배포가이드

기술스택

백엔드

- IntelliJ
- Spring boot 2.7.2
- java 17
- gradle 7.5
- Lombok
- JPA
- SockJS

- Stomp
- MySql 5.7
- OpenVidu 2.22.0

프론트엔드

- vscode
- node 16.16.0
- Vue2
- Vuetify
- mediapipe

서버

- AWS
- Docker
- nginx

프론트엔드

1. 환경변수 설정

```
# moweb_front 폴더 안에 .env 파일 생성 후 아래 변수 작성

# 백엔드 api url
VUE_APP_MOWEB_API_URL = https://i7a507.p.ssafy.io/moweb-api

# 루트 서버 url
VUE_APP_ROOT_URL = https://i7a507.p.ssafy.io

# 오픈비두 url
VUE_APP_OPENVIDU_SERVER_URL = https://i7a507.p.ssafy.io:8443
```

```
# 카카오 api 키
VUE_APP_KAKAO_API_KEY = 59074e20c9d80e6e5200a4bd60122af7

# 오픈비두 시크릿 키
VUE_APP_OPENVIDU_SERVER_SECRET = MY_SECRET

# 백엔드 websocket url
VUE_APP_STOPM_SERVER_URL = https://i7a507.p.ssafy.io/moweb-api/ws/moweb
```

