서버 설정 가이드

1. 실행 환경 설정

- Ubuntu 20.04 LTS

Git 설치

```
sudo apt-get update
sudo apt-get install git
```

- git 설치 후 clone 진행

로컬 환경변수 설정

- FE/moomuweb/.env

```
# 네이버맵 api 키
REACT_APP_API_KEY_ID=
REACT_APP_API_KEY=
```

- BE/moomu/.env

```
PROJECT_NAME=moomu
BACKEND_CORS_ORIGINS=["http://localhost:8000", "http://localhost:3000", "https://k7b202.p.ssafy.io", "http://localhost:19006"]

DB_USERNAME = 'root'
DB_PASSWORD = {루트 패스워드}
DB_HOST = 'localhost'
DB_PORT = '3306'
DB_SCHEMA = 'moomu'
ACCESS_KEY = 'b202'
REFRESH_KEY = 'b202r'

# NaverMapAPI
URL = 'https://naveropenapi.apigw.ntruss.com/map-direction-15/v1/driving'
CLINET_ID =
CLIENT_SECRET =
```

Docker 설치

```
# docker 설치 스크립트를 wget으로 가져와서 실행
sudo wget -qO- https://get.docker.com/ | sh

# 설치 확인
docker -v
docker compose version
```

DB 구동

- mysql
  - 컨테이너 실행 후 moomu 스키마 생성

```
# mysql 이미지 가져오기
docker pull mysql

# 컨테이너 실행
docker run --name {컨테이너 이름} -e MYSQL_ROOT_PASSWORD={루트 패스워드} -d -p 3306:3306 mysql
```

- redis

```
# redis 이미지 가져오기
docker pull redis

# 컨테이너 실행
docker run --name {컨테이너 이름} -d -p 6379:6379 redis
```

2. 빌드 파일 작성

- git clone 커맨드를 실행한 디렉토리에서 진행.

Nginx

- nginx.conf

```
user nginx;
worker_processes auto;
events {
  worker_connections 1024;
}
http{
  include mime.types;

  access_log /var/log/nginx/access.log;
  error_log /var/log/nginx/error.log;

  server {
    listen 80;
    listen [::]:80;

    location / {
      root  /home/build;
      index index.html index.htm;
      try_files   $uri $uri/ /index.html;
    }
    location /api/{
      proxy_pass http://172.17.0.1:8000/;
    }
  }
}
```

Dockerfile

- fastapiDF

```
FROM python:3.10

WORKDIR /code
COPY ./S07P31B202/BE/moomu /code
RUN pip install --no-cache-dir --upgrade -r /code/requirements.txt

CMD ["uvicorn", "app.main:app", "--host", "0.0.0.0", "--port", "8000"]
```

- socketDF

```
FROM python:3.10

WORKDIR /code
COPY ./S07P31B202/SOCKET /code
RUN pip install --no-cache-dir --upgrade -r /code/requirements.txt

CMD ["uvicorn", "app.main:app", "--host", "0.0.0.0", "--port", "8000"]
```

- nginxDF

```
FROM node:16.17.0 as builder
COPY ./S07P31B202/FE/moomuweb /home/react
WORKDIR /home/react
RUN npm install
RUN npm run build
```

```
FROM nginx
# nginx 설정 복사
COPY nginx.conf /etc/nginx
# 빌드 파일 복사
COPY --from=builder /home/react/build /home/build
# 인증서 복사
COPY ./data/certbot/conf /etc/letsencrypt
EXPOSE 80
CMD ["nginx", "-g", "daemon off;"]
```

Docker-compose

- docker-compose.yaml

```
version: '3'
services:
  fastapi:
    build:
      context: .
      dockerfile: fastapiDF
    ports:
      - 8000:8000
    container_name: fastapi
    extra_hosts:
      - localhost:host-gateway
  socket:
    build:
      context: .
      dockerfile: socketDF
    ports:
      - 9000:8000
    container_name: socket
    extra_hosts:
      - localhost:host-gateway
  nginx:
    build:
      context: .
      dockerfile: nginxDF
    ports:
      - 80:80
    container_name: nginx
```

3. 배포

```
# 컨테이너 실행
sudo docker compose up -d --build

# 컨테이너 5개 구동중인 것 확인
sudo docker ps
```

```
CONTAINER ID   IMAGE          COMMAND                  CREATED        STATUS        PORTS                                              NAMES
392eb3826899   proj-nginx     "/docker-entrypoint.…"   25 hours ago   Up 25 hours   0.0.0.0:80->80/tcp, :::80->80/tcp                  nginx
c39916778abb   proj-fastapi   "uvicorn app.main:ap…"   28 hours ago   Up 28 hours   0.0.0.0:8000->8000/tcp, :::8000->8000/tcp          fastapi
d7bf8bc61ac4   proj-socket    "uvicorn app.main:ap…"   45 hours ago   Up 45 hours   0.0.0.0:9000->8000/tcp, :::9000->8000/tcp          socket
c04416f36b6e   redis          "docker-entrypoint.s…"   3 weeks ago    Up 3 weeks    0.0.0.0:6379->6379/tcp, :::6379->6379/tcp          redis
68c224150eb7   mysql          "docker-entrypoint.s…"   3 weeks ago    Up 3 weeks    0.0.0.0:3306->3306/tcp, :::3306->3306/tcp          mysql
```