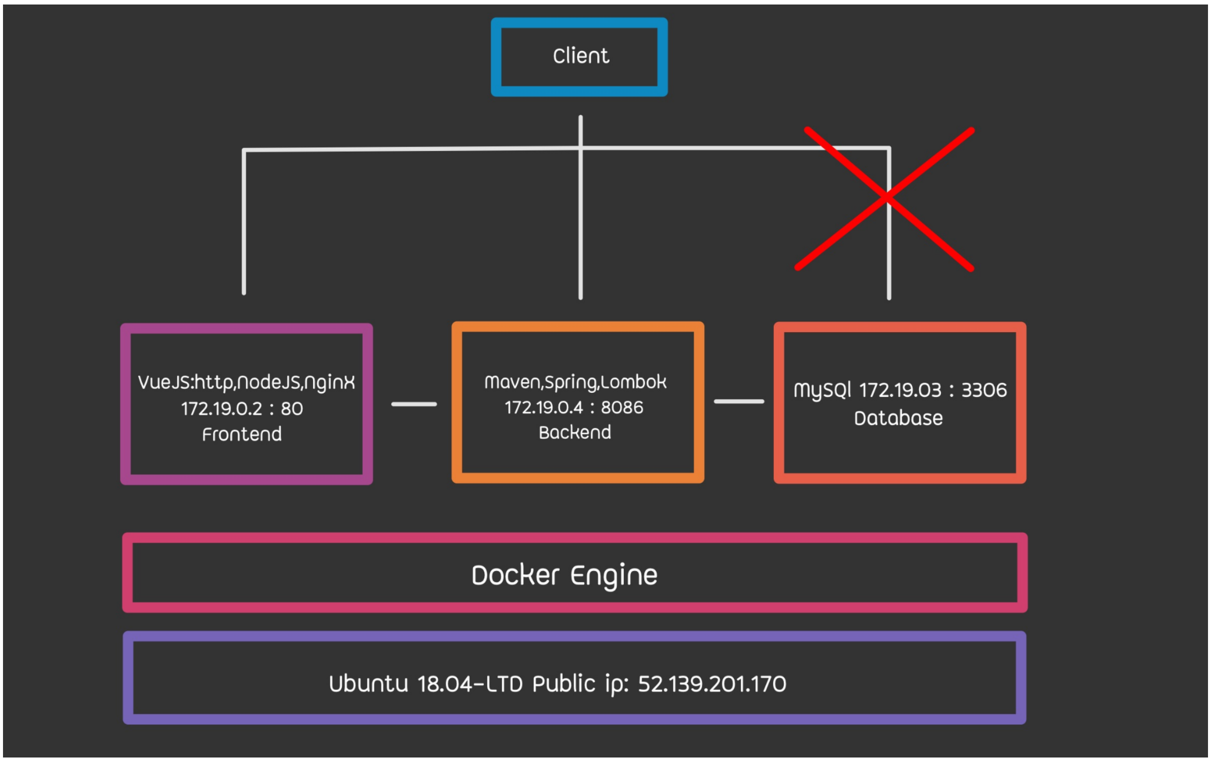
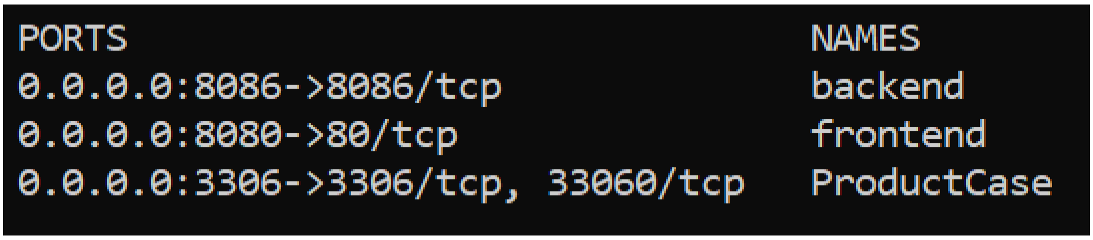
DevOps

1.