2. npm install

```
# 패키지 설치
npm install
```

3. 개발환경 실행

```
# 8081 포트로 실행됨
npm run serve
```

4. 빌드

```
npm run build
```

백엔드

1. application.properties 변수

아래 3가지 방법 중 하나 사용

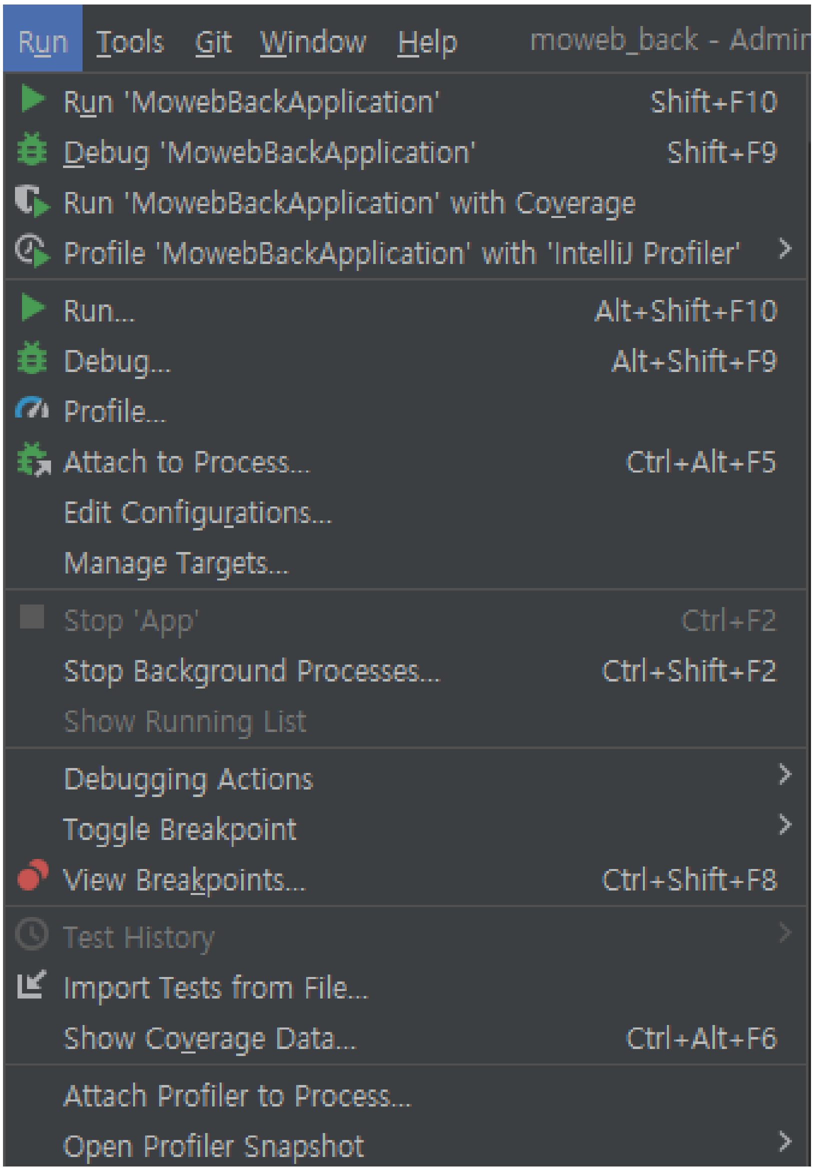
- application.properties 직접 설정

- 시스템 환경변수 설정
- intellij run configuration 설정

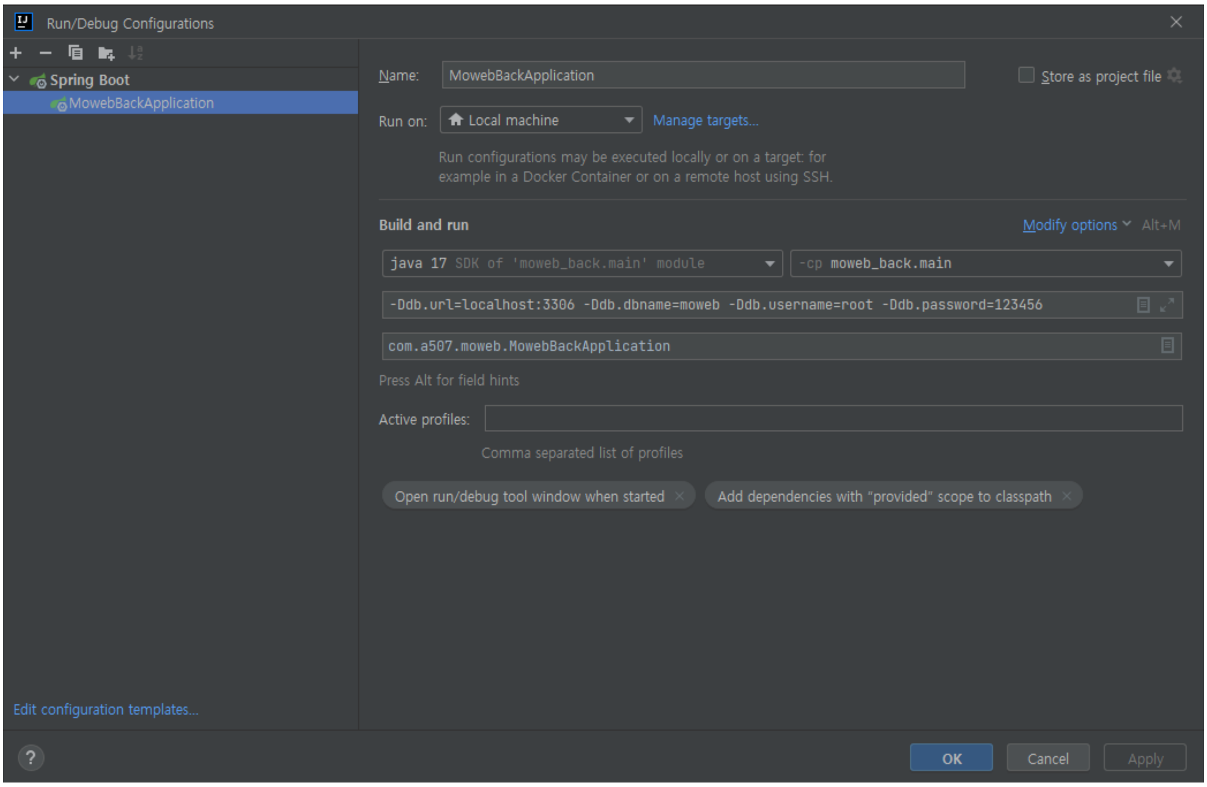
```
# 데이터베이스 이름
db_dbname

# 데이터베이스 패스워드
db_password

# 데이터베이스 url
db_url

# 데이터베이스 아이디
db_username

# swagger url
hosturl
```

1. 로컬 실행

Menu → Run → Edit Configurations 선택


Run Tools Git Window Help
moweb_back - Admir
Run 'MowebBackApplication' Shift+F10
Debug 'MowebBackApplication' Shift+F9
Run 'MowebBackApplication' with Coverage
Profile 'MowebBackApplication' with 'IntelliJ Profiler'
Run... Alt+Shift+F10
Debug... Alt+Shift+F9
Profile...
Attach to Process... Ctrl+Alt+F5
Edit Configurations...
Manage Targets...
Stop 'App' Ctrl+F2
Stop Background Processes... Ctrl+Shift+F2
Show Running List
Debugging Actions
Toggle Breakpoint
View Breakpoints... Ctrl+Shift+F8
Test History
Import Tests from File...
Show Coverage Data... Ctrl+Alt+F6
Attach Profiler to Process...
Open Profiler Snapshot

빌드 옵션에 application.properties에 필요한 값 옵션 설정



Run/Debug Configurations

Name: MowebBackApplication

Run on: Local machine

Run configurations may be executed locally or on a target: for example in a Docker Container or on a remote host using SSH.

Build and run

```
java 17 SDK of 'moweb_back.main' module
-cp moweb_back.main
-Ddb.url=localhost:3306 -Ddb.dbname=moweb -Ddb.username=root -Ddb.password=123456
com.a507.moweb.MowebBackApplication
```

Press Alt for field hints

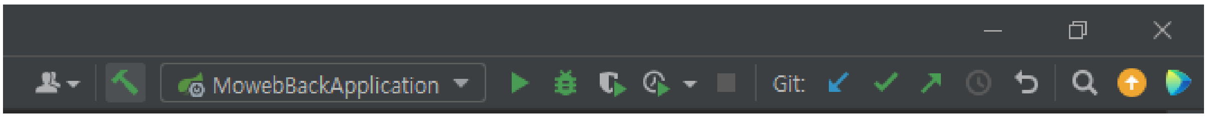
Active profiles:

