

포팅 메뉴얼

1. spring-boot jar 파일 생성

```
gradlew build
```

2. 배포에 필요한 요소들 설치

- docker

```
//오래된 버전 삭제
sudo apt-get remove docker docker-engine docker.io containerd runc

//apt package index 업데이트
sudo apt-get update

//repository 이용 위한 package 설치
sudo apt-get install \
    ca-certificates \
    curl \
    gnupg \
    lsb-release

//도커 GPG Key 등록
curl -fsSL https://download.docker.com/linux/ubuntu/gpg | sudo gpg --dearmor -o /usr/share/keyrings/docker-archive-keyring.gpg

//stable repository 등록
echo \
    "deb [arch=amd64 signed-by=/usr/share/keyrings/docker-archive-keyring.gpg] https://download.docker.com/linux/ubuntu \
    $(lsb_release -cs) stable" | sudo tee /etc/apt/sources.list.d/docker.list > /dev/null

//도커 최신 버전 설치
sudo apt-get update
sudo apt-get install docker-ce docker-ce-cli containerd.io

//도커 버전 확인
docker --version
```

- mysql

```
//버전을 입력하지 않으면 자동으로 최신 버전을 다운로드 해준다
docker pull mysql
```

- unzip

```
apt-get install unzip
```

- certbot

```
sudo apt-get install certbot python3-certbot-nginx
```

3. Docker File 생성

```
FROM openjdk:8-jdk-alpine
ARG JAR_FILE=target/*.jar
COPY [jar파일 이름] app.jar
ENTRYPOINT ["java", "-jar", "/app.jar"]

# 컨테이너 내 작업 경로
WORKDIR [경로]
```

4. 컨테이너 생성 및 실행

- docker network & docker volume

```
//네트워크 생성 -- 컨테이너끼리 엮기 위함
docker network create [네트워크 이름]

//생성된 네트워크 목록
docker network ls

//도커 볼륨 생성 -- 도커의 폴더와 우분투의 폴더를 엮기 위함
docker volume create [volume_name]

//생성된 볼륨 목록
docker volume ls
```

- mysql 컨테이너 생성 및 실행

```
docker run --name [db 이름] -p 3306:3306 --network [network 이름] -e MYSQL_ROOT_PASSWORD=[비밀번호] -e MYSQL_DATABASE=[db 커넥션 이름] -
```

- 이후 db 포트에 접속하여 스키마 생성

- spring boot 컨테이너 생성 및 실행

```
//docker file과 jar파일이 있는 폴더 내에서
//spring boot 이미지 생성
docker build -t [이미지 이름] .

//spring boot 컨테이너 생성 및 실행 + 볼륨으로 폴더 엮기
docker run -v [엮을 폴더 경로]:[엮을 폴더 경로] -d -p [포트번호]:[포트번호] [실행할 이미지 이름] --network [network 이름] --name [생성할 컨테이너 이
```

5. nginx

- 설정 파일 수정

```
server {
    .
    .
    .
    location /api {
        proxy_pass http://localhost:[spring 포트 번호]/api;
    }

    location / {
        root [dist 파일 위치];
        index index.html index.html;
        try_files $uri $uri/ /index.html;
    }
    .
    .
    .
}
```