1.1) Architecture diagram

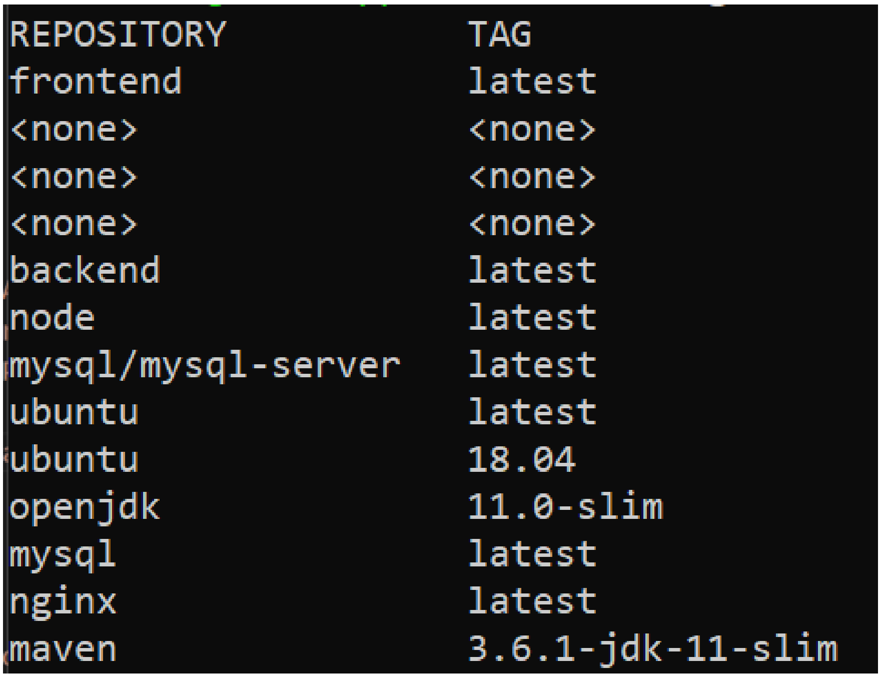


1.2) port design



```
PORTS                                   NAMES
0.0.0.0:8086->8086/tcp                  backend
0.0.0.0:8080->80/tcp                    frontend
0.0.0.0:3306->3306/tcp, 33060/tcp       ProductCase
```

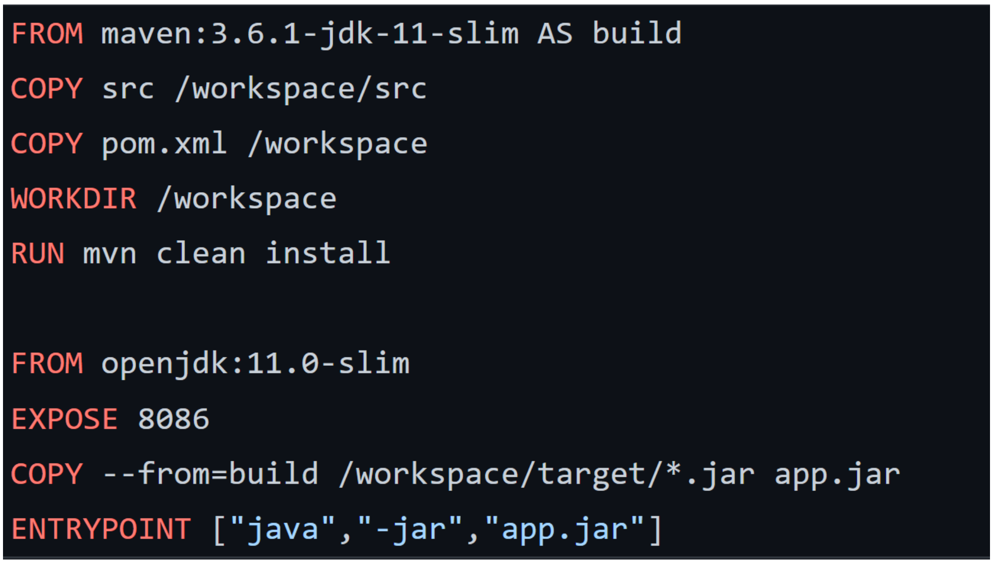
2 List of Docker images that you used with versions/tags



```
REPOSITORY           TAG
frontend             latest
<none>               <none>
<none>               <none>
<none>               <none>
backend              latest
node                 latest
mysql/mysql-server   latest
ubuntu               latest
ubuntu               18.04
openjdk              11.0-slim
mysql                latest
nginx                latest
maven                3.6.1-jdk-11-slim
```

