



A302: StudyWithUs (SWUS : 스윗어스)

포팅 매뉴얼

삼성SW청년아카데미 서울캠퍼스 8기

공통프로젝트 (23.01.03 ~ 23.02.17)

노정환, 서형준, 신민혜, 이정현, 임혜진, 조혜진

목차

0. StudyWithUs (스윗어스)이란?

1. 프로젝트 기술스택

- 1.1. 이슈관리
- 1.2. 형상관리
- 1.3. 커뮤니케이션
- 1.4. 개발 환경과 기술스택

2. EC2 개발환경 세팅

- 2.1 AWS EC2 Docker 세팅
- 2.2 AWS EC2 DB 세팅
- 2.3 AWS EC2 Nginx, HTTPS 세팅
- 2.4 AWS EC2 Jenkins 세팅

3. 프론트엔드 빌드&배포 설정

- 3.1 Jenkins CI/CD
- 3.2 Nginx 구성파일
- 3.3 Dockerfile

4. 백엔드 빌드&배포 설정

- 4.1 Jenkins CI/CD
- 4.2 Dockerfile

5. 외부 서비스

- 5.1 Google Gmail SMTP
- 5.2 OpenVidu

0. StudyWithUs (스윗어스)이란?

팬데믹 이후 많은 사람들이 오프라인으로 독서실이나 카페에 가서 공부하기 부담을 느끼고 있습니다. Study with Us (SWUS) 는 이러한 부담감을 덜어주기 위함엔 집중했습니다. 또한 다른 사람들이 공부하는 모습과 그 날의 학습 시간들을 보면서 서로 자극을 주며 같이 집중할 수 있게 해주는 러닝 메이트의 역할을 해주고 싶었습니다.

집에서 혼자 하면 공부하면 집중되지 않은 많은 사람들이 밖에 나가지 않고 다른 사람들과 같이 공부하고 있다는 느낌을 주어 몰입감있는 공부 환경을 만들고자 했습니다.

1. 프로젝트 기술스택

본 프로젝트에서 사용한 툴과 버전 정보는 아래와 같습니다.

1.1. 이슈관리

- Jira

1.2. 형상관리

- GitLab

1.3. 커뮤니케이션

- 소통: Mattermost, Webex, KakaoTalk

- 문서작업: Notion

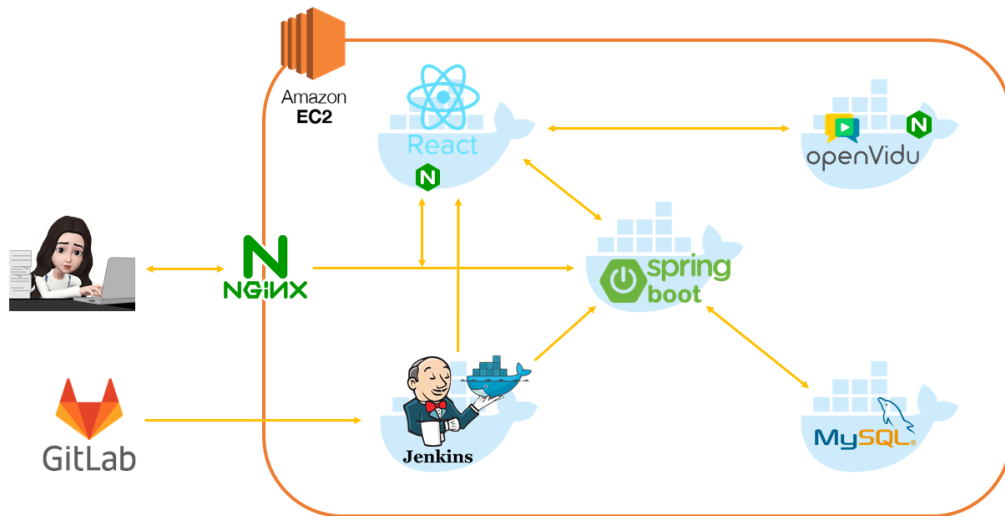
1.4. UI/UX

- Figma

1.4. 개발 환경과 기술스택

개발 환경	기술 스택
Server	- AWS EC2 - OS: Ubuntu 22.04.1 LTS
IDE	- Backend: IntelliJ IDEA 2022.1.3 - Frontend: Visual Studio Code 1.75.1
DevOps	- GitLab - Nginx: nginx/1.18.0 (Ubuntu) - Jenkins 2.375.2 - Docker version 23.0.0
SSL	- letsencrypt - certbot 0.40.0
DataBase	- MySQL 8.0.32
Backend	- Java OpenJDK 1.11.0 - Springboot Gradle 2.7.8 - Spring Data JPA - Lombok - Junit5
Frontend	- NodeJS 18.13.0 - npm 8.19.3 - react 18.12.0 - redux 4.2.0 - material ui 5.11.8
Others	- OpenVidu 2.25.0 - Google Gmail SMTP

