

# Catch up 포팅 매뉴얼

삼성 청년 SW 아카데미 서울 캠퍼스 10기 공통 프로젝트 (23.02.03 ~ 23.02.16)

담당 컨설턴트 : 오형남

A105 - 박근수(팀장) , 박기남, 손지은, 손준성, 허준호

# 목차

- 1. 프로젝트 기술 스택
- 2. EC2 서버 세팅
- 3. 빌드 상세 내용
- 4. 외부 서비스

# 1. 프로젝트 기술 스택

- 이슈 관리 : Jira
- 버전 관리 : Gitlab
- 의사 소통, 협업 : Notion, Jira, MatterMost
- 개발 환경
  - \* OS : Window
  - \* IDE : IntelliJ, WebStorm, VS Code
- 배포 환경
  - \* 서버 : AWS EC2
  - \* OS : Ubuntu 20.04.6 LTS (Focal Fossa)
  - \* DataBase
    - MySQL 8.2
    - Redis
  - \* CI / CD : shell script
  - \* Reverse Proxy : NginX
  - \* SSL : CertBot, Let's Encrypt
  - \* Media Server : Kurento 7.0.0
  - \* TurnServer : Coturn
- 프론트엔드 (React)
  - \* React
  - \* Styled-component
  - \* StompJS
  - \* Axios
  - \* Recoil
  - \* Recoil-persist
- 백엔드 (Spring boot)
  - \* Java : zulu 17
  - \* Stomp
  - \* Spring Session
  - \* Spring Security
  - \* Spring JPA
  - \* Lombok

## 2. EC2 서버 설정

### ❖ Docker 설치

```
apt-get update
```

```
apt-get install sudo
```

```
apt-get install vim
```

```
sudo apt-get install apt-transport-https ca-certificates curl gnupg-agent software-properties-common
```

```
curl -fsSL https://download.docker.com/linux/ubuntu/gpg | sudo apt-key add
```

```
sudo add-apt-repository \
```

```
"deb [arch=amd64] https://download.docker.com/linux/ubuntu \
```

```
$(lsb_release -cs) stable" sudo apt-get update && sudo apt-get install docker-ce docker-ce-cli containerd.io
```

## 2. EC2 서버 설정

### ❖ Docker-Compose 설치

```
sudo apt install jq
```

```
VERSION=$(curl --silent https://api.github.com/repos/docker/compose/releases/latest |  
jq .name -r)
```

```
DESTINATION=/usr/bin/docker-compose
```

```
sudo curl -L https://github.com/docker/compose/releases/download/${VERSION}/docker-  
compose-$(uname -s)-$(uname -m) -o
```

```
$DESTINATION
```

```
sudo chmod 755 $DESTINATION
```

## 2. EC2 서버 설정

### ❖ 방화벽 설정

서버에서 사용할 포트는 22, 80, 443, 8080번 포트이므로,  
해당 포트들을 ufw를 이용해 개방하고,  
ufw를 실행시킵니다

```
Sudo ufw allow 8443
```

```
Sudo ufw allow 3478
```

```
Sudo ufw allow 8080
```

```
Sudo ufw allow 3000
```

### ❖ Coturn 설치

Coturn은 EC2 로컬에 설치되어 사용됩니다.

```
sudo apt-get update
```

```
sudo apt-get install --no-install-recommends coturn
```

## 2. EC2 서버 설정

### ❖ Nginx 설치

```
sudo apt-get install nginx
```

그 이후 아래의 폴더에서 리버스 프록시 관련 설정을 진행합니다.

```
ubuntu@ip-172-26-11-74: /etc/nginx/conf.d$ ls -l
total 4
-rw-r--r-- 1 root root 1098 Feb  7 12:14 my-server.conf
```

Code: [https://lab.ssafy.com/s10-webmobile1-sub2/S10P12A105/-/blob/master/Architecture/nginx\\_my-server.conf?ref\\_type=heads](https://lab.ssafy.com/s10-webmobile1-sub2/S10P12A105/-/blob/master/Architecture/nginx_my-server.conf?ref_type=heads)

## 2. EC2 서버 설정

### ❖ SSL 적용하기

해당 프로젝트에서는 CertBot과 Let's Encrypt라는 무료 인증서 발급 기관의 도움을 받아 간단한 명령어로 SSL을 적용합니다.

아래의 명령어를 입력해 certbot을 설치합니다.

```
sudo apt-get update
```

```
sudo snap install core; sudo snap refresh core
```

```
sudo snap install certbot --classic
```



## 2. EC2 서버 설정

### ❖ SSL 적용하기

아래의 명령어를 입력하여 SSL을 자동으로 적용합니다.

```
sudo certbot --nginx
```

명령어를 입력하면 CertBot의 절차에 맞춰 이메일을 입력하고 SSL을 적용할 도메인을 선택하여 적용합니다

```
ubuntu@ip-172-26-11-74:/etc/nginx/conf.d$ sudo certbot --nginx
Saving debug log to /var/log/letsencrypt/letsencrypt.log

Which names would you like to activate HTTPS for?
We recommend selecting either all domains, or all domains in a VirtualHost/server block.
-----
1: i10a105.p.ssafy.io
-----
Select the appropriate numbers separated by commas and/or spaces, or leave input
blank to select all options shown (Enter 'c' to cancel):
```

nginx 설정을 끝마친 url이라면 SSL을 적용할 수 있습니다. 그이후 다음의 명령어를 통해 nginx를 재시작합니다. `sudo service nginx restart`