3. Dockerfile

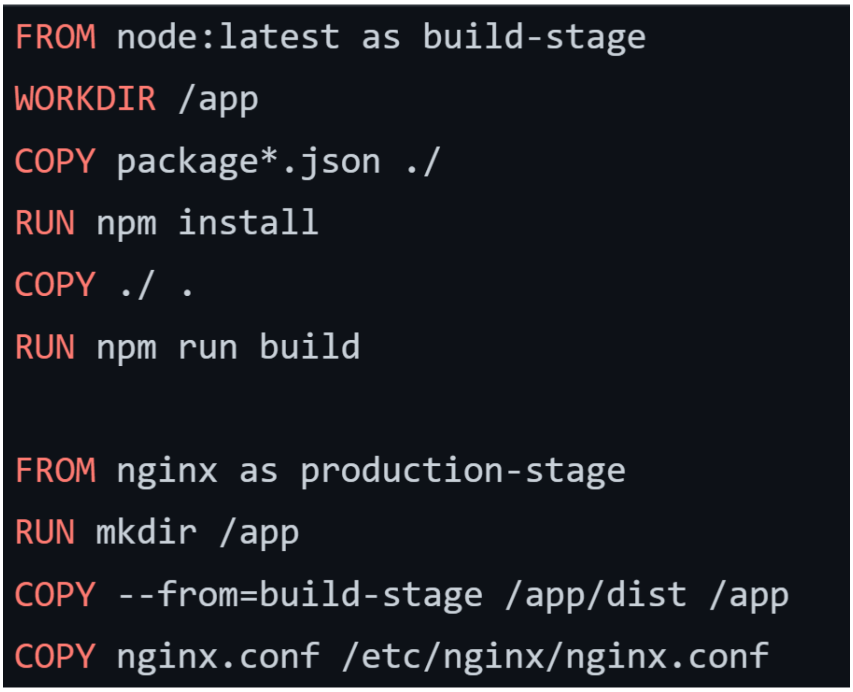
3.1) Backend



```
FROM maven:3.6.1-jdk-11-slim AS build
COPY src /workspace/src
COPY pom.xml /workspace
WORKDIR /workspace
RUN mvn clean install

FROM openjdk:11.0-slim
EXPOSE 8086
COPY --from=build /workspace/target/*.jar app.jar
ENTRYPOINT ["java","-jar","app.jar"]
```

