

**D202 : 손말**

삼성SW청년아카데미 구미캠퍼스 7기

특화 프로젝트

2022/08/29 ~ 2022/10/07

**포팅 매뉴얼**

담당 컨설턴트 : 이승윤

김남희(팀장), 배시현, 배한용, 서재형, 정봉진, 편예린

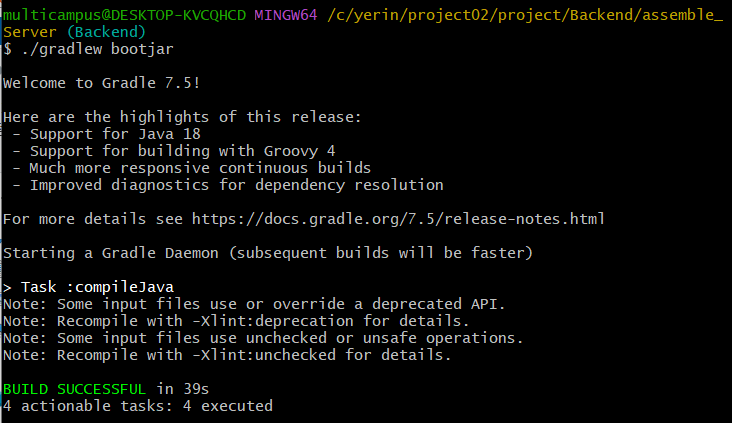
**목차**

1. 프로젝트 기술 스택…………………………………………………………….3
2. 빌드 상세내용……………………………………………………………………..4
3. 배포 특이사항……………………………………………………………………..7
4. DB 계정……………………………………………………………………………..10
5. 외부 서비스……………………………………………………………………….12
6. **프로젝트 기술 스택**
   * 형상관리 : Gitlab
   * 이슈관리 : Jira
   * 프로젝트관리 : Notion
   * 커뮤니케이션 : Mattermost, Webex
   * 디자인 : Figma
   * OS : Windows 10, 11
   * Database : MariaDB 3.0.7
   * language : Kotlin, Java11
   * Front-End (Android)
     + Android Studio Dolphin | 2021.3.1
     + Kotlin 1.7.20
     + JDK 11.0.13
     + Gradle 7.5
     + SDK
       1. min 21
       2. target 33
       3. compile 33
     + WebRTC: 1.0.32
     + Retrofit: 2.9.0
   * Back-End
     + JDK 11
     + Spring Boot 2.7.3
     + Gradle 7.5
     + Lombok, JPA, Swagger UI 3.0.0, Spring Security 5.7.3
     + Firebase 8.1.0
   * Server
     + AWS EC2
     + Ubuntu 20.04 LTS
     + Docker
     + Jenkins
     + CertBot
   * IDE
     + HeidiSQL 12.1.0
     + Android Studio Dolphin | 2021.3.1
     + IntelliJ IDEA | 2022.1.4
     + Spring Tool Suite 3.9.14
   * AI
     + PyTorch: 1.12
     + CudaToolkit: 11.6
     + TensorflowLite : 2.10.0
     + Jupyter
     + Anaconda3
7. **빌드 상세 내용**

