

# 빌드 및 배포

---

프로젝트 기간 : 2022:07.04 ~ 8.19

삼성SW청년아카데미 광주캠퍼스 7기  
김도원 김강현 권덕민 윤성한 유승우 정혜원

## 1. 기술스택

구분	기술스택	상세내용	버전
공통	형상관리	gitlab	-
	이슈관리	Jira	-
	커뮤니케이션	Mattermost, Notion	-
BackEnd	DB	MySQL	5.8
		JPA	-
	Java	Zulu	-
	Spring	Spring Boot	2.7.2
	IDE	IntelliJ	2022.2
	Build	Gradle	7.4.1
	WebRTC	Tomcat	9.0.64
FrontEnd	HTML5		-
	CSS3		-
	JavaScript(ES6)		-
	IDE	Visual Studio Code	1.63.2
	mui/material		5.9.1
	tanstack/react-query		4.0.10
	axios		0.27.2
	firebase		9.9.2
	formik		2.2.9
	kakao		0.4.1
	react-kakao-maps-sdk		1.1.1
	react-qr-reader		2.2.1
	recoil		0.7.4
	recoil-persist		4.2.0
	sass		1.54.1
	react-router-dom		6.3.0
	styled-components		5.3.5
	yup		0.32.11
Server	서버	AWS EC2	-
	플랫폼	Ubuntu	20.04.4 LTS
	배포	Docker	20.10.17
		Jenkins	2.364.2

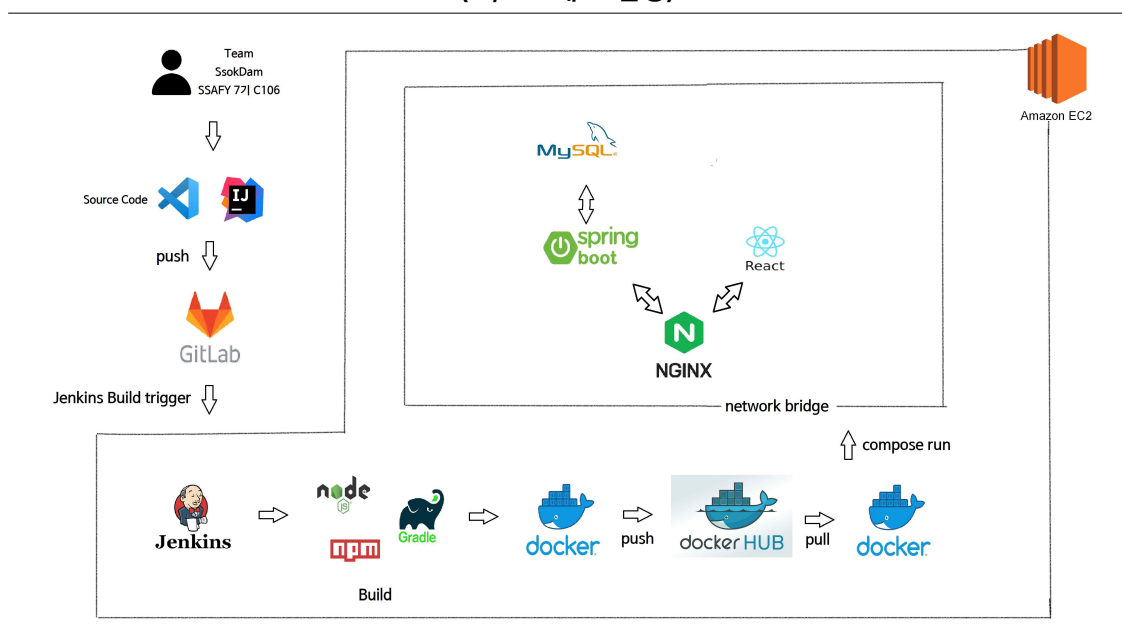
Embedded	ESP32	GPS Sensor	TinyGPSPlus 1.03
		AirQuailty Sensor	
		WiFi	
		Temp Sensor	
		Gas Sensor	
		CRUD	Arduino JSON 0.10
		Servo motor	ESP Servo 1.0

## 2. 상세 내용

### □ 개요

아래 그림은 “ssokdam” 서비스의 배포환경 및 CI/CD 배포 흐름도입니다. 팀원들이 각자 작성한 프로젝트를 GitLab에 push 하면, Jenkins Build Trigger를 통해 FrontEnd와 BackEnd를 빌드하게 됩니다.

