

**BETTING ROYAL**

SSAFY 7기 서울 4반 4팀

(2022.07.05 ~ 2022.08.19)

**포팅 매뉴얼**

팀장 : 유동윤

팀원 : 박수민, 성성민, 이득교, 이상택, 지용현

**< 목차 >**

1. 기술 스택 --------------------------------------------------------- 1
2. AWS에서 빌드 주요 내용 ---------------------------------------- 2
3. 배포 시 특이사항 ------------------------------------------------- 3
4. DB 연결 ----------------------------------------------------------- 3
5. 외부 서비스 ------------------------------------------------------- 5
6. **기술 스택**
7. 개발 환경
   1. OS

Local : Windows 10

AWS : Ubuntu 20.04.4 LTS

* 1. IDE

IntelliJ IDE 2022.1.3

Visual Studio Code 1.70.0

* 1. UI / UX : Figma
  2. DataBase

MySQL workbench 8.0.20 (Windows 10)

MySQL 8.0.30 (Ubuntu 20.04.2 LTS)

* 1. Docker : 20.10.17

1. 상세 스택
   1. Backend

JAVA - openjdk 1.8.0\_342

Gradle 6.7

SpringBoot 2.4.5, Quarydsl 4.4.0, Lombok 1.18.20, Swagger2 3.0.0

Stomp 2.3.3-1

* 1. FrontEnd

HTML5, CSS3, JavaScript(ES6)

React 18.2.0, face-api.js 0.22.2, stompjs 2.3.3, openvidu-browser 2.22.0

Node.js 16.16.0, npm 8.15.0

1. 이슈 관리 : Jira
2. 형상 관리 : Gitlab
3. 커뮤니케이션 : Notion, MatterMost
4. **AWS에서 빌드 주요 내용**
5. Clone

git clone <https://lab.ssafy.com/s07-webmobile1-sub2/S07P12A404.git>

1. Frontend 빌드

[S07P12A404/frontend] 폴더로 들어갑니다.

cd S07P12A404/frontend

package-lock.json에 정의된 모든 dependencies에 대하여 install을 합니다.

npm install

install이 완료되었다면 빌드를 진행합니다.

npm run build

build를 하게 되면 frontend/build 폴더에 프로젝트가 빌드 됩니다. nignx에서 해당 폴더의 index.html을 띄워주도록 설정해 두었습니다.

nginx와 certbot을 install해서 저희 도메인에 대한 인증서를 발급받습니다.

sudo apt-get install -y nginx nginx-common nginx-full

sudo add-apt-repository ppa:certbot/certbot

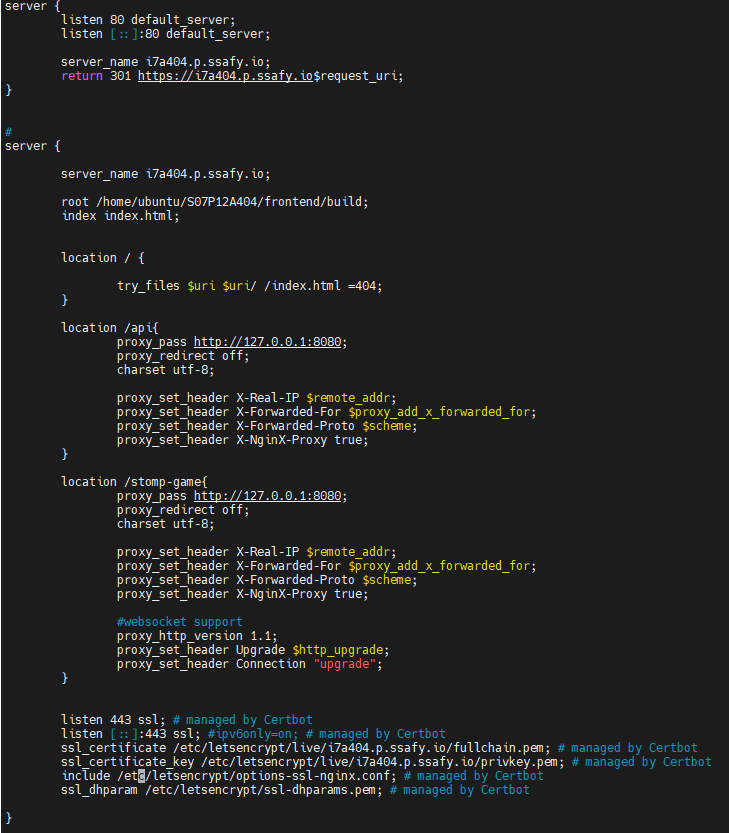
sudo apt-get install python-certbot-nginx

sudo certbot --nginx –d i7a404.p.ssafy.io

인증서를 받은 후 다음 경로로 가서 nginx의 포트, uri 등의 설정을 해줍니다.

cd /etc/ngins/sites-available/

sudo vi default

아래 사진과 같이 설정해줍니다.

1. Backend 빌드

[S07P12A404/backend] 폴더로 들어갑니다.

cd S07P12A404/backend

Build를 하고 jar파일을 실행합니다. nohup을 추가하여 코드가 백그라운드에서 실행되도록 합니다.

./gradlew build

cd build/lib/

nohup java -jar ssafy-web-project-1.0-SNAPSHOT.jar &

1. **배포 시 특이사항**

EC2에서 Kurento Media Server, Coturn, Redis, Nginx, Openvidu-server가 돌아갈 수 있는 docker를 start해야합니다.