**2-1. 백엔드 빌드 내용**

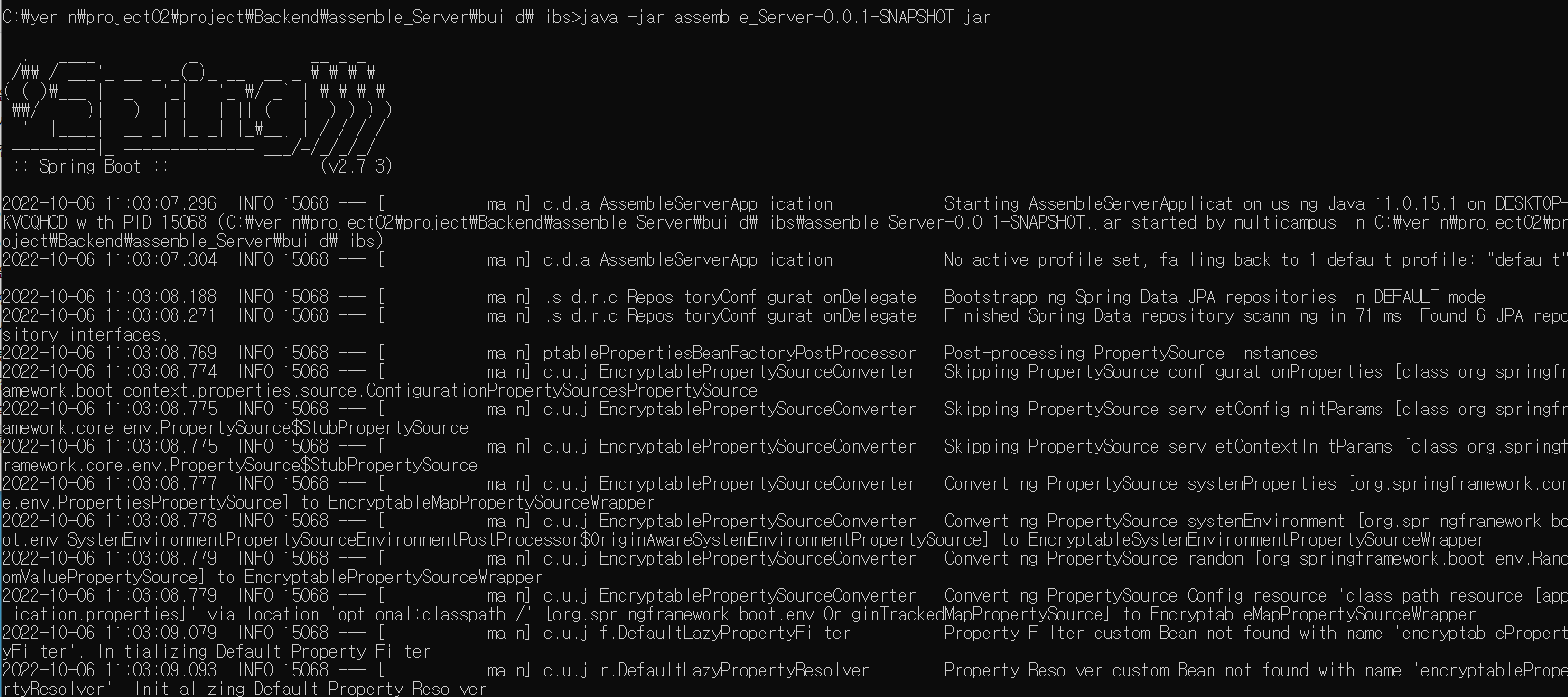
1. STS빌드하기

git bash창 열기 => Backend/assemble\_Server로 위치 이동 => **./gradlew bootjar**로 build시작



BUILD SUCCESS가 되면, 빌드가 성공적으로 완료된 것이며, Backend/assemble\_Server/build/libs 위치에 “assemble\_Server-0.0.1-SNAPSHOT.jar”파일이 생성됨을 확인할 수 있다.

실행 예시



2. IntelliJ 빌드하기

**2-2. 프론트엔드 빌드 내용**

카메라, 마이크 권한을 사용하기 때문에 **에뮬레이터가 아닌 휴대폰에서 사용**해야 한다.

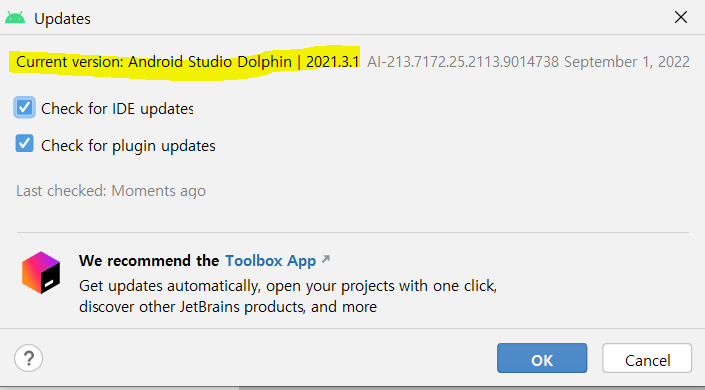
에뮬레이터를 사용할 경우, 빌드가 되더라도 앱이 강제 종료되며 기능들을 테스트할 수 없다.