3.2) Fontend

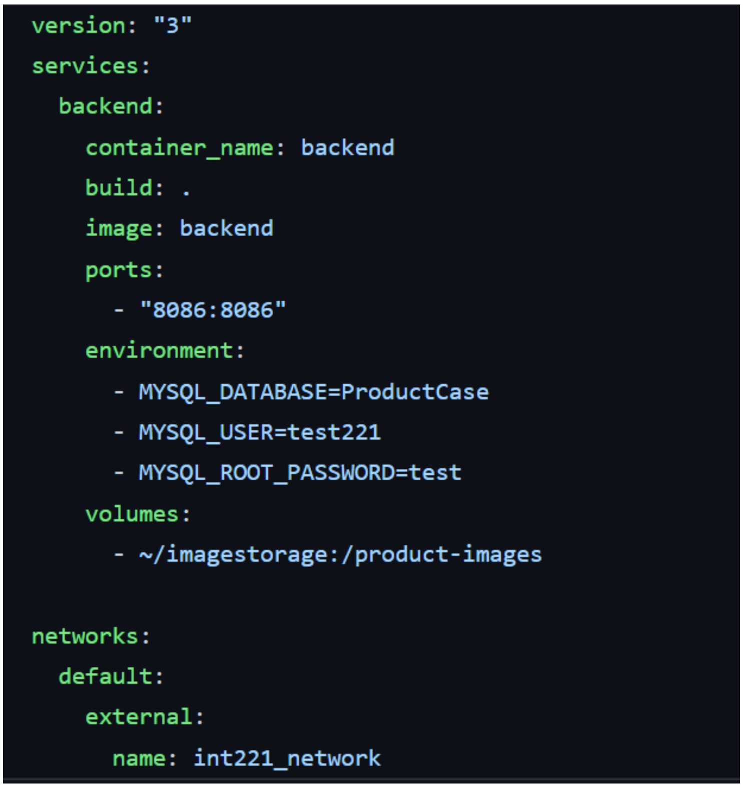


```
FROM node:latest as build-stage
WORKDIR /app
COPY package*.json ./
RUN npm install
COPY ./ .
RUN npm run build

FROM nginx as production-stage
RUN mkdir /app
COPY --from=build-stage /app/dist /app
COPY nginx.conf /etc/nginx/nginx.conf
```

4. Docker-compose.yml

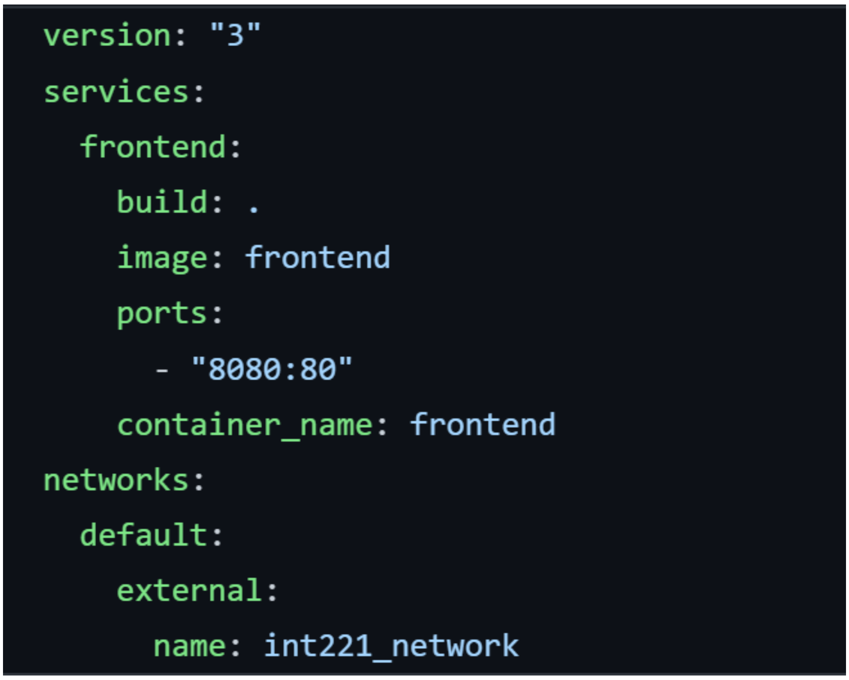
4.1) Backend



```
version: "3"
services:
  backend:
    container_name: backend
    build: .
    image: backend
    ports:
      - "8086:8086"
    environment:
      - MYSQL_DATABASE=ProductCase
      - MYSQL_USER=test221
      - MYSQL_ROOT_PASSWORD=test
    volumes:
      - ~/imagestorage:/product-images

networks:
  default:
    external:
      name: int221_network
```

