

# 포팅 메뉴얼

👤 Created By	
🕒 Last Edited Time	@2022년 2월 17일 오후 3:45
👤 Last Edited By	

## 1. Gitlab 소스 클론 이후 빌드 및 배포할 수 있도록 정리한 문서

- JVM : openjdk version "1.8.0\_192"
- WAS : Apache Tomcat/9.0.56
- IDE
  - IntelliJ IDEA 2021.3.1 (Ultimate Edition)
  - VS Code
  - MySQL Workbench 8.0.26

## 배포과정

### a. Ubuntu서버 설정

해당 명령어를 순서대로 실행해서 Docker설치됨

```
sudo apt-get update

sudo apt-get install \
apt-transport-https \
ca-certificates \
curl \
gnupg \
lsb-release

curl -fsSL [https://download.docker.com/linux/ubuntu/gpg](https://download.docker.com/linux/ubuntu/gpg) | sudo apt-key add -

sudo add-apt-repository "deb [arch=amd64] [https://download.docker.com/linux/ubuntu](https://download.docker.com/linux/ubuntu)"

sudo apt-get update

sudo apt-get install docker-ce
```

해당 명령어를 순서대로 실행해서 Docker Compose 설치

```
sudo curl -L "https://github.com/docker/compose/releases/download/1.29.2/docker-compose-$(uname -s)-$(uname -m)" -o /usr/local/bin/docker-compose
sudo chmod +x /usr/local/bin/docker-compose

docker-compose --version 으로 설치 여부를 확인
```

아래 과정을 진행해서 서버에 HTTPS 적용

해당 명령어 순서대로 실행

```
sudo apt-get install nginx
nginx -v
sudo apt-get install letsencrypt
sudo systemctl stop nginx
sudo letsencrypt certonly --standalone -d [도메인주소]
```

아래 그림과 같이 진행

```

root@ip-172-26-2-121:/opt/openvidu# letsencrypt certonly --standalone -d i6e204.p.ssafy.io
Saving debug log to /var/log/letsencrypt/letsencrypt.log
Plugins selected: Authenticator standalone, Installer None
Enter email address (used for urgent renewal and security notices) (Enter 'c' to
cancel): kublaihkan@gmail.com

-----
Please read the Terms of Service at
https://letsencrypt.org/documents/LE-SA-v1.2-November-15-2017.pdf. You must
agree in order to register with the ACME server at
https://acme-v02.api.letsencrypt.org/directory
-----
(A)gree/(C)ancel: A

-----
Would you be willing to share your email address with the Electronic Frontier
Foundation, a founding partner of the Let's Encrypt project and the non-profit
organization that develops Certbot? We'd like to send you email about our work
encrypting the web, EFF news, campaigns, and ways to support digital freedom.
-----
(Y)es/(N)o: Y
Obtaining a new certificate
Performing the following challenges:
http-01 challenge for i6e204.p.ssafy.io
Waiting for verification...
Cleaning up challenges

IMPORTANT NOTES:
 - Congratulations! Your certificate and chain have been saved at:

```

해당 명령어 순서대로 실행

```

cd /etc/nginx/sites-available
vi config

```

다음과 같이 작성

```

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

    location /api {
        proxy_pass http://localhost:8080/api;
    }

    listen 443 ssl;
    ssl_certificate /etc/letsencrypt/live/i6e204.p.ssafy.io/fullchain.pem;
    ssl_certificate_key /etc/letsencrypt/live/i6e204.p.ssafy.io/privkey.pem;
}

server {
    if ($host = i6e204.p.ssafy.io) {
        return 301 https://$host$request_uri;
    }

    listen 80;
    server_name i6e204.p.ssafy.io;
    return 404;
}

```

해당 명령어를 마지막으로 실행 시 설치 끝

```

sudo ln -s /etc/nginx/sites-available/config /etc/nginx/sites-enabled/config
sudo nginx -t
sudo systemctl restart nginx

```

아래 과정을 진행해서 OpenVidu 설치

해당 명령어 순서대로 실행

```

sudo su
cd /opt
curl [https://s3-eu-west-1.amazonaws.com/aws.openvidu.io/install_openvidu_latest.sh](https://s3-eu-west-1.amazonaws.com/aw

```

```
cd openvidu
nano .env
```

아래 그림과 같이 설정

```
# For example: 198.51.100.1, or openvidu.example.com
DOMAIN_OR_PUBLIC_IP=i6e204.p.ssafy.io

# OpenVidu SECRET used for apps to connect to OpenVidu
OPENVIDU_SECRET=YOURSTAR
```

```
CERTIFICATE_TYPE=letsencrypt

# If CERTIFICATE_TYPE=letsencrypt, you need to provide a valid email address
LETSencrypt_EMAIL=user@example.com
```

```
# if you have chosen to deploy with the option CERTIFICATE_TYPE=letsencrypt
HTTP_PORT=8442
```

```
# Changes the port of all services exposed by OpenVidu.
# SDKs, REST clients and browsers will have to connect to this port
HTTPS_PORT=8443
```

```
# Whether to enable recording module or not
OPENVIDU_RECORDING=true
```

```
# Use recording module with debug mode.
OPENVIDU_RECORDING_DEBUG=false
```

```
# Openvidu Folder Record used for save the openvidu recording videos. Change it
# with the folder you want to use from your host.
OPENVIDU_RECORDING_PATH=/opt/openvidu/recordings
```

```
# System path where OpenVidu Server should look for custom recording layouts
OPENVIDU_RECORDING_CUSTOM_LAYOUT=/opt/openvidu/custom-layout
```

```
# if true any client can connect to
# https://OPENVIDU_SERVER_IP:OPENVIDU_PORT/recordings/any_session_file.mp4
# and access any recorded video file. If false this path will be secured with
# OPENVIDU_SECRET param just as OpenVidu Server dashboard at
# https://OPENVIDU_SERVER_IP:OPENVIDU_PORT
# Values: true | false
OPENVIDU_RECORDING_PUBLIC_ACCESS=true
```

```
./openvidu start
```

위 명령어를 통해 실행

===== 여기까지 진행 후 =====

설정한 사이트주소의 포트 8443이 접근이 안된다면 docker-compose logs nginx를 확인

ssl certificate가 없다는 에러가 있다면 HTTPS 구성 때 생성한 키를 수동으로 옮겨야함

Ubuntu 서버에 /etc/letsencrypt/live/[사이트명] 으로 이동 후 모든 파일들을 복사

/opt/openvidu/certificates/live/[사이트명] 에 붙여넣어준다.

docker-compose restart nginx 명령어 입력하면 정상 작동

b. 이미지 제작은 개발 PC에서 Docker 20.10.12 사용, 서버에서 받아서 배포

a. Backend

- a. Gradle에 bootJar 를 이용해 Jar 파일 생성
  - b. docker build -t [이미지명] . 을 이용해 이미지 제작
  - c. sudo docker run -d -v /home/ubuntu/upload:/home/ubuntu/upload -p 8080:8080 —name backend [이미지명] 을  
사용해서 서버에서 실행
  - d. 위의 -v 명령어인 볼륨매핑을 해줘야 이미지 저장 및 불러오기가 제대로 작동
- b. Frontend
- a. npm install 로 필요 파일 다운
  - b. npm run build 로 빌드 파일 생성
  - c. docker build -t [이미지명] . 을 이용해 이미지 제작
  - d. docker run -d -p 3000:3000 —name frontend [이미지명]
- c. DB
- a. [i6e204.p.ssafy.io:3306](https://i6e204.p.ssafy.io:3306) MySql 사용
  - b. schema : yourstar
  - c. username : yourstar, password : yourstar