Comma separated list of profiles

Open run/debug tool window when started ×

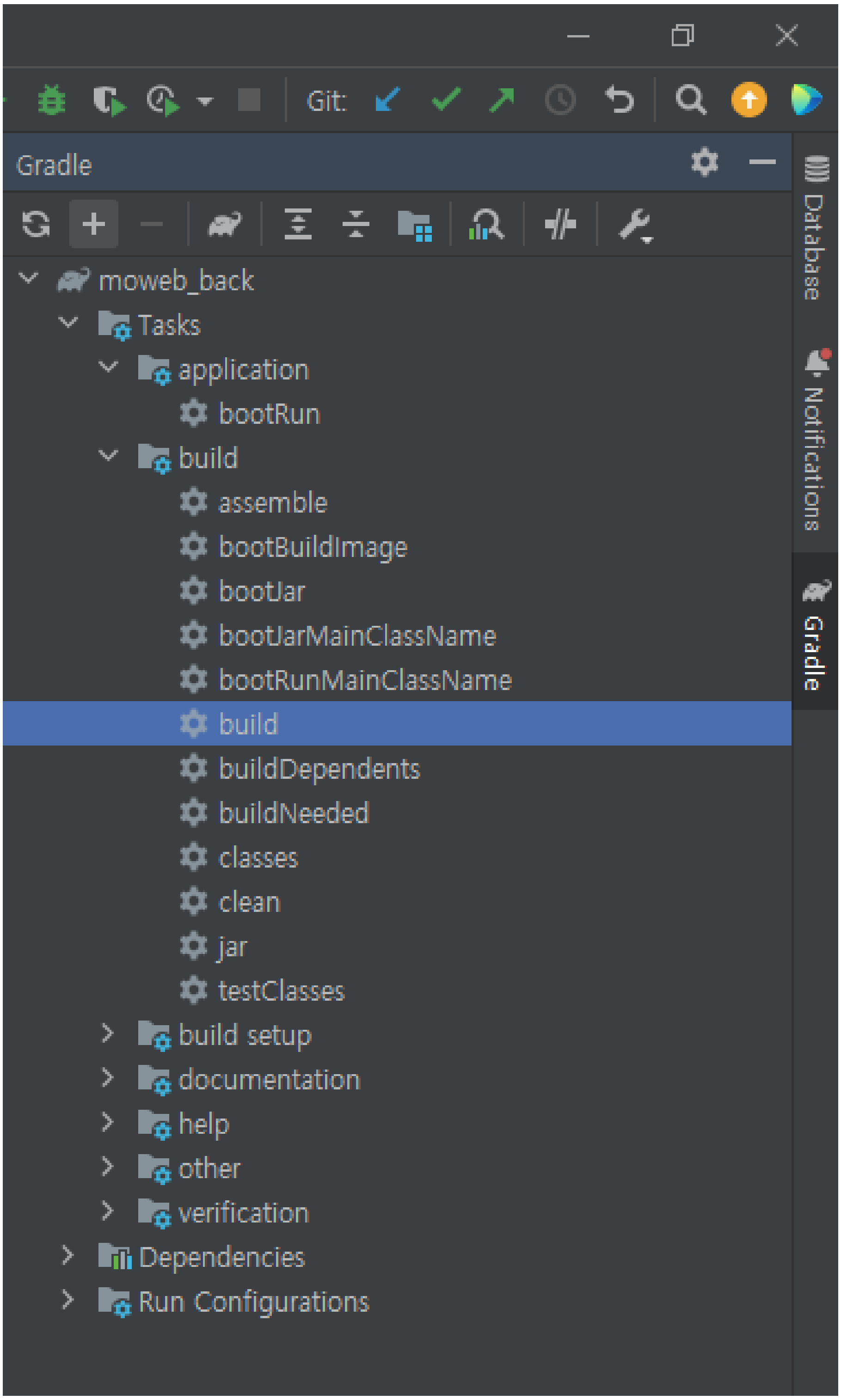
Add dependencies with "provided" scope to classpath ×