4.2) Fontend



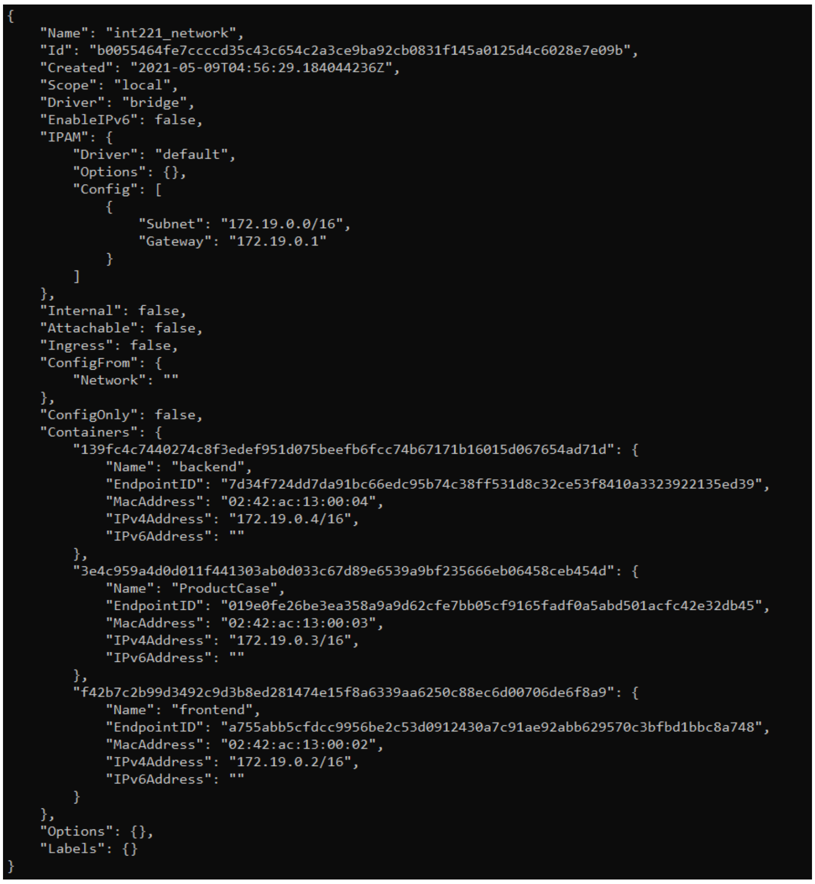
```
version: "3"
services:
  frontend:
    build: .
    image: frontend
    ports:
      - "8080:80"
    container_name: frontend
networks:
  default:
    external:
      name: int221_network
```

4.3) Database

```
version: '3'
services:
 database:
  build: .
  image: mysql
  container_name: ProductCase
  environment:
   MYSQL_DATABASE: ProductCase
   MYSQL_ROOT_PASSWORD: project221@SIT
  volumes:
    - ./my-db/datadir:/var/lib/mysql
    - ./config/my.cnf:/etc/my.cnf
    - ./config:/etc/my.cnf.dls
  ports:
   - "3306:3306"
```

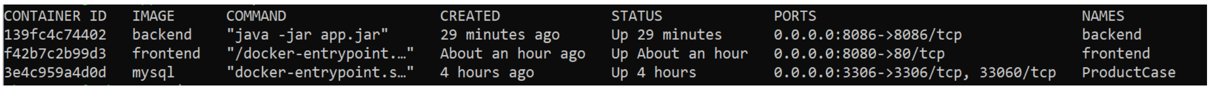
```
networks:
  default:
    external:
      name: int221_network
```