2. EC2 개발환경 세팅



2.1. AWS EC2 Docker 세팅

1. apt-get 패키지를 업데이트후, EC2 에 Docker 설치한다.

```
$ sudo apt-get update

# Docker 최신버전 설치
$ sudo apt-get install docker-ce docker-ce-cli containerd.io

# Docker 설치 확인
$ sudo docker -v

# 필요시에는 docker-compose도 설치해준다.
$ sudo apt-get install docker-compose
$ docker-compose --version
```

2.2 AWS EC2 DB 세팅

1. mysql docker hub 에서 mysql 을 pull 받는다.

```
$ docker pull mysql:8.0.32
```

2. pull 받은 이미지로 도커 컨테이너를 돌린다.

```
$ docker run --name mysql-container -e MYSQL_ROOT_PASSWORD=[YOUR_PASSWORD] \
  -d -p [exposed_port_num]:3306 mysql:8.0.32
```

2.3 AWS EC2 Nginx, HTTPS 세팅

1. EC2 에 Let's Encrypt 를 설치하고, SSL 인증서를 발급받는다.

```
# Let's Encrypt 설치
$ sudo apt-get install letsencrypt

# 인증서 적용 및 .pem 키 발급
$ sudo letsencrypt certonly --standalone -d [도메인]
```

2. EC2 에 Nginx 를 설치한다.

```
# apt-get 업데이트
$ sudo apt-get update

# nginx 설치하기
$ sudo apt install nginx -y

# nginx 실행
$ sudo service nginx start
```

3. /etc/nginx/sites-available 로 이동한 후, Nginx 설정 파일을 작성한다.

```
server {
    location /{
        proxy_pass http://localhost:3000;
    }

    location /api {
        proxy_pass http://localhost:8081;
    }

    listen 443 ssl; # managed by Certbot
    ssl_certificate /etc/letsencrypt/live/i8a302.p.ssafy.io/fullchain.pem; # managed by Certbot
    ssl_certificate_key /etc/letsencrypt/live/i8a302.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 {
    if ($host = i8a302.p.ssafy.io) {
        return 301 https://$host$request_uri;
    } # managed by Certbot

    listen 80;
    server_name i8a302.p.ssafy.io;
    return 404; # managed by Certbot
}
```

4. 파일을 연동하고 테스트 후 Nginx 를 재시작해준다.

```
$ sudo ln -s /etc/nginx/sites-available/[파일명] /etc/nginx/sites-enabled/[파일명]

# 다음 명령어에서 successful 이 뜨면 nginx를 실행할 수 있다.
$ sudo nginx -t

# nginx 재시작
$ sudo systemctl restart nginx
```

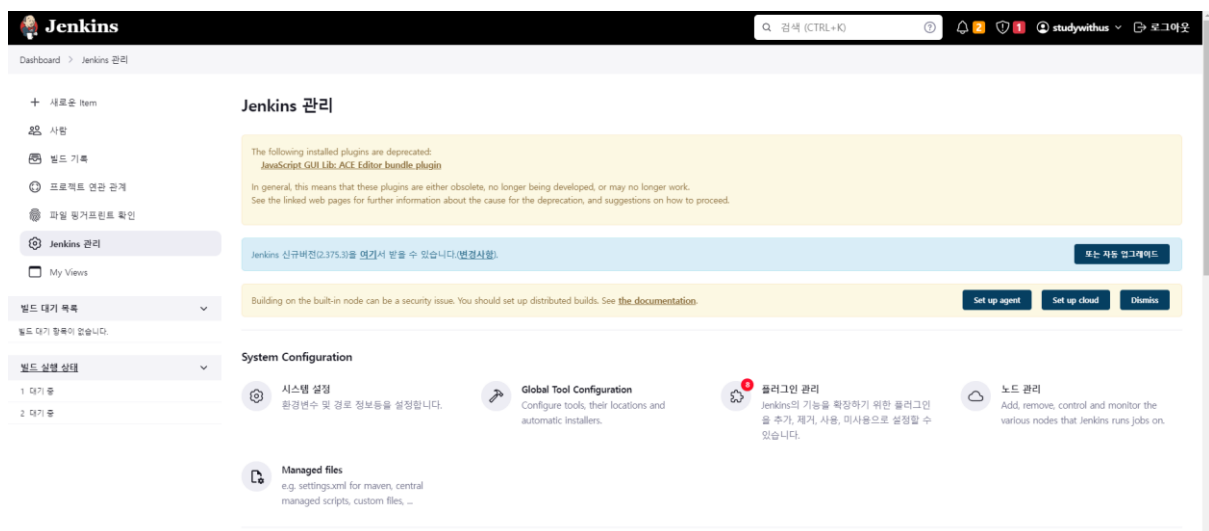
2.4 AWS EC2 Jenkins 세팅

1. 젠킨스 컨테이너를 돌린다.

```
$ sudo docker run -itd --name jenkins -p 8080:8080 -v /docker/jenkins:/var/jenkins_home \
-v /var/run/docker.sock:/var/run/docker.sock -e TZ=Asia/Seoul -u root jenkins/jenkins:its
```

2. 젠킨스에 접속해서 Admin 계정을 생성하고 플러그인들을 설치한다.