빌드 과정은 다음과 같다.

1. Android Studio 설치 혹은 버전 확인 – Dolphin (2021.3.1)

|  |
| --- |
| Android Studio 버전 확인을 원할 경우,  Help – Check for Updates – 오른쪽 아래 팝업의 Update 파란색 글씨 클릭 – Configure 버튼 누르면 버전 확인 가능 |

예 :

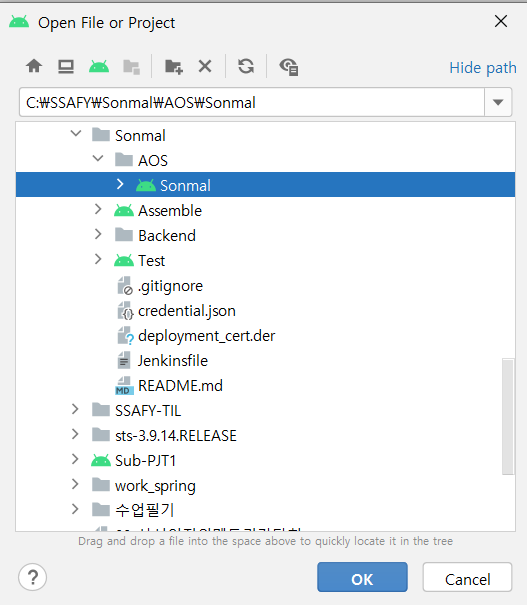


1. Kotlin 버전 확인 (N-1.7.20인지 확인)

|  |
| --- |
| File – Settings – Languages & Frameworks – Kotlin 클릭.  Current Kotlin plugin version 확인. |

1. 프로젝트 열기

|  |
| --- |
| File – Open 탭 클릭 후,  AOS 폴더 안, Sonmal(안드로이드 아이콘) 클릭 후에 OK 버튼 클릭. |



1. 프로젝트의 Gradle Version 확인

|  |
| --- |
| File – Project Structure 탭 클릭.  Project 탭에서 Gradle Version이 맞는지 확인. (Gradle Version : 7.5) |

1. Gradle JDK 확인

|  |
| --- |
| File – Settings – Build, Execution, Deployment – Build Tools – Gradle 클릭 후,  Gradle JDK가 version 11.0.13으로 되어 있는지 확인 |

1. 휴대폰과 Android Studio 연결을 위해, 휴대폰 개발자 모드 켜서 USB 디버깅 허용 시키기

휴대폰마다 방법이 다르므로, ‘휴대폰 기종 + 개발자 모드’를 검색 키워드로 검색해, 개발자 모드를 활성화 하고 USB 디버깅을 허용시켜준다.

예시 링크 : <https://learnandcreate.tistory.com/796>

1. 아래와 같이 휴대폰 연결을 확인하고, 초록색 play 버튼 클릭.



1. Build 성공 시, 휴대폰에 앱 첫번째 화면이 나오면서 설치를 확인할 수 있다!
2. **배포 특이사항**

**3-1. HTTPS 설정**

\*미리 도메인 등록&연결 해놓거나, SSAFY EC2 도메인 사용. 80포트는 열려 있어야 함

1) 80포트 열기

sudo ufw allow 80

sudo ufw status

2) Certbot 설치

sudo apt-get update

sudo snap install core; sudo snap refresh core

sudo snap install –classic certbot

sudo ln -s /snap/bin/certbot /usr/bin/certbot

3) Nginx 설치

sudo apt install nginx

4) Certbot 인증키 발급

sudo certbot –nginx –d [발급받은\_도메인주소]

5) nginx 설정

cd /etc/nginx/sites-available

sudo nano default



6) nginx 시작

sudo nginx

**3-2. 젠킨스 컨테이너 세팅**

1. Docker 설치

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](http://containerd.io/)

1. 젠킨스 설치 및 Docker volume연결

sudo docker run -d --name jenkins -u root --privileged \ -p '9090:8080' \ -v '/home/ubuntu/docker-volume/jenkins:/var/jenkins\_home' \ -v '/var/run/docker.sock:/var/run/docker.sock' \ -v '/usr/bin/docker:/usr/bin/docker' \ jenkins/jenkins