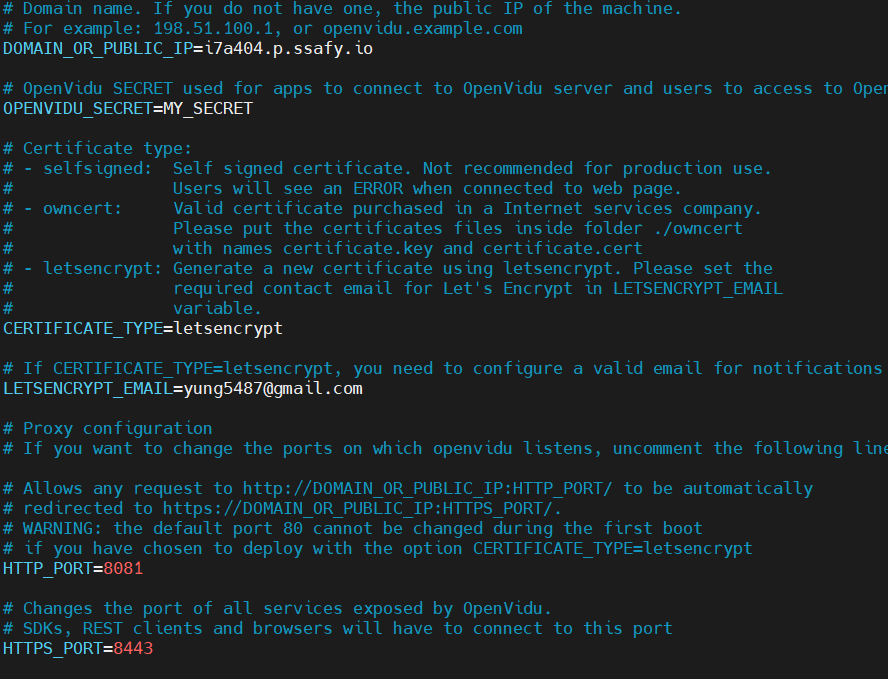
cd /opt/openvidu

nohup sudo ./openvidu start &

Openvidu를 사용하기 위해 /opt/openvidu/.env 파일을 설정합니다.

cd /opt/openvidu

sudo vi .env



\* openvidu on premises를 따라 설정한 후 ./openvidu start를 해도 잘 안되는 경우

./openvidu logs –f nginx 를 실행시켜 nginx의 무료 인증서 받아오기 횟수 초과로

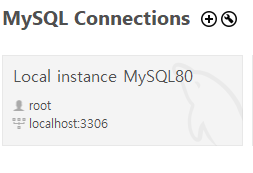
오류가 난다면 서버의 nginx와 certbot을 통해 받은 인증서 /etc/letsencrypt 폴더를

/opt/openvidu/certificates로 옮겨 직접 인증서를 설정해주도록 합니다.

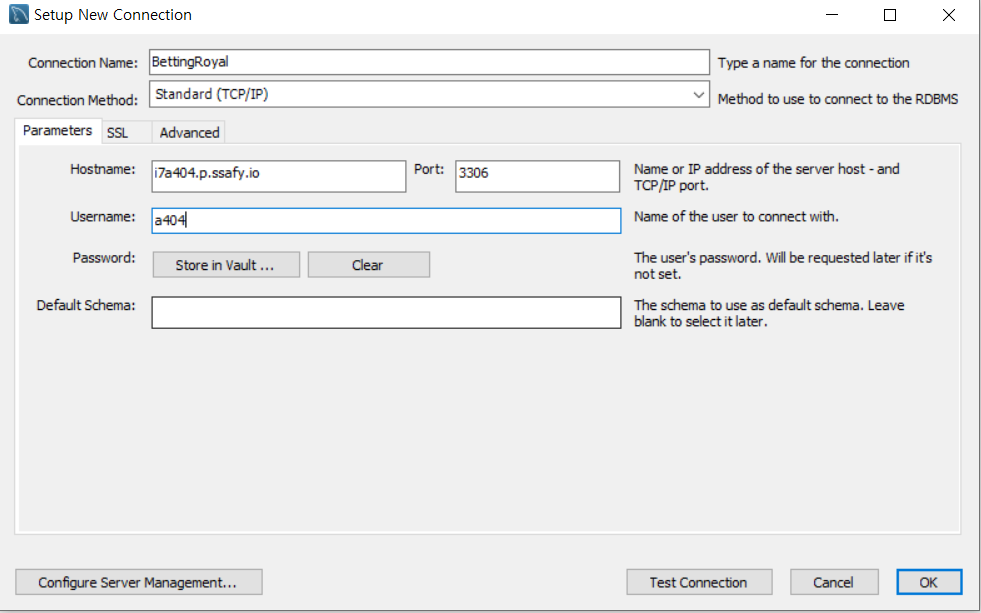
1. **DB 연결**

MySQL server는 EC2 내에서 동작 중인 상태입니다. DB를 Local PC에서도 관리하기 위해서는 아래의 방법을 따라야합니다.

1. Local PC MySQL workbench에서 new Connection



1. 원하는 이름으로 Connection Name를 지정하고 username에 ‘a404’, store in vault에 MySQL server에서 설정한 a404의 비밀번호 ‘ssafy404$)$’를 적습니다.



1. Test Connection을 눌러 Success가 뜬다면 OK버튼을 누릅니다.
2. **외부 서비스**
3. Openvidu

WebRTC를 구현하기 위해 Openvidu를 사용하였습니다.