배포하는데 설정이 필요한 GitLab, Publish Over SSH, NodeJS 등을 포함해 필요한 플러그인들을 설치한다.



```
server {
    listen 3000;
    location / {
        root /app/build;
        index index.html;
        try_files $uri $uri/ /index.html;
    }
}
```


3.3. Dockerfile

```
FROM nginx

RUN mkdir /app

WORKDIR /app

RUN mkdir ./build

ADD ./build ./build

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

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

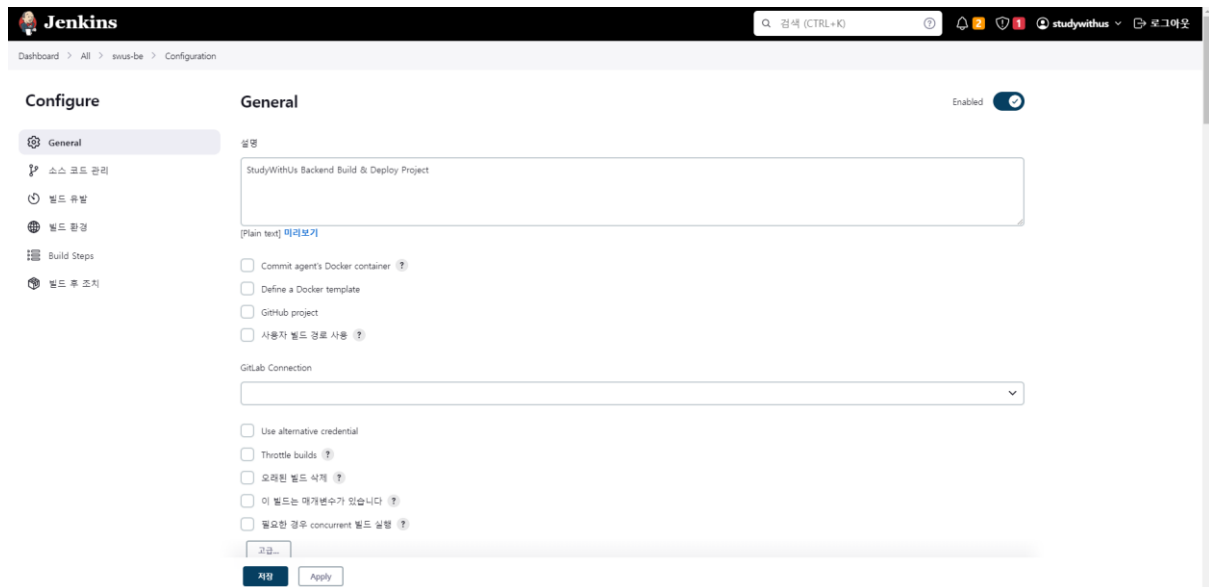
EXPOSE 80

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

4. 백엔드 빌드&배포 설정

4.1. Jenkins CI/CD

Freestyle project 로 생성하고 배포를 위한 설정을 해준다.



4.2 Dockerfile

```
FROM openjdk:11-jdk
VOLUME /tmp
EXPOSE 8081

ARG JAR_FILE=build/libs/swus-0.0.1-SNAPSHOT.jar

COPY ${JAR_FILE} app.jar

ENTRYPOINT ["java", "-jar", "/app.jar"]
ARG DEBIAN_FRONTEND=noninteractive
ENV TZ=Asia/Seoul
RUN apt-get install -y tzdata
```

5. 외부 서비스

5.1 OpenVidu

WebRTC 를 구현하기 위해 OpenVidu 를 사용하였습니다

5.1.1 OpenVidu 설치

EC2 에 먼저 Docker, Docker-Compose 가 설치된 상태에서 root 권한으로 설치를 진행한다. ``cd /opt`` 를 입력해서 OpenVidu 설치 권장 폴더 `/opt` 이동을 한다. 아래 명령어를 입력해서 ``/opt`` 경로에 OpenVidu 를 설치한다.