5. Docker network inspect



```
{
    "Name": "int221_network",
    "Id": "b0055464fe7ccccd35c43c654c2a3ce9ba92cb0831f145a0125d4c6028e7e09b",
    "Created": "2021-05-09T04:56:29.184044236Z",
    "Scope": "local",
    "Driver": "bridge",
    "EnableIPv6": false,
    "IPAM": {
        "Driver": "default",
        "Options": {},
        "Config": [
            {
                "Subnet": "172.19.0.0/16",
                "Gateway": "172.19.0.1"
            }
        ]
    },
    "Internal": false,
    "Attachable": false,
    "Ingress": false,
    "ConfigFrom": {
        "Network": ""
    },
    "ConfigOnly": false,
    "Containers": {
        "139fc4c7440274c8f3edef951d075beefb6fcc74b67171b16015d067654ad71d": {
            "Name": "backend",
            "EndpointID": "7d34f724dd7da91bc66edc95b74c38ff531d8c32ce53f8410a3323922135ed39",
            "MacAddress": "02:42:ac:13:00:04",
            "IPv4Address": "172.19.0.4/16",
            "IPv6Address": ""
        },
        "3e4c959a4d0d011f441303ab0d033c67d89e6539a9bf235666eb06458ceb454d": {
            "Name": "ProductCase",
            "EndpointID": "019e0fe26be3ea358a9a9d62cfe7bb05cf9165fadf0a5abd501acfc42e32db45",
            "MacAddress": "02:42:ac:13:00:03",
            "IPv4Address": "172.19.0.3/16",
            "IPv6Address": ""
        },
        "f42b7c2b99d3492c9d3b8ed281474e15f8a6339aa6250c88ec6d00706de6f8a9": {
            "Name": "frontend",
            "EndpointID": "a755abb5cfdcc9956be2c53d0912430a7c91ae92abb629570c3bfbd1bbc8a748",
            "MacAddress": "02:42:ac:13:00:02",
            "IPv4Address": "172.19.0.2/16",
            "IPv6Address": ""
        }
    },
    "Options": {},
    "Labels": {}
}
```

6. Docker containers list



```
CONTAINER ID   IMAGE      COMMAND                  CREATED              STATUS              PORTS                               NAMES
139fc4c74402   backend    "java -jar app.jar"      29 minutes ago       Up 29 minutes       0.0.0.0:8086->8086/tcp              backend
f42b7c2b99d3   frontend   "/docker-entrypoint.…"   About an hour ago    Up About an hour    0.0.0.0:8080->80/tcp                frontend
3e4c959a4d0d   mysql      "docker-entrypoint.s…"   4 hours ago          Up 4 hours          0.0.0.0:3306->3306/tcp, 33060/tcp   ProductCase
```

7. Describe configurations that you did to set the environment for FE,BE,proxy

7.1) Frontend : สำหรับการทำ Dockerfile ใช้ node image และ nginx image สร้าง image ใน frontend และทำการ copy environment เพื่อ build image โดยผ่านการใช้ port 80

ส่วน docker-compose.yml build container name frontend and image frontend และผ่าน docker network int221_network

7.2) Database : สำหรับ docker-compose.yml จะ build container name ProductCase ,images mysql และ set mysql root password โดยใช้ port 3306 docker network int221_network

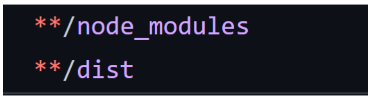
7.3) Backend : Dockerfile ใช้ maven และ openjdk build image กับ .jar file ใช้ port 8086

ส่วน docker-compose.yml build container backend และ name backend และ build environment กับ database connection และ set port 8086 docker network int221_network