## 2. EC2 서버 설정

### ❖ Shell Script 설정

해당 프로젝트에서는 Shell Script를 사용하여 빌드와 배포를 자동화하였습니다.

```
# 이 스크립트는 특정 Git 레포지토리의 프론트엔드(FE)와 백엔드(BE) 프로젝트를 클론, 빌드 및 실행하는 과정을 자동화합니다.  
# 사용법: ./start.sh [FE_branch_name] [BE_branch_name]  
# FE_branch_name과 BE_branch_name은 선택적입니다. 기본값은 각각 FE_develop와 BE_develop입니다.
```

Code : [https://lab.ssafy.com/s10-webmobile1-sub2/S10P12A105/-/blob/master/Architecture/start.sh?ref\\_type=heads](https://lab.ssafy.com/s10-webmobile1-sub2/S10P12A105/-/blob/master/Architecture/start.sh?ref_type=heads)

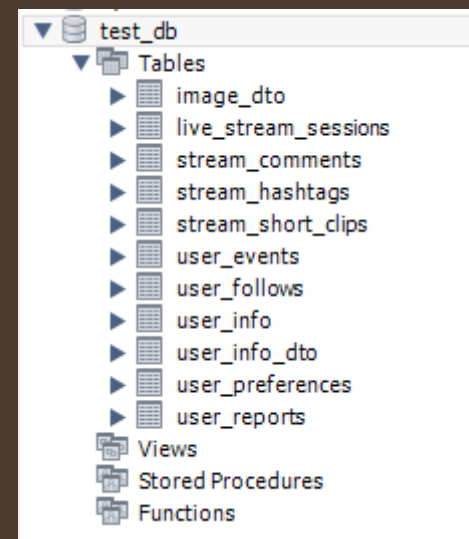
## 2. EC2 서버 설정

### ❖ MySQL 설정

dbInit.sql을 실행하여 데이터베이스를 생성하여 사용할 수 있습니다.

Mysql의 데이터는 docker-compose.yml의 경로를 토대로 볼륨을 사용하여 로컬에 따로 저장됩니다.

Link: [https://lab.ssafy.com/s10-webmobile1-sub2/S10P12A105/-/blob/BE\\_develop/BackEnd/Chocolate/dbInit.sql](https://lab.ssafy.com/s10-webmobile1-sub2/S10P12A105/-/blob/BE_develop/BackEnd/Chocolate/dbInit.sql)



## 2. EC2 서버 설정

### ❖ Coturn Config 설정

turn server의 URL을 공인 IP로 설정해줘야합니다.

/etc/turnserver.conf

/etc/default/coturn

위 두 개의 파일을 gitlab의 파일로 수정해줍니다.

Link: <https://lab.ssafy.com/s10-webmobile1-sub2/S10P12A105/-/blob/master/Architecture/turnserver.conf>,  
<https://lab.ssafy.com/s10-webmobile1-sub2/S10P12A105/-/blob/master/Architecture/coturn>

## 2. EC2 서버 설정

### ❖ Kurento webendpoint.ini 설정

Kurento에서 사용할 설정을 적용합니다.

```
Sudo docker exec -it e758f0f87f42 /bin/bash  
vim /etc/kurento/modules/kurento/WebRtcEndpoint.conf.ini
```

```
stunServerAddress=stun.l.google.com  
stunServerPort=19302  
urnURL=name:credential@URL:port
```

# URL은 turn server가 있는 공인 IP입니다.

# name,credential은 coturn config에서 설정한 username, password입니다.

# 3.빌드 상세 내용

## ❖ Build 과정

실제 프로젝트를 배포해야 할 때에는 필요한 모든 파일들을 적절한 위치에 배치하고, 필요한 환경 변수, API-KEY 등을 전부 입력한 뒤 빌드를 진행해야 합니다.

폴더 명에 맞는 소스 파일을 적절히 위치시키면 빌드 준비는 끝이 납니다.

채워 넣어야 하는 모든 값을 개인의 값으로 채웠다면, BackEnd폴더에서 아래의 명령어를 입력합니다.

```
Sudo docker-compose up -d
```

해당 명령어를 입력하면 docker에 redis, mysql, kurento 서버를 띄워 사용할 수 있게 됩니다.

메인브랜치의 ./start.sh를 실행하여 빌드/배포를 한번에 진행할 수 있습니다.

## 4.외부 서비스

### ❖ Kakao Map API

Kakao에서 발급받은 api-key를 적용하기 위해 프론트엔드의 index.html의 key부분을 수정해주어야 합니다.

```
<script type="text/javascript" src="//dapi.kakao.com/v2/maps/sdk.js?appkey={YOUR_key}&libraries=services"></script>
```

공식문서 : <https://apis.map.kakao.com/web/guide/>

Code : [https://lab.ssafy.com/s10-webmobile1-sub2/S10P12A105/-/blob/FE\\_develop/FrontEnd/public/index.html](https://lab.ssafy.com/s10-webmobile1-sub2/S10P12A105/-/blob/FE_develop/FrontEnd/public/index.html)

## 4.외부 서비스

### ❖ Naver 길찾기 서비스 연동

특정 웹 페이지로 사용자를 리디렉션하여  
네이버의 길찾기 기능을 활용할 수 있습니다.

네이버 길찾기 서비스의 기능은 오른쪽과 같이  
활용할 수 있습니다.

```
{/* <a href="
http://map.naver.com/index.nhn
?slng=127.03770
&slat=37.4996
&stext=김싸피님의 위치
&elng=126.9259
&elat=37.55705
&etext=홍대비비버스킹장소
&menu=route
&pathType=1
">출발지에서 도착지까지 길찾기</a> */}
```