실행



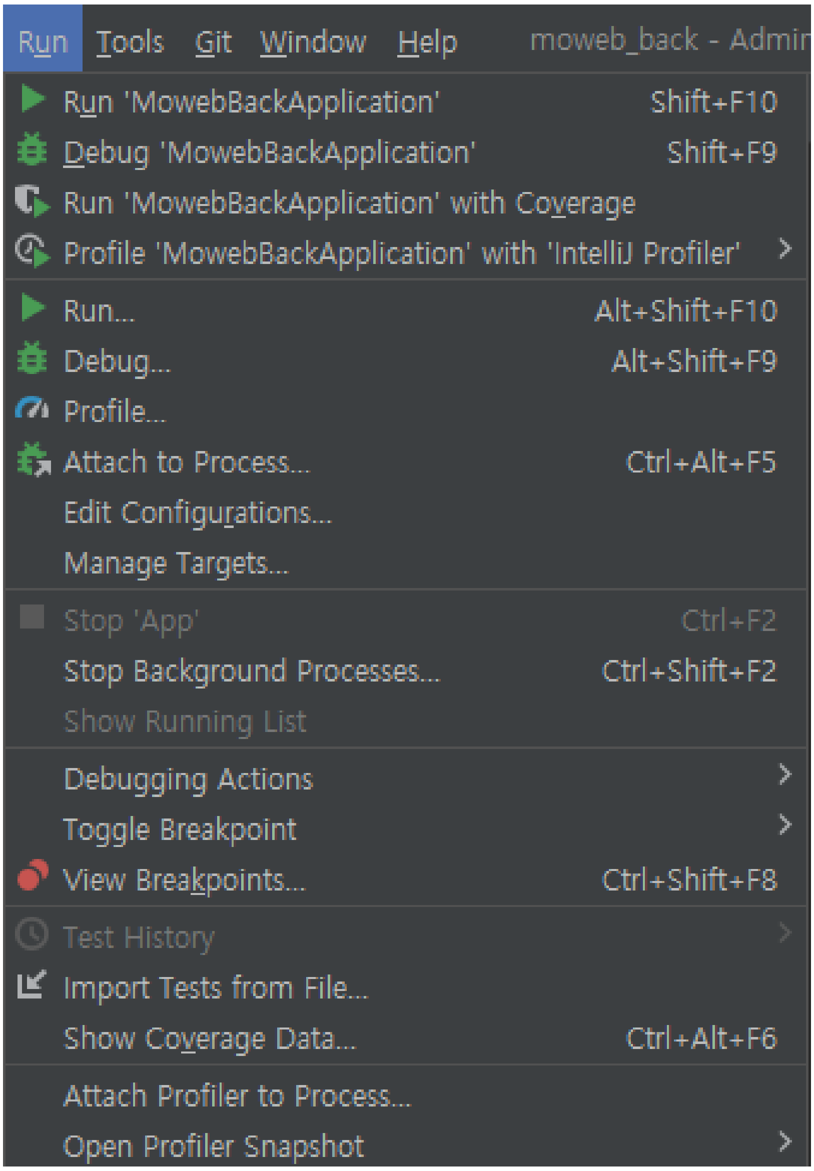
2. intellij 빌드

오른쪽 메뉴 → build → build 선택

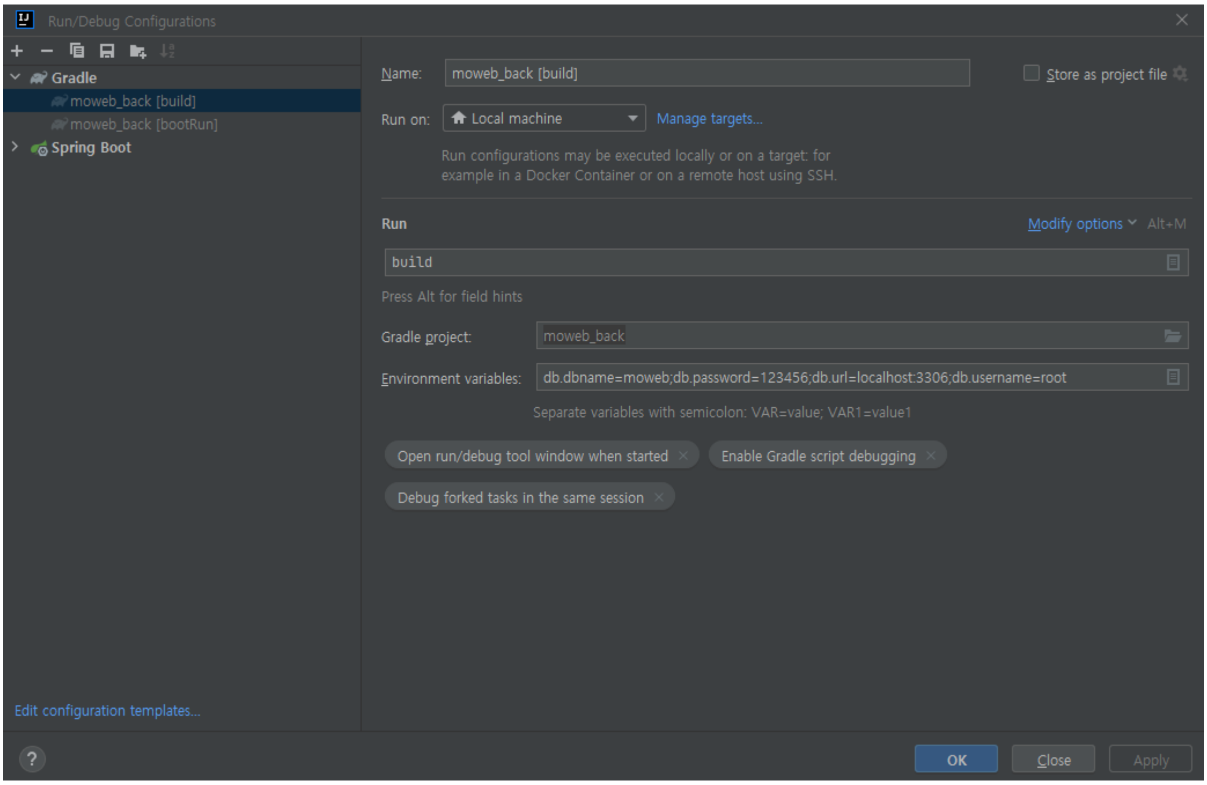


moweb_back
- Tasks
  - application
    - bootRun
  - build
    - assemble
    - bootBuildImage
    - bootJar
    - bootJarMainClassName
    - bootRunMainClassName
    - build
    - buildDependents
    - buildNeeded
    - classes
    - clean
    - jar
    - testClasses
  - build setup
  - documentation
  - help
  - other
  - verification
- Dependencies
- Run Configurations

Menu → Run → Edit Configurations 선택


Run Tools Git Window Help
moweb_back - Admir
Run 'MowebBackApplication' Shift+F10
Debug 'MowebBackApplication' Shift+F9
Run 'MowebBackApplication' with Coverage
Profile 'MowebBackApplication' with 'IntelliJ Profiler'
Run... Alt+Shift+F10
Debug... Alt+Shift+F9
Profile...
Attach to Process... Ctrl+Alt+F5
Edit Configurations...
Manage Targets...
Stop 'App' Ctrl+F2
Stop Background Processes... Ctrl+Shift+F2
Show Running List
Debugging Actions
Toggle Breakpoint
View Breakpoints... Ctrl+Shift+F8
Test History
Import Tests from File...
Show Coverage Data... Ctrl+Alt+F6
Attach Profiler to Process...
Open Profiler Snapshot

Gradle → build → Environment variables 설정



Name: moweb_back [build]

□ Store as project file

Run on: Local machine

Run configurations may be executed locally or on a target: for example in a Docker Container or on a remote host using SSH.

Run

build

Press Alt for field hints

Gradle project: moweb_back

Environment variables: db.dbname=moweb;db.password=123456;db.url=localhost:3306;db.username=root

Separate variables with semicolon: VAR=value; VAR1=value1

Open run/debug tool window when started

Enable Gradle script debugging

Debug forked tasks in the same session

3. command line build

```
# 빌드
./gradlew clean build

# 실행
java -jar build/libs/moweb_back-0.0.1-SNAPSHOT.jar --db.url=localhost:3
306 --db.dbname=moweb --db.username=root --db.password=123456
```

배포

1. 서버 시간 설정

```
# 1. 서버 시간 확인
date

# 2. 서버 시간 변경
sudo ln -sf /usr/share/zoneinfo/Asia/Seoul /etc/localtime
```

2. 도커 설치

```
#1. apt update
sudo apt update

#2. docker 설치
sudo apt install docker.io

#3. 도커 권한 설정
sudo chmod 666 /var/run/docker.sock
```

3. 도커 컴포즈 설치

```
#1. 다운받을 폴더 생성
mkdir -p ~/.docker/cli-plugins/

#2. 다운받기
curl -SL https://github.com/docker/compose/releases/download/v2.9.0/doc
ker-compose-linux-x86_64 -o ~/.docker/cli-plugins/docker-compose

#3. 실행권한 설정
chmod +x ~/.docker/cli-plugins/docker-compose

#4. 링크파일 생성
ln -s ~/.docker/cli-plugins/docker-compose /usr/bin/docker-compose
```