8. Other files that you use with explanation of what you did

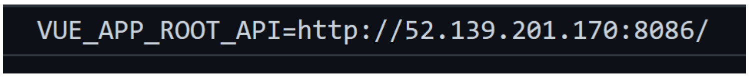
#Frontend

.dockerignore



```
**/node_modules
**/dist
```

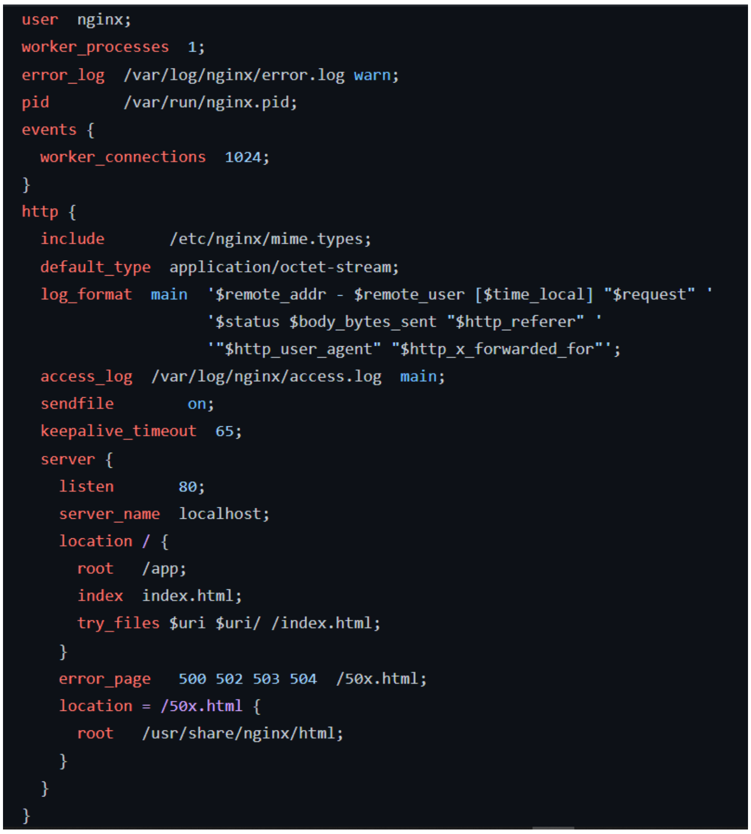
.env



```
VUE_APP_ROOT_API=http://52.139.201.170:8086/
```