```
curl https://s3-eu-west-1.amazonaws.com/aws.openvidu.io/install_openvidu_latest.sh | bash
```

설치 후 `cd openvidu` 를 입력하여 폴더로 이동한다. 도메인 설정과 OpenVidu 통신을 위한 Secret 값이 담긴 `.env` 설정 파일을 `nano .env` 명령어 입력하여 아래와 같이 수정한다.

```
DOMAIN_OR_PUBLIC_IP=<DOMAIN> # ex) i8a302.p.ssafy.io
OPENVIDU_SECRET=<SECRET_STRING> # ex) SWUS

# 자가서명 인증서도 사용 가능하지만, 오류창이 뜨는 issue가 존재하므로 이미 발급한 letsencrypt 사용
CERTIFICATE_TYPE=letsencrypt
LETSencrypt_EMAIL=<YOUR_EMAIL> # ex) no-reply-swus@gmail.com

HTTP_PORT=<HTTP_PORT_NUMBER> # ex) 8442
HTTPS_PORT=<HTTPS_PORT_NUMBER> # ex) 8443
```

5.1.2 OpenVidu 실행

`/opt/openvidu` 경로에서 `./openvidu start` 를 입력하여 OpenVidu 를 실행한다. 만약 OpenVidu 의 화상 미팅 실행하는데 안된다면, 아래 명령어를 입력하여 kms 를 따로 실행시켜준다.

```
docker run -d -p 4443:4443 -e OPENVIDU_SECRET=SWUS openvidu/openvidu-server-kms:latest
```

5.2 Google Gmail SMTP

비밀번호 찾기 시, 적절한 회원 정보를 입력하면 기존 비밀번호의 일부를 등록한 이메일로 전송하였습니다. Gmail SMTP 를 사용하기 위해서는 1) IMAP 을 사용하도록 설정한 뒤, 2) 앱 비밀번호를 발급받아 3) 이를 바탕으로 spring-boot-starter-mail 이 해당 계정에 접근해 메일을 발송할 수 있도록 해야 합니다.

설정

기본설정 라벨 받은편지함 계정 및 가져오기 필터 및 차단된 주소 전달 및 POP/IMAP 부가기능 채팅 및 Meet 고급 오프라인 테마

전달: [자세히 알아보기](#) 전달 주소 추가

도움말: [필터를 만들면](#) 메일 중 일부만 전달할 수도 있습니다.

POP 다운로드:
[자세히 알아보기](#)

1. 상태: POP 사용 안함
☐ 모든 메일에 POP 사용하기
☐ 지금부터 수신되는 메일에만 POP를 사용하기

2. POP로 메시지를 여는 경우 Gmail 사본을 받은편지함에 보관하기

3. 이메일 클라이언트 구성 (예: Outlook, Eudora, Netscape Mail)
[설정 방법](#)

IMAP 액세스:
(IMAP을 사용하여 다른 클라이언트에서 Gmail에 액세스)
[자세히 알아보기](#)

상태: IMAP을 사용할 수 있습니다.
☒ IMAP 사용
☐ IMAP 사용 안함

IMAP에서 메일을 삭제된 것으로 표시하는 경우:
☒ 자동 삭제 사용 - 서버를 즉시 업데이트(기본값)
☐ 자동 삭제 사용 안함 - 클라이언트가 서버를 업데이트할 때까지 대기

메일이 삭제된 것으로 표시되고 마지막으로 표시된 IMAP 폴더에서 삭제된 경우:
☒ 메일 보관(기본값)
☐ 메일을 휴지통으로 이동
☐ 메일을 즉시 완전삭제

폴더 크기 제한
☒ IMAP 폴더에서 메일 수를 제한하지 않습니다(기본값).
☐ 이만큼의 메일만 포함하도록 IMAP 폴더를 제한합니다. 1,000

이메일 클라이언트 구성(예: Outlook, Thunderbird, iPhone)
[설정 방법](#)

변경사항 저장 취소

< 앱 비밀번호

앱 비밀번호를 사용하면 2단계 인증을 지원하지 않는 기기의 앱에서 Google 계정에 로그인할 수 있습니다. 비밀번호를 한 번만 입력하면 기억할 필요가 없습니다. [자세히 알아보기](#)

앱 비밀번호

생성된 앱 비밀번호


Email

securesally@gmail.com

Password

••••••••••

기기용 앱 비밀번호



사용 방법

설정하려는 애플리케이션 또는 기기의 Google 계정 설정으로 이동합니다. 비밀번호를 위에 표시된 16자리 비밀번호로 교체합니다. 일반적인 비밀번호와 마찬가지로 이 앱 비밀번호는 Google 계정에 대한 완전한 액세스 권한을 부여합니다. 비밀번호를 기억하지 않아도 되므로 적어 놓거나 다른 사용자와 공유하지 마세요.

[확인](#)