4. mysql 설정

```
#1. 도커 mysql 5.7 이미지 pull
sudo docker pull mysql:5.7

#2. 도커 MYSQL 설치
sudo docker run -d -p 3306:3306 -v ~/mysql:/var/lib/mysql -e MYSQL_ROOT
_PASSWORD='5moweB0!@7' --name mysql5.7 mysql:5.7 --character-set-server
=utf8mb4 --collation-server=utf8mb4_unicode_ci
```

```
#3. 도커 mysql 접속
sudo docker exec -it mysql5.7 mysql -u root -p

#4. mysql root 이름 변경
update user set user='a507' where user='root';

#5. 테이블 생성
CREATE DATABASE moweb DEFAULT CHARACTER SET utf8mb4 COLLATE utf8mb4_general_ci;

#6. 계정 생성
create user 'user_a507'@localhost identified by '5moweB0!@7';
create user 'user_a507'@'%' identified by '5moweB0!@7';

#7. schema 생성
create database moweb;

#8. moweb 데이터베이스 권한 부여
grant all privileges on moweb.* to 'user_a507'@localhost identified by '5moweB0!@7';
grant all privileges on moweb.* to 'user_a507'@'%' identified by '5moweB0!@7';

#9. 권한 확인
show grants for 'user_a507'@localhost;
show grants for 'user_a507'@'%';

#10. 시간 설정
set time_zone='Asia/Seoul';
set global time_zone='Asia/Seoul';

#11. 디비 변경사항 메모리에 반영
flush privileges;
```

5. nginx 설치

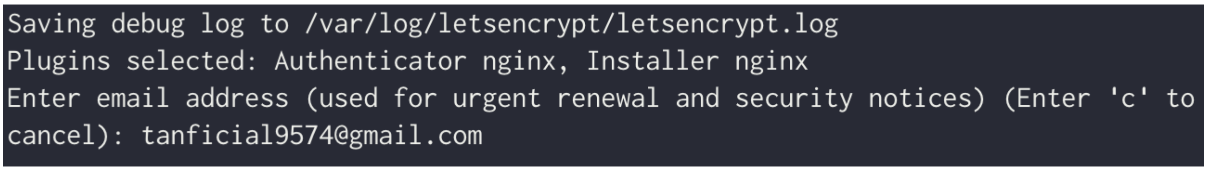
```
#1. apt udpate
sudo apt update

#2. nginx 설치
sudo apt install nginx

#3. nignx 시작
sudo service nginx start
```

6. letsencrypt certbot ssl 발급 및 https 설정

certbot 설치 및 실행

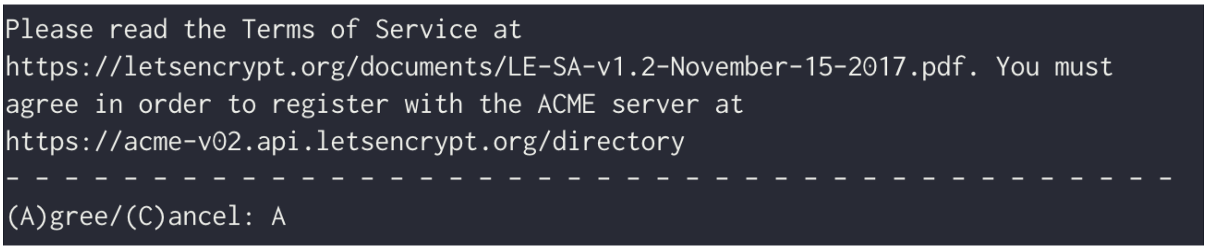
```
#1. apt update
sudo apt udpate

#2. cerbot, certbot nginx 플러그인 설치
sudo apt install certbot python3-cerbot-nginx

#3. nginx 설정에 서버 도메인네임 입력
#3-1. 설정 파일 열기
sudo vi /etc/nginx/sites-available

#3-2. 서버 도메인 네임 입력
server_name i7a507.p.ssafy.io;

#4. nginx 재시작
sudo service nginx restart

#5. certbot 실행
sudo certbot --nginx
```

이메일 입력



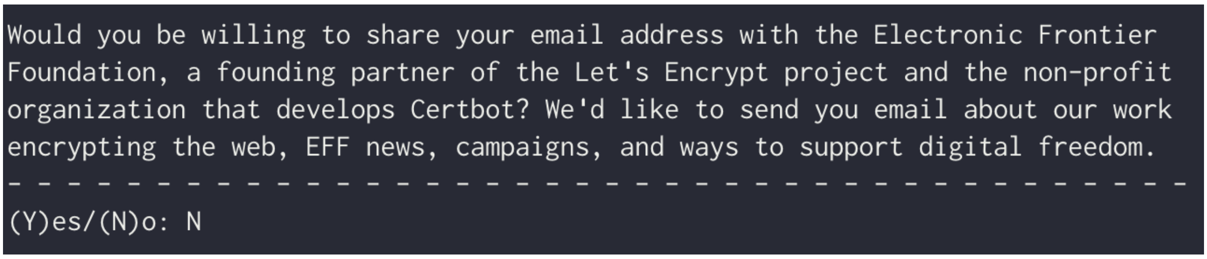
```
Saving debug log to /var/log/letsencrypt/letsencrypt.log
Plugins selected: Authenticator nginx, Installer nginx
Enter email address (used for urgent renewal and security notices) (Enter 'c' to
cancel): tanficial9574@gmail.com
```