set api to backend

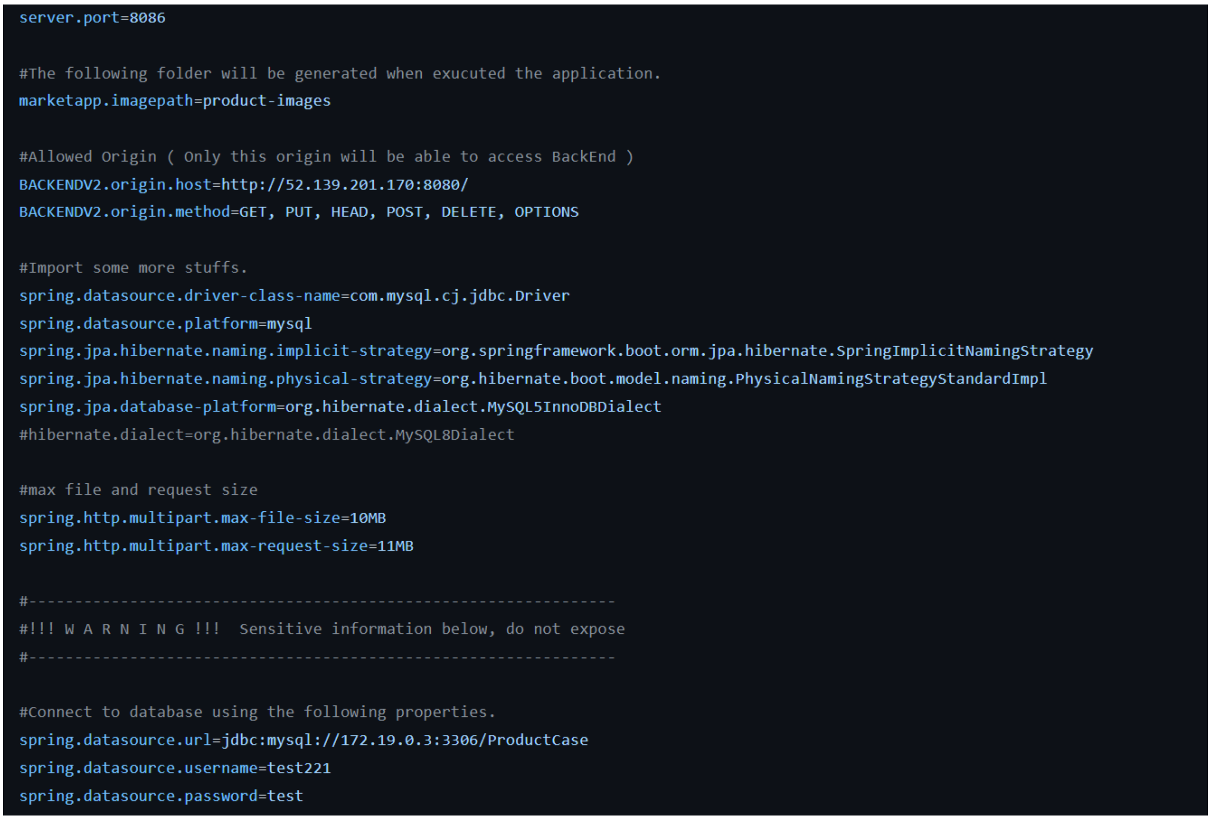
Nginx.conf



```
user  nginx;
worker_processes  1;
error_log  /var/log/nginx/error.log warn;
pid        /var/run/nginx.pid;
events {
  worker_connections  1024;
}
http {
  include       /etc/nginx/mime.types;
  default_type  application/octet-stream;
  log_format  main  '$remote_addr - $remote_user [$time_local] "$request" '
                    '$status $body_bytes_sent "$http_referer" '
                    '"$http_user_agent" "$http_x_forwarded_for"';
  access_log  /var/log/nginx/access.log  main;
  sendfile        on;
  keepalive_timeout  65;
  server {
    listen       80;
    server_name  localhost;
    location / {
      root   /app;
      index  index.html;
      try_files $uri $uri/ /index.html;
    }
    error_page   500 502 503 504  /50x.html;
    location = /50x.html {
      root   /usr/share/nginx/html;
    }
  }
}
```

#Backend

Application.properties



```
server.port=8086

#The following folder will be generated when exucuted the application.
marketapp.imagepath=product-images

#Allowed Origin ( Only this origin will be able to access BackEnd )
BACKENDV2.origin.host=http://52.139.201.170:8080/
BACKENDV2.origin.method=GET, PUT, HEAD, POST, DELETE, OPTIONS

#Import some more stuffs.
spring.datasource.driver-class-name=com.mysql.cj.jdbc.Driver
spring.datasource.platform=mysql
spring.jpa.hibernate.naming.implicit-strategy=org.springframework.boot.orm.jpa.hibernate.SpringImplicitNamingStrategy
spring.jpa.hibernate.naming.physical-strategy=org.hibernate.boot.model.naming.PhysicalNamingStrategyStandardImpl
spring.jpa.database-platform=org.hibernate.dialect.MySQL5InnoDBDialect
#hibernate.dialect=org.hibernate.dialect.MySQL8Dialect

#max file and request size
spring.http.multipart.max-file-size=10MB
spring.http.multipart.max-request-size=11MB

#----------------------------------------------------------------
#!!! W A R N I N G !!!  Sensitive information below, do not expose
#----------------------------------------------------------------

#Connect to database using the following properties.
spring.datasource.url=jdbc:mysql://172.19.0.3:3306/ProductCase
spring.datasource.username=test221
spring.datasource.password=test
```