3) 젠킨스 접속

- 설정한 포트로 젠킨스 접속

- 초기 비밀번호 : sudo cat /var/lib/jenkins/secrets/initialAdminPassword

4) 젠킨스 설정

- 플러그인 설치 (Gitlab, SSH)

- 새로운 아이템 추가 > Pipeline

- 빌드 유발 : Gitlab webhook – 고급 – Secret Token에 깃랩 리포지토리에서 발급한 token 추가

- Definition을 Pipeline script from SCM 선택

- RepositoryURL , Credential에 Gitlab ID/PW 입력

- Branches to build= \*/Backend로 지정

- Script Path = Jenkinsfile

5) 서버 로그

sudo docker ps –a

sudo docker logs –f [backimg docker container id]

\*젠킨스로 빌드시 프로젝트 경로는

/home/ubuntu/docker-volume/jenkins/workspace/d202v2/Backend/assemble\_Server

**3-3. OpenVidu 세팅**

1. Docker, Docker Compose 설치
2. Openvidu 설치

sudo su

cd /opt

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

1. OpenVidu 환경 설정

cd openvidu

nano .env

|  |
| --- |
| DOMAIN\_OR\_PUBLIC\_IP=서버 도메인  OPENVIDU\_SECRET=암호  CERTIFICATE\_TYPE=letsencrypt  HTTPS\_PORT=443 |

1. Docker 설정

nano docker-compose.yml

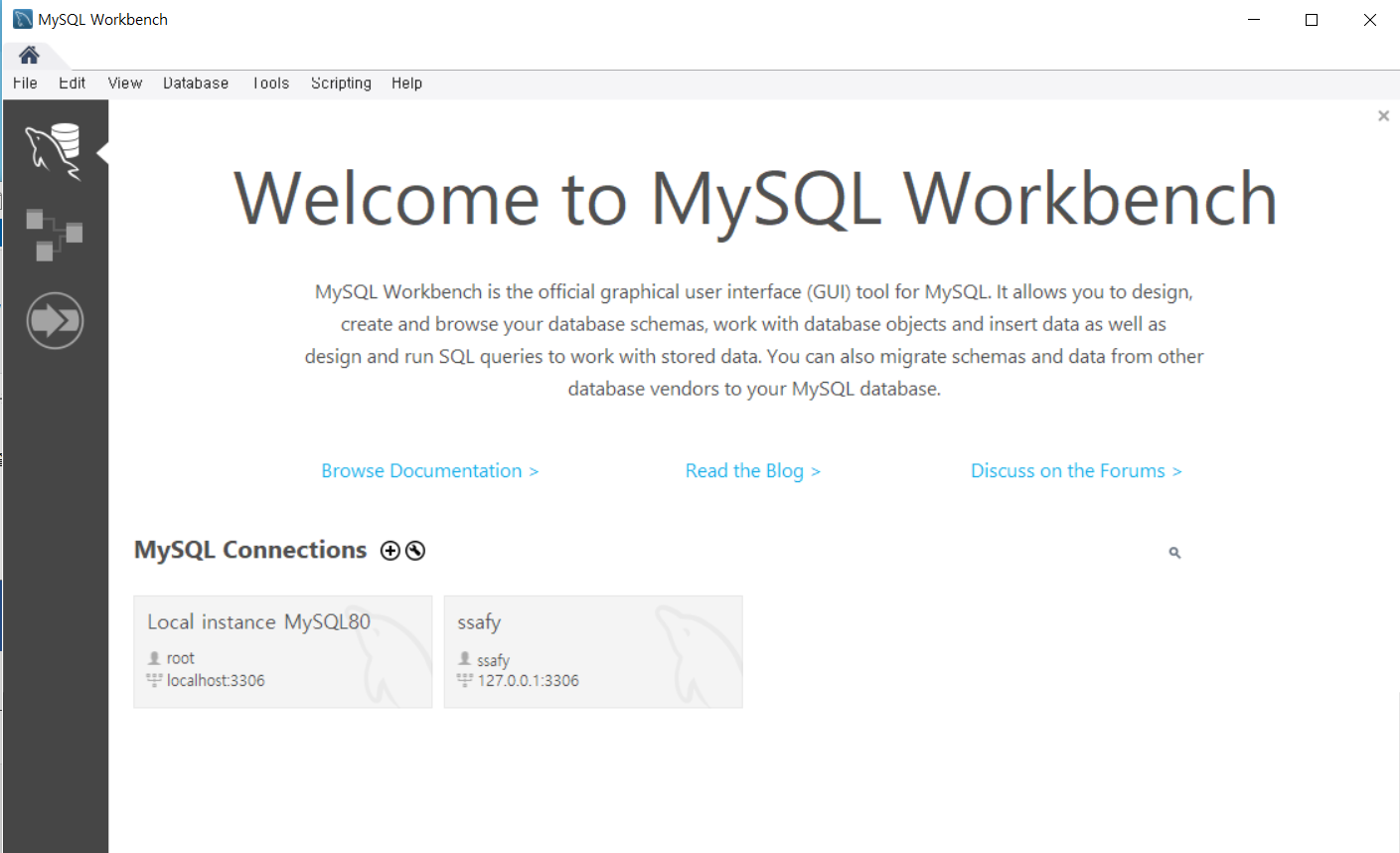
|  |
| --- |
| SERVER\_SSL\_ENABLED=true |

