**1. 프로젝트 버전**

1. 웹 서버

a. Apache Tomcat : 9.0.76

b. Nginx : 1.18.0

2. 미디어 서버 : OpenVidu 2.28.0

3. 데이터 관리 서버 : Redis 7.0.12

4. 가상 서버 : Docker

a. Docker : 24.0.5

b. Docker Compose : 2.20.2

5. 배포 : Jenkins 2.417

6. JVM : 11.0.19

7. FE

a. NodeJS : 16.20.1

b. ReactJS : 17.0.2

8. BE : Spring Boot 2.7.13

9. DB : MySQL 8.0.34

**2. 환경 변수**

1. Port

a. Nginx : 80(http), 443(https)

b. FE : 3000

c. BE : 8000

d. MySQL : 3306

e. OpenVidu : 8002(http), 8003(https)

f. Redis : 6379

g. email : 587

2. Spring DB Setting

spring.datasource.hikari.driver-class-name=com.mysql.cj.jdbc.Driver  
spring.datasource.url=jdbc:mysql://localhost:3306/ourola?useUniCode=yes&characterEncoding=UTF-8

3. JPA

spring.jpa.hibernate.ddl-auto=validate  
spring.jpa.database=*mysql*spring.jpa.show-sql=true  
spring.jpa.database-platform=org.hibernate.dialect.MySQL8Dialect  
spring.jpa.properties.hibernate.format\_sql=true  
spring.jpa.properties.hibernate.globally\_quoted\_identifiers=true

4. JWT

jwt.secretKey=JWT 시크릿 키 값  
# 1 hour  
jwt.access.expiration=3600000  
jwt.access.header=Authorization  
# 2 week  
jwt.refresh.expiration 1209600000  
jwt.refresh.header=Authorization-refresh

5. File  
spring.servlet.multipart.max-file-size=50MB  
spring.servlet.multipart.max-request-size=50MB  
spring.servlet.multipart.location=/app/file  
file.Feed=/feedFile/  
file.group=/groupImg/  
file.profile=/profileImg/  
file.shopFile=/shopFile/  
file.shopMainFile=/shopMainFile/  
file.openLiveFile=/openLiveFile/

6. OpenVidu

server.ssl.enabled:false  
OPENVIDU\_URL:https://i9d204.p.ssafy.io:8003/  
OPENVIDU\_SECRET: OpenVidu 시크릿 키 값

7. Email

spring.mail.host=smtp.gmail.com  
spring.mail.port=587  
spring.mail.username=서버 측 이메일   
spring.mail.password=서버 측 비밀번호   
spring.mail.properties.mail.smtp.starttls.enable=true  
spring.mail.properties.mail.smtp.auth=true  
spring.profiles.group."local"="local, jwt, oauth"  
spring.profiles.active=local  
spring.profiles.include=oauth

8. Redis

spring.redis.host=i9d204.p.ssafy.io  
spring.redis.port=6379

9. Spring Security

spring.security.oauth2.client.registration.google.client-id=구글 id   
spring.security.oauth2.client.registration.google.client-secret=구글 pw   
spring.security.oauth2.client.registration.google.scope=profile, email  
spring.security.oauth2.client.registration.google.redirect-uri=구글 리다이렉트 URI  
spring.security.oauth2.client.registration.kakao.client-id=카카오 id  
spring.security.oauth2.client.registration.kakao.client-secret=카카오 pw  
spring.security.oauth2.client.registration.kakao.redirect-uri=카카오 리다이렉트 URI   
spring.security.oauth2.client.registration.kakao.client-authentication-method=POST  
spring.security.oauth2.client.registration.kakao.authorization-grant-type=카카오 인증코드   
spring.security.oauth2.client.registration.kakao.scope=profile\_nickname  
spring.security.oauth2.client.registration.kakao.client-name=Kakao  
spring.security.oauth2.client.provider.kakao.authorization-uri= 카카오 인증 URI  
spring.security.oauth2.client.provider.kakao.token-uri=카카오 토큰 URI spring.security.oauth2.client.provider.kakao.user-info-uri=카카오 유저정보 URI spring.security.oauth2.client.provider.kakao.user-name-attribute=id

**3. 배포**

1. Server Name : i9d204.p.ssafy.io

2. Jenkins

|  |
| --- |
| version: '3'  services:  jenkins:  container\_name: jenkins  image: jenkins/jenkins  restart: on-failure  ports: - "8080:8080"  volumes:  - /var/jenkins\_home:/var/jenkins\_home  - /var/run/docker.sock:/var/run/docker.sock  - /usr/bin/docker:/usr/bin/docker  - /var/www/backend:/var/www/backend  - /var/www/frontend:/var/www/frontend user: root |

**4. 주요 접속 계정**

1. DB

Id : ssafy

Pw : ssafy

2. 웹 사이트

Id : [test123@gmail.com](mailto:test123@gmail.com)

Pw : test123