

빌드 및 배포

프로젝트 기간 : 2022:08.22 ~ 10.07

삼성SW청년아카데미 서울캠퍼스 7기
권덕민 유승우 정관철 장우주 장한울 최영현

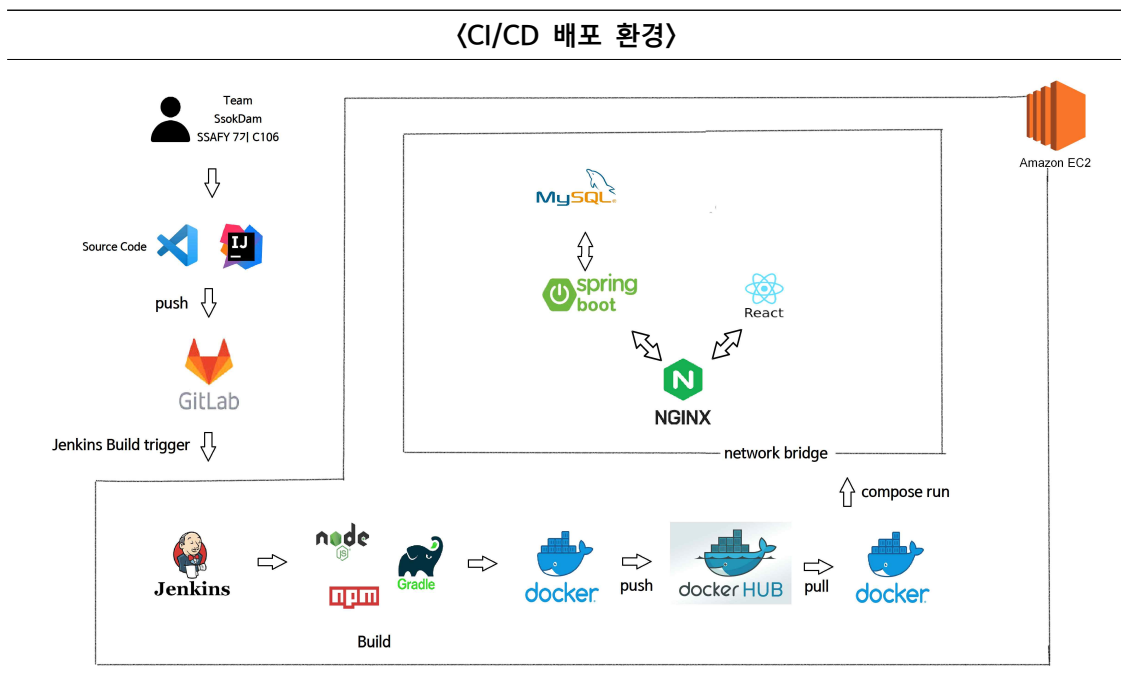
1. 기술스택

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
	Flask		
	spark		
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
	formik		2.2.9
	react-qrcode-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

2. 상세 내용

□ 개요

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



각 프로젝트를 빌드 후에는 Docker 이미지를 만들고 이를 컨테이너로 띄웁니다. SSAFY에서 지원받은 EC2 싱글 인스턴스로 인프라를 구축했습니다. 추후 서비스화를 위해 Nginx는 리버스 프록시 서버로 설정하였습니다. 동시에 Nginx 서버로부터 8000 포트를 BackEnd 서버로 설정하여 구축했고, 3000포트를 FrontEnd 서버로 설정하여 구축했습니다.

□ FrontEnd

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

```
FROM node:16.17.0-alpine
COPY package.json ./
COPY yarn.lock ./
RUN yarn install
COPY . .
RUN yarn build
EXPOSE 3000
CMD ["yarn","dev"]
```

- docker 이미지 생성

```
docker build -t front:0.1 .
```

- docker 컨테이너 실행

```
docker run -p 3000:3000 --name front front: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 /youtubeAnalysis-0.0.1-SNAPSHOT.jar app.jar
ENV JAVA_OPTS=""
ENTRYPOINT ["java","-jar","/app.jar"]
```

- docker 이미지 생성

```
docker build -t back:0.1 .
```

- docker 컨테이너 실행

```
docker run -p 8000:8000 --name back back:0.1
```

3. 프로퍼티 정의

□ MongoDB

○ 계정 생성

```
create user 'mongoDB@%' 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 {  
    root /var/www/html;  
  
    # Add index.php to the list if you are using PHP  
    index index.html index.htm index.nginx-debian.html;  
    server_name j7c203.p.ssafy.io; # managed by Certbot  
  
    location / {  
        proxy_set_header HOST $host;  
        proxy_pass http://43.200.1.125:3000$request_uri;  
        proxy_redirect off;  
        proxy_buffer_size 128k;  
        proxy_buffers 4 256k;  
        proxy_busy_buffers_size 256k;  
    }  
    location /api {  
        proxy_set_header HOST $host;  
        proxy_pass http://43.200.1.125:8000$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/j7c203.p.ssafy.io/fullchain.pem; # managed by Certbot  
    ssl_certificate_key /etc/letsencrypt/live/j7c203.p.ssafy.io/privkey.pem; # managed by Certbot  
    ssl_dhparam /etc/letsencrypt/ssl-dhparams.pem; # managed by Certbot  
}  
  
server {  
    if ($host = j7c203.p.ssafy.io) {  
        return 301 https://$host$request_uri;  
    } # managed by Certbot  
  
    listen 80 ;  
    listen [::]:80 ;  
    server_name j7c203.p.ssafy.io;  
    return 404; # managed by Certbot  
}
```