

**ArtHorizon, 미술의 지평선**

**포팅 매뉴얼**

**목차**

**1. Gitlab 소스코드 클론 이후 빌드 및 배포 과정 정리**

**A. 개발환경**

**Frontend**

**Backend**

**AI**

**배포**

**IDE**

**B. 배포 시 특이사항**

**Nginx**

**C. 빌드 & 배포**

**I. Frontend**

**II. Backend**

**III. AI**

**IV. DB**

**2. 프로젝트에서 사용하는 외부 서비스 정보**

**1. Gitlab 소스코드 클론 이후 빌드 및 배포 과정 정리**

**A. 개발환경**

**Frontend**

* HTML / CSS
* JavaScript ES6+
* Node.js v16.17.0
* React 18.2.0
* React-Router 6.4.0
* Redux-Toolkit

**Backend**

* Java SDK 1.8.0\_192
* Spring boot 2.7.3
* MySQL 8.0

**AI**

* FastAPI 0.85.0
* Python 3.10.7
* Pytorch 1.12.1

**배포**

* AWS EC2
* Ubuntu
* Nginx

**IDE**

* VSCode
* Intellij IDEA 2022.1.3
* JupyterHub

**B. 배포 시 특이사항**

**Nginx**

/frontend/nginx/default.conf

server {

listen 80 default\_server;

listen [::]:80 default\_server;

index index.html;

server\_name j7d201.p.ssafy.io;

proxy\_set\_header X-Real-IP $remote\_addr;

proxy\_set\_header X-Forwarded-For $proxy\_add\_x\_forwarded\_for;

proxy\_set\_header Host $host;

location / {

root /usr/share/nginx/html;

try\_files $uri $uri/ /index.html;

}

// Springboot

location /api {

proxy\_pass http://j7d201.p.ssafy.io:8081;

}

location /images {

alias /home/ubuntu/S07P22D201/frontend/docker-volume/images

}

// FastAPI

location /medici {

proxy\_pass http://j7d201.p.ssafy.io:8000;

}

}

**C. 빌드 & 배포**

• git clone https://lab.ssafy.com/s07-ai-image-sub2/S07P22D201.git

**I. Frontend**

frontend/Dockerfile

FROM node:lts-alpine as build-stage

WORKDIR /usr/src/app

COPY package\*.json ./

RUN npm install --silent

COPY . /usr/src/app

RUN npm run build

FROM nginx:stable-alpine as production-stage

RUN rm /etc/nginx/conf.d/default.conf

COPY ./nginx/default.conf /etc/nginx/conf.d/

RUN rm -rf /usr/share/nginx/html/\*

COPY --from=build-stage /usr/src/app/build /usr/share/nginx/html

EXPOSE 80

CMD ["nginx", "-g", "daemon off;"]

**Docker image 생성 및 배포**

sudo docker build –t frontimg .

sudo docker run -d --name frontend -p 80:80 -v

/usr/share/nginx/html/images:/home/ubuntu/S07P22D201/frontend/docker-volume/images frontimg

**II. Backend**

application.properties

1. DB 접속 설정

spring.datasource.url=jdbc:mysql://j7d201.p.ssafy.io:3308/ssafyproject2?characterEncoding=UTF-8&serverTimezone=UTC  
spring.datasource.username=PJT2D201  
spring.datasource.password=d201horizon!

2. JWT secret key 설정

secret.jwt.key="team\_medici"

3. 서버 포트 설정

server.port = 8081

4. spring boot 카멜케이스(camelCase) 스네이크 케이스(snake\_case)로 자동 변환 금지 설정

spring.jpa.hibernate.naming.implicit-strategy=org.hibernate.boot.model.naming.ImplicitNamingStrategyLegacyJpaImpl  
spring.jpa.hibernate.naming.physical-strategy=org.hibernate.boot.model.naming.PhysicalNamingStrategyStandardImpl

back/Dockerfile

FROM openjdk:8-jdk

ARG JAR\_FILE=build/libs/arthorizon-0.0.1-SNAPSHOT.jar

COPY ${JAR\_FILE} app.jar

ENTRYPOINT ["java","-jar","/app.jar"]

**빌드**

./gradlew bootJar

**Docker image 생성 및 배포**

sudo docker build –t backimg .

sudo docker run -d --name back -p 8081:8081 –v

/usr/share/nginx/html/images:/home/ubuntu/S07P22D201/back/docker-volume/images backimg

**III. FastAPI**

AIpjt/Dockerfile

FROM python:3.10

WORKDIR /prod

COPY ./require.txt /prod/requirements.txt

RUN pip install --no-cache-dir --upgrade -r /prod/requirements.txt

COPY ./app /prod/app

RUN apt-get update

RUN apt-get -y install libgl1-mesa-glx

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

**Docker image 생성 및 배포**

sudo docker build –t fastimg .

sudo docker run –d --name fast –p 8000:8000 fastimg

**IV. DB**

**MobaXterm을 이용하여 ubuntu에 mysql 설치**

apt-get update

sudo apt-get install mysql-server

**루트 계정 접속 및 사용자 계정 권한 부여**

/usr/bin/mysql –u root –p

Mysql > create user ‘root’@’%’ identified by ‘[password]’;

Mysql > grant all privileges on \*.\* to ‘root’@’%’ with grant option;

**실행 상태 확인**

Systemctl status mysql #mysql 상태확인

Sudo systemctl start mysql #mysql 시작

Systemctl stop mysql #mysql 실행중지

Netstat –tnlp #포트확인

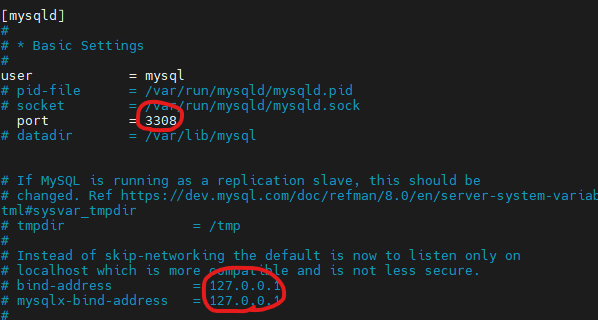
**포트를 열어준다**

Cd /etc/mysql/mysql.conf.d #절대경로로 설정파일 접근

Vi mysqld.cnf #설정파일 편집모드

Port 번호는 3306으로 그대로 둘 경우 충돌이 날 수 있으므로 변경하기를 권장한다.

Bind-address나 mysqlx-bind-address는 사용되지 않도록 주석 처리한다.



**리눅스 편집 명령어**

i로 입력모드

esc로 모드 탈출

:wq!로 강제 저장, 이러지 않으려면 sudo vi로 열어야 함

**2. 프로젝트에서 사용하는 외부 서비스 정보**

**A. CLIP(Contrastive Language-Image Pre-Training)**

이미지와 텍스트를 연관시켜 학습시킨 AI모델

URL : <https://github.com/openai/CLIP>

**B. Kakaopay API – 카카오페이 API**

URL : https://developers.kakao.com/docs/latest/ko/kakaopay/common