#### 〈CI/CD 배포 환경〉



각 프로젝트를 빌드 후에는 Docker 이미지를 만들고 이를 Docker Hub에 push 합니다. Docker Hub로부터 서비스에 필요한 이미지를 받아와 컨테이너로 띄웁니다. SSAFY에서 지원받은 EC2 싱글 인스턴스로 인프라를 구축했습니다. 추후 서비스화를 위해 Nginx는 리버스 프록시 서버로 설정하였습니다. 동시에 Nginx 서버로부터 9090 포트를 BackEnd 서버로 설정하여 구축했습니다.

## □ FrontEnd

- Docker 이미지 생성을 위해 Dockerfile을 작성합니다.  
(해당 파일은 프로젝트 내에 이미 작성되어있습니다.)

```
FROM node:alpine
WORKDIR "/Frontend/usr/src/app"
COPY package.json ./
RUN npm install --save --legacy-peer-deps
COPY ./ ./
CMD ["npm","run","start"]
```

- docker 이미지 생성

```
docker build -t syou1123/react:0.1 .
```

## □ Backend

- Gradle 빌드

```
./gradlew build
```

- Docker 이미지 생성을 위해 Dockerfile을 작성합니다.  
(해당 파일은 프로젝트 내에 이미 작성되어있습니다.)

```
FROM openjdk:17-jdk-alpine
ADD /build/libs/project-0.0.1-SNAPSHOT.jar app.jar
ENV JAVA_OPTS=""
ENTRYPOINT ["java","-jar","/app.jar"]
```

- docker 이미지 생성

```
docker build -t syou1123/springboot:0.1 .
```

## □ Docker-compose

- Docker 이미지 생성을 위해 Dockerfile을 작성합니다.  
(해당 파일은 프로젝트 내에 이미 작성되어있습니다.)

```
version: '3'
services:
  nginx:
    image: nginx:1.21.5-alpine
    ports:
      - 80:80
      - 443:443
    volumes:
      - /etc/nginx/nginx.conf:/etc/nginx/nginx.conf
      - /var/log/nginx:/var/log/nginx
      - /etc/letsencrypt:/etc/letsencrypt
      - /etc/localtime:/etc/localtime:ro -e TZ=Asia/Seoul
      - /etc/localtime:/etc/localtime:ro
      - /usr/share/zoneinfo/Asia/Seoul:/etc/timezone:ro
    container_name: myweb-proxy
    depends_on:
      - web
      - api
  web:
    image: swyou1123/react:0.1
    ports:
      - 3000:3000
    volumes:
      - /etc/localtime:/etc/localtime:ro -e TZ=Asia/Seoul
      - /etc/localtime:/etc/localtime:ro
      - /usr/share/zoneinfo/Asia/Seoul:/etc/timezone:ro
    container_name: myweb-web
**** 중략 ****
```

### 3. 프로퍼티 정의

#### □ MySQL

##### ○ 계정 생성

```
create user 'swyou'@'%' identified by 'qudrlxksdir1!';  
grant all privileges on *.* to 'swyou'@'%' with grant option; flush  
privileges
```

##### ○ 스키마 생성

```
create database if not exists project collate utf8mb4_general_ci;
```

#### □ Nginx

##### ○ 환경 설정

```
server{  
    server_name www.ssokdam.com ssokdam.com;  
    location / {  
        proxy_set_header HOST $host;  
        proxy_pass http://3.36.78.244:3000$request_uri;  
        proxy_redirect off;  
        proxy_buffer_size 128k;  
        proxy_buffers 4 256k;  
        proxy_busy_buffers_size 256k;  
    }  
    location /api {  
        proxy_set_header HOST $host;  
        proxy_pass http://3.36.78.244:9090$request_uri;  
        proxy_redirect off;  
        proxy_buffer_size 128k;  
        proxy_buffers 4 256k;  
        proxy_busy_buffers_size 256k;  
    }  
}  
  
listen [::]:443 ssl ipv6only=on; # managed by Certbot  
listen 443 ssl; # managed by Certbot  
ssl_certificate /etc/letsencrypt/live/ssokdam.com/fullchain.pem; # managed by Certbot  
ssl_certificate_key /etc/letsencrypt/live/ssokdam.com/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 {  
    if ($host = www.ssokdam.com) {  
        return 301 http://$host$request_uri;  
    } # managed by Certbot  
  
    if ($host = ssokdam.com) {  
        return 301 http://$host$request_uri;  
    } # managed by Certbot  
  
    listen 80 ;  
    listen [::]:80 ;  
    server_name www.ssokdam.com ssokdam.com;  
    return 404; # managed by Certbot  
}
```