

포팅 매뉴얼

프로젝트 기술 스택

☐ Frontend

- React 18.2.0
- React Router 6.3.0
- Recoil 0.7.5
- Material UI 5.10.3
- Axios 0.27.2
- Typescript 4.8.2
- sweetalert2 11.4.37

☐ Backend

- openjdk 1.8.0_192
- Java Zulu 8.33.0.1
- eclipse 4.16.0 (2020-06)
- Spring Boot 2.7.1
- Spring Data JPA
- Spring Security 5.6.6
- MySQL 5.7.37

☐ CI/CD

- Docker version 20.10.18, build b40c2f6
- NGINX
- AWS EC2
- Jenkins : jenkins/jenkins:lts (Docker image)

BE 빌드

BE Dockerfile

```
FROM openjdk:8
EXPOSE 8080
ARG JAR_FILE=build/libs/farming-0.0.1-SNAPSHOT.jar
COPY ${JAR_FILE} app.jar
ENTRYPOINT ["java", "-jar", "/app.jar"]
ENV TZ=Asia/Seoul
RUN apt-get install -y tzdata
```

FE 빌드

FE Dockerfile

```
FROM nginx
EXPOSE 3000

# root 에 app 폴더를 생성
RUN mkdir /app
# work dir 고정
WORKDIR /app
# work dir 에 build 폴더 생성 /app/build
RUN mkdir ./build
# host pc의 현재경로의 build 폴더를 workdir 의 build 폴더로 복사
ADD ./build ./build
# nginx 의 default.conf 를 삭제
RUN rm /etc/nginx/conf.d/default.conf
# host pc 의 nginx.conf 를 아래 경로에 복사
COPY ./nginx.conf /etc/nginx/conf.d

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

nginx.conf

```
server {
    listen 80;
    location / {
        root    /app/build;
        index   index.html;
        try_files $uri $uri/ /index.html;
    }
}
```

배포

원활한 서비스 구동을 위해 배포해야 하는 도커 컨테이너는

- MySQL
- BE
- FE
- Jenkins

총 4개입니다.

각 컨테이너를 pull하고 실행하기에 앞서 BE와 MySQL 컨테이너를 원활하게 연결해주기 위해서 커스텀 네트워크를 만들어서 적용시켜줘야 합니다.

```
docker network create farminging
```

Nginx 설정과 ssl 인증서 발급 및 적용

```
# 설치
sudo apt-get install nginx
```

letsencrypt 설치를 위해 다음과 같은 순서로 명령어를 입력합니다.

```
sudo apt-get install letsencrypt
```

```
sudo systemctl stop nginx
```

```
sudo letsencrypt certonly --standalone -d www제외한 도메인 이름
```

이렇게 한 후, "Congratulations!"로 시작하는 문구가 보이면, 인증서 발급이 완료된 것입니다.

인증서 발급 후, /etc/nginx/sites-available로 이동한 후, 적절한 이름의 파일을 생성하여 다음과 같이 작성합니다.

```
server {
    location /{
        proxy_pass http://j7a608.p.ssafy.io:3000/;
        proxy_set_header Host $http_host;
        proxy_set_header X-Real-IP $remote_addr;
        proxy_set_header X-Forwarded-For $proxy_add_x_forwarded_for;
        add_header Content-Security-Policy "upgrade-insecure-requests";
    }
    location /api {
        proxy_pass http://j7a608.p.ssafy.io:8080/api;
        proxy_set_header Host $http_host;
        proxy_set_header X-Real-IP $remote_addr;
        proxy_set_header X-Forwarded-For $proxy_add_x_forwarded_for;
        add_header Content-Security-Policy "upgrade-insecure-requests";
    }

    listen 443 ssl; # managed by Certbot
    ssl_certificate /etc/letsencrypt/live/j7a608.p.ssafy.io/fullchain.pem; # managed by Certbot
    ssl_certificate_key /etc/letsencrypt/live/j7a608.p.ssafy.io/privkey.pem; # managed by Certbot
    # include /etc/letsencrypt/options-ssl-nginx.conf; # managed by Certbot
    # ssl_dhparam /etc/letsencrypt/ssl-dhparams.pem; # managed by Certbot
}

server {
    listen 80;

    location / {
        return 301 https://j7a608.p.ssafy.io$request_uri;
    }
}
```

그 후에 차례로 명령을 실행합니다.

```
sudo ln -s /etc/nginx/sites-available/default /etc/nginx/sites-enabled/default
```

```
# 다음 명령어에서 successful이 뜨면 nginx를 실행할 수 있다.
```

```
sudo nginx -t
```

```
sudo systemctl restart nginx
```

이렇게 실행하면, http로 80포트 접근시, 443 포트(https)로 리다이렉트 됩니다. 그리고 백엔드 url을 /api/**로 분기처리할 수 있습니다.

```
https://도메인주소 로 접근하면 배포한 웹 페이지에 접속할 수 있게됩니다.
```

MySQL

```
docker pull mysql:5.7.37
```

```
docker run -d --name mysql -p 3306:3306 -v mysql-volume:/var/lib/mysql --network farming -e MYSQL_ROOT_PASSWORD=ssafy mysql:5.7.37
```

아래와 같은 명령어로 mysql bash를 실행시켜 서버의 MySQL 도커 이미지에 접근해 컨트롤할 수 있습니다.

```
docker exec -it -e LC_ALL=C.UTF-8 mysql bash

cd /etc/mysql/mysql.conf.d
vim mysqld.cnf
```

mysqld.cnf에 아래 내용 추가해서 한글 입출력 설정

```
[client]
default-character-set = utf8mb4

[mysqld]
default-character-set = utf8mb4
character-set-server = utf8mb4
collation-server = utf8_general_ci
init-connect = SET NAMES utf8

[mysqldump]
default-character-set = utf8mb4

[mysql]
default-character-set = utf8mb4
```

mysql bash에서 exit한 뒤

```
docker container restart mysql
```

```
docker exec -it -e LC_ALL=C.UTF-8 mysql bash

mysql -u A301 -p

[비밀번호 입력]

[DDL, DML, etc. 입력]
```