서비스 이용 동의



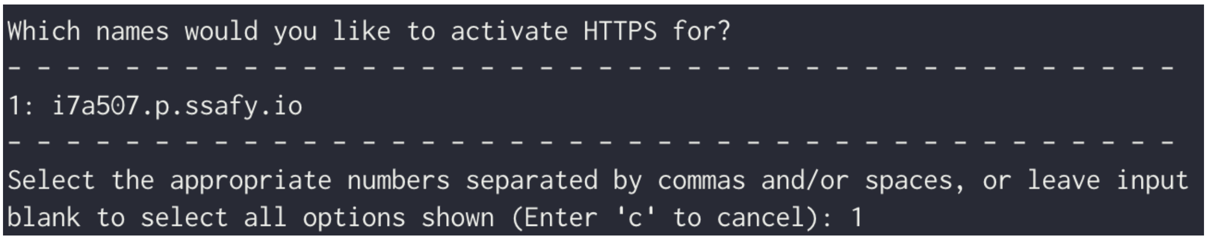
```
Please read the Terms of Service at
https://letsencrypt.org/documents/LE-SA-v1.2-November-15-2017.pdf. You must
agree in order to register with the ACME server at
https://acme-v02.api.letsencrypt.org/directory
- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
(A)gree/(C)ancel: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

이메일 수신 동의



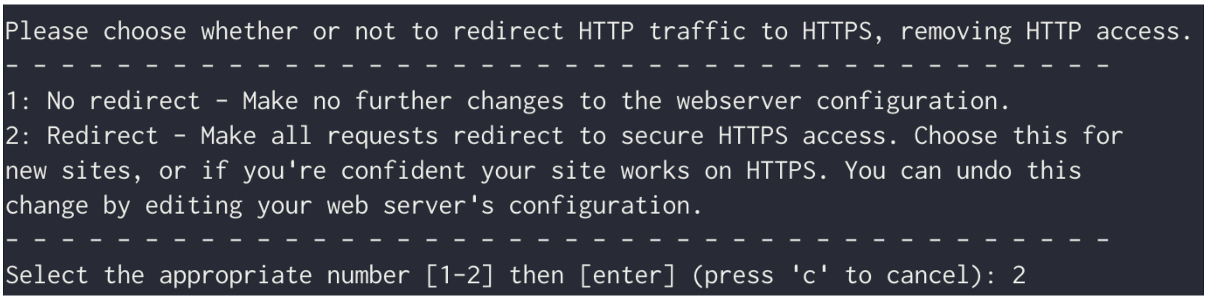
```
Would you be willing to share your email address with the Electronic Frontier
Foundation, a founding partner of the Let's Encrypt project and the non-profit
organization that develops Certbot? We'd like to send you email about our work
encrypting the web, EFF news, campaigns, and ways to support digital freedom.
- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
(Y)es/(N)o: N
```

발급받을 도메인 네임 선택 or 입력



```
Which names would you like to activate HTTPS for?
- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
1: i7a507.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): 1
```

nginx https redirect 자동설정



```
Please choose whether or not to redirect HTTP traffic to HTTPS, removing HTTP access.
- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
1: No redirect - Make no further changes to the webserver configuration.
2: Redirect - Make all requests redirect to secure HTTPS access. Choose this for
new sites, or if you're confident your site works on HTTPS. You can undo this
change by editing your web server's configuration.
- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
Select the appropriate number [1-2] then [enter] (press 'c' to cancel): 2
```

7. nginx 설정

설정 파일에 아래 프록시 설정 입력

```
        client_max_body_size 50M;

        location / {
                # First attempt to serve request as file, then
                # as directory, then fall back to displaying a 404.
                # try_files $uri $uri/ =404;
                ##
                proxy_pass http://localhost:8000;
                proxy_redirect off;
                charset utf-8;

                proxy_set_header X-Real-IP $remote_addr;
                proxy_set_header X-Forwarded-For $proxy_add_x_forwarded
_for;
                proxy_set_header X-Forwarded-Proto $scheme;
                proxy_set_header X-NginX-Proxy true;
                ##
        }
        location /moweb-api {
                # First attempt to serve request as file, then
                # as directory, then fall back to displaying a 404.
                # try_files $uri $uri/ =404;
                ##
                proxy_pass http://localhost:8080/moweb-api;
                proxy_redirect off;
                charset utf-8;

                proxy_set_header X-Real-IP $remote_addr;
                proxy_set_header X-Forwarded-For $proxy_add_x_forwarded
_for;
                proxy_set_header X-Forwarded-Proto $scheme;
                proxy_set_header X-NginX-Proxy true;

                proxy_set_header Upgrade $http_upgrade;
                proxy_set_header Connection "upgrade";
                ##
        }
```

nginx 재시작

```
# nginx 재시작
sudo service nginx restart
```

8. openvidu 설치, 설정, 실행

공식문서

- https://docs.openvidu.io/en/2.17.0/deployment/deploying-on-premises/

설치

```
#1. root 계정
sudo su

#2. opt 폴더로 이동
cd /opt

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

#4. root 접속 종료
exit

#5. openvidu 폴더로 이동
cd /openvidu

#6. openvidu 설정파일 수정
sudo vi .env
```

설정파일 수정할 내용

```
DOMAIN_OR_PUBLIC_IP=i7a507.p.ssafy.io

LETSENCRYPT_EMAIL=tanficial9574@gmail.com

CERTIFICATE_TYPE=letsencrypt

HTTPS_PORT=8443
```

시작

```
# openvidu 시작
./openvidu start
```

9. 프로젝트 git clone

```
git clone https://lab.ssafy.com/s07-webmobile1-sub2/S07P12A507.git
```

10. 환경변수 설정

환경변수 설명

11. docker compse 실행

```
# docker compose 실행
docker compse up -d --build
```

12. 전체 설정파일

1. nginx 설정파일

```
server {

        # SSL configuration
        #
        # listen 443 ssl default_server;
        # listen [::]:443 ssl default_server;
        #
        # Note: You should disable gzip for SSL traffic.
        # See: https://bugs.debian.org/773332
        #
        # Read up on ssl_ciphers to ensure a secure configuration.
        # See: https://bugs.debian.org/765782
        #
        # Self signed certs generated by the ssl-cert package
        # Don't use them in a production server!
        #
        # include snippets/snakeoil.conf;

        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 i7a507.p.ssafy.io;

        client_max_body_size 50M;

        location / {
                # First attempt to serve request as file, then
                # as directory, then fall back to displaying a 404.
                # try_files $uri $uri/ =404;
                ##
                proxy_pass http://localhost:8000;
                proxy_redirect off;
                charset utf-8;

                proxy_set_header X-Real-IP $remote_addr;
                proxy_set_header X-Forwarded-For $proxy_add_x_forwarded
_for;
                proxy_set_header X-Forwarded-Proto $scheme;
                proxy_set_header X-NginX-Proxy true;
                ##
        }
        location /moweb-api {
                # First attempt to serve request as file, then
                # as directory, then fall back to displaying a 404.
                # try_files $uri $uri/ =404;
                ##
                proxy_pass http://localhost:8080/moweb-api;
                proxy_redirect off;
                charset utf-8;

                proxy_set_header X-Real-IP $remote_addr;
                proxy_set_header X-Forwarded-For $proxy_add_x_forwarded
_for;
                proxy_set_header X-Forwarded-Proto $scheme;
                proxy_set_header X-NginX-Proxy true;

                proxy_set_header Upgrade $http_upgrade;
                proxy_set_header Connection "upgrade";
                ##
        }