1. OpenVidu 시작

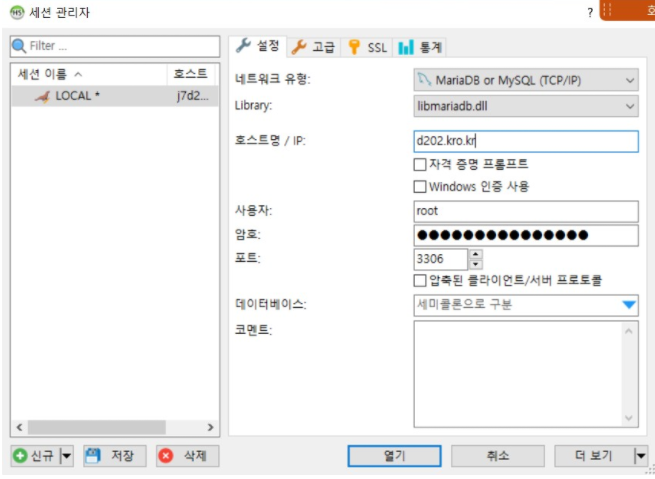
./openvidu start

1. **DB 계정**

**4-1. MariaDB 원격 접속**



* HeidiSQL을 실행합니다.
* MySQL Connection을 추가하기 위해 (+) 버튼을 클릭합니다.
  1. **Connection 설정**

****

* Hostname : d202.kro.kr / Port : 3306
* Username : root
* Password : kbbsjp855270#@!

1. **외부 서비스**

**5-1. Kakao Login 설정**

<https://developers.kakao.com/console/app>

- 애플리케이션 추가 후 네이티브 앱 키를 AndroidManifest.xml에 추가

- 키 해시 발급 후 Android플랫폼에 package 명과 함께 추가

keytool -exportcert -alias androiddebugkey -keystore %USERPROFILE%\.android\debug.keystore -storepass android -keypass android | openssl sha1 -binary | openssl base64

\*Release용 (구글 플레이스토어 등록 시 키 해시 추가해야 함)

keytool -exportcert -alias <RELEASE\_KEY\_ALIAS> -keystore <RELEASE\_KEY\_PATH> | openssl sha1 -binary | openssl base64

-팀 관리에서 카카오 로그인 이용할 이메일 추가

\* 플레이 스토어 등록 시 Google Play Console 앱 서명의 SHA-1 값을 입력

echo <SHA-1> | xxd -r -p | openssl base64

**5-2. Naver Login 설정**

<https://developers.naver.com/apps/#/list>

- 애플리케이션 등록

- 안드로이드 ApplicationClass NaverIdLoginSDK.initialize 에 ClientID, Client Secret 추가

- 멤버관리에서 네이버 로그인 이용할 아이디 추가