        # pass PHP scripts to FastCGI server
        #
        #location ~ \.php$ {
        #       include snippets/fastcgi-php.conf;
        #
        #       # With php-fpm (or other unix sockets):
        #       fastcgi_pass unix:/var/run/php/php7.4-fpm.sock;
        #       # With php-cgi (or other tcp sockets):
        #       fastcgi_pass 127.0.0.1:9000;
        #}

        # deny access to .htaccess files, if Apache's document root
        # concurs with nginx's one
        #
        #location ~ /\.ht {
        #       deny all;
        #}

    listen [::]:443 ssl ipv6only=on; # managed by Certbot
    listen 443 ssl; # managed by Certbot
    ssl_certificate /etc/letsencrypt/live/i7a507.p.ssafy.io/fullchain.p
em; # managed by Certbot
    ssl_certificate_key /etc/letsencrypt/live/i7a507.p.ssafy.io/privke
y.pem; # managed by Certbot
    include /etc/letsencrypt/options-ssl-nginx.conf; # managed by Certb
ot
    ssl_dhparam /etc/letsencrypt/ssl-dhparams.pem; # managed by Certbot

}


# Virtual Host configuration for example.com
#
# You can move that to a different file under sites-available/ and syml
ink that
# to sites-enabled/ to enable it.
#
#server {
#       listen 80;
#       listen [::]:80;
#
#       server_name example.com;
#
#       root /var/www/example.com;
#       index index.html;
#
#       location / {
#               try_files $uri $uri/ =404;
#       }
#}

server {
    if ($host = i7a507.p.ssafy.io) {
        return 301 https://$host$request_uri;
    } # managed by Certbot
```

```
        listen 80 default_server;
        listen [::]:80 default_server;

        server_name i7a507.p.ssafy.io;
    return 404; # managed by Certbot


}
```

2. openvidu 설정파일

```
# OpenVidu configuration
# ----------------------
# Documentation: https://docs.openvidu.io/en/stable/reference-docs/openvidu-config/

# NOTE: This file doesn't need to quote assignment values, like most shells do.
# All values are stored as-is, even if they contain spaces, so don't quote them.

# Domain name. If you do not have one, the public IP of the machine.
# For example: 198.51.100.1, or openvidu.example.com
DOMAIN_OR_PUBLIC_IP=i7a507.p.ssafy.io

# OpenVidu SECRET used for apps to connect to OpenVidu server and users to access to OpenVidu Dashboard
OPENVIDU_SECRET=MY_SECRET

# Certificate type:
# - selfsigned:  Self signed certificate. Not recommended for production use.
#                Users will see an ERROR when connected to web page.
# - owncert:     Valid certificate purchased in a Internet services company.
#                Please put the certificates files inside folder ./owncert
#                with names certificate.key and certificate.cert
# - letsencrypt: Generate a new certificate using letsencrypt. Please set the
#                required contact email for Let's Encrypt in LETSENCRYPT_EMAIL
#                variable.
```

```
CERTIFICATE_TYPE=letsencrypt

# If CERTIFICATE_TYPE=letsencrypt, you need to configure a valid email for notifications
LETSENCRYPT_EMAIL=tanficial9574@gmail.com

# Proxy configuration
# If you want to change the ports on which openvidu listens, uncomment the following lines

# Allows any request to http://DOMAIN_OR_PUBLIC_IP:HTTP_PORT/ to be automatically
# redirected to https://DOMAIN_OR_PUBLIC_IP:HTTPS_PORT/.
# WARNING: the default port 80 cannot be changed during the first boot
# if you have chosen to deploy with the option CERTIFICATE_TYPE=letsencrypt
HTTP_PORT=4080

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

# Old paths are considered now deprecated, but still supported by default.
# OpenVidu Server will log a WARN message every time a deprecated path is called, indicating
# the new path that should be used instead. You can set property SUPPORT_DEPRECATED_API=false
# to stop allowing the use of old paths.
# Default value is true
# SUPPORT_DEPRECATED_API=true

# If true request to with www will be redirected to non-www requests
# Default value is false
# REDIRECT_WWW=false

# How many workers to configure in nginx proxy.
# The more workers, the more requests will be handled
# Default value is 10240
# WORKER_CONNECTIONS=10240

# Access restrictions
# In this section you will be able to restrict the IPs from which you can access to
# Openvidu API and the Administration Panel
# WARNING! If you touch this configuration you can lose access to the platform from some IPs.
# Use it carefully.
# This section limits access to the /dashboard (OpenVidu CE) and /inspector (OpenVidu Pro) pages.
# The form for a single IP or an IP range is:
# ALLOWED_ACCESS_TO_DASHBOARD=198.51.100.1 and ALLOWED_ACCESS_TO_DASHBOARD=198.51.100.0/24
# To limit multiple IPs or IP ranges, separate by commas like this:
# ALLOWED_ACCESS_TO_DASHBOARD=198.51.100.1, 198.51.100.0/24
# ALLOWED_ACCESS_TO_DASHBOARD=

# This section limits access to the Openvidu REST API.
# The form for a single IP or an IP range is:
# ALLOWED_ACCESS_TO_RESTAPI=198.51.100.1 and ALLOWED_ACCESS_TO_RESTAPI=198.51.100.0/24
# To limit multiple IPs or or IP ranges, separate by commas like this:
# ALLOWED_ACCESS_TO_RESTAPI=198.51.100.1, 198.51.100.0/24
# ALLOWED_ACCESS_TO_RESTAPI=

# Whether to enable recording module or not
OPENVIDU_RECORDING=false

# 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=false

# Which users should receive the recording events in the client side
# (recordingStarted, recordingStopped). Can be all (every user connected to
# the session), publisher_moderator (users with role 'PUBLISHER' or
# 'MODERATOR'), moderator (only users with role 'MODERATOR') or none
# (no user will receive these events)
OPENVIDU_RECORDING_NOTIFICATION=publisher_moderator

# Timeout in seconds for recordings to automatically stop (and the session involved to be closed)
# when conditions are met: a session recording is started but no user is publishing to it or a session
# is being recorded and last user disconnects. If a user publishes within the timeout in either case,
# the automatic stop of the recording is cancelled
# 0 means no timeout
OPENVIDU_RECORDING_AUTOSTOP_TIMEOUT=120

# Maximum video bandwidth sent from clients to OpenVidu Server, in kbps.
# 0 means unconstrained
OPENVIDU_STREAMS_VIDEO_MAX_RECV_BANDWIDTH=1000

# Minimum video bandwidth sent from clients to OpenVidu Server, in kbps.
# 0 means unconstrained
OPENVIDU_STREAMS_VIDEO_MIN_RECV_BANDWIDTH=300

# Maximum video bandwidth sent from OpenVidu Server to clients, in kbps.
# 0 means unconstrained
OPENVIDU_STREAMS_VIDEO_MAX_SEND_BANDWIDTH=1000

# Minimum video bandwidth sent from OpenVidu Server to clients, in kbps.
# 0 means unconstrained
OPENVIDU_STREAMS_VIDEO_MIN_SEND_BANDWIDTH=300

# All sessions of OpenVidu will try to force this codec. If OPENVIDU_STREAMS_ALLOW_TRANSCODING=true
# when a codec can not be forced, transcoding will be allowed
# Default value is VP8
# OPENVIDU_STREAMS_FORCED_VIDEO_CODEC=VP8

# Allow transcoding if codec specified in OPENVIDU_STREAMS_FORCED_VIDEO_CODEC can not be applied
# Default value is false
# OPENVIDU_STREAMS_ALLOW_TRANSCODING=false

# true to enable OpenVidu Webhook service. false' otherwise
# Values: true | false
OPENVIDU_WEBHOOK=false
```

```
# HTTP endpoint where OpenVidu Server will send Webhook HTTP POST messages
# Must be a valid URL: http(s)://ENDPOINT
#OPENVIDU_WEBHOOK_ENDPOINT=

# List of headers that OpenVidu Webhook service will attach to HTTP POST messages
#OPENVIDU_WEBHOOK_HEADERS=

# List of events that will be sent by OpenVidu Webhook service
# Default value is all available events
OPENVIDU_WEBHOOK_EVENTS=[sessionCreated,sessionDestroyed,participantJoined,participantLeft,webrtcConnectionCreated,webrtcConnectionDestroyed,recordingStatusChanged,filterEventDispatched,mediaNodeStatusChanged]

# How often the garbage collector of non active sessions runs.
# This helps cleaning up sessions that have been initialized through
# REST API (and maybe tokens have been created for them) but have had no users connected.
# Default to 900s (15 mins). 0 to disable non active sessions garbage collector
OPENVIDU_SESSIONS_GARBAGE_INTERVAL=900

# Minimum time in seconds that a non active session must have been in existence
# for the garbage collector of non active sessions to remove it. Default to 3600s (1 hour).
# If non active sessions garbage collector is disabled
# (property 'OPENVIDU_SESSIONS_GARBAGE_INTERVAL' to 0) this property is ignored
OPENVIDU_SESSIONS_GARBAGE_THRESHOLD=3600

# Call Detail Record enabled
# Whether to enable Call Detail Record or not
# Values: true | false
OPENVIDU_CDR=false

# Path where the cdr log files are hosted
OPENVIDU_CDR_PATH=/opt/openvidu/cdr

# Kurento Media Server image
# --------------------------
# Docker hub kurento media server: https://hub.docker.com/r/kurento/kurento-media-server
# Uncomment the next line and define this variable with KMS image that you want use
# KMS_IMAGE=kurento/kurento-media-server:6.16.0
# Kurento Media Server Level logs
# -------------------------------
# Uncomment the next line and define this variable to change
# the verbosity level of the logs of KMS
# Documentation: https://doc-kurento.readthedocs.io/en/stable/features/
logging.html
# KMS_DOCKER_ENV_GST_DEBUG=

# Openvidu Server Level logs
# --------------------------
# Uncomment the next line and define this variable to change
# the verbosity level of the logs of Openvidu Service
# RECOMENDED VALUES: INFO for normal logs DEBUG for more verbose logs
# OV_CE_DEBUG_LEVEL=INFO

# Java Options
# --------------------------
# Uncomment the next line and define this to add
# options to java command
# Documentation: https://docs.oracle.com/cd/E37116_01/install.111210/e2
3737/configuring_jvm.htm#OUDIG00058
# JAVA_OPTIONS=-Xms2048m -Xmx4096m -Duser.timezone=UTC
```

3. 환경변수

```
# 백엔드 api url
VUE_APP_MOWEB_API_URL = https://i7a507.p.ssafy.io/moweb-api

# 루트 서버 url
VUE_APP_ROOT_URL = https://i7a507.p.ssafy.io

# 오픈비두 url
VUE_APP_OPENVIDU_SERVER_URL = https://i7a507.p.ssafy.io:8443

# 카카오 api 키
VUE_APP_KAKAO_API_KEY = 59074e20c9d80e6e5200a4bd60122af7

# 오픈비두 시크릿 키
VUE_APP_OPENVIDU_SERVER_SECRET = MY_SECRET

# 백엔드 websocket url
VUE_APP_STOPM_SERVER_URL = https://i7a507.p.ssafy.io/moweb-api/ws/moweb

# 데이터베이스 이름
db_dbname = moweb
```

```
# 데이터베이스 패스워드
db_password = 5moweB0!@7

# 데이터베이스 url
db_url = localhost:3306

# 데이터베이스 아이디
db_username = user_a507

# swagger url
hosturl = i7a507.p.ssafy.io
```