

1. GitLab 소스 클론 이후 빌드 및 배포 매뉴얼

1-1. 개발 환경

항목	내용
GitLab Repo	https://lab.ssafy.com/s13-webmobile1-sub1/S13P11E205
Backend 디렉토리	Stalk_BE
Frontend 디렉토리	Stalk_FE
IDE	IntelliJ IDEA 2023.3.8, VSCode(최신 버전)
JDK/JVM	OpenJDK 17.0.15
Gradle	8.x (Wrapper 사용)
Node.js	v22.x ~ v23.x
OS/배포 대상	AWS EC2 Ubuntu 22.04 LTS
형상관리	GitLab

1-2. 서버 구성

구성요소	버전	비고
Nginx	최신 LTS 안정 버전 (예: 1.24.x)	Reverse Proxy, SSL 인증서 적용
Spring Boot	3.x (Java 17 호환)	WAS 포함
OpenVidu Server	2.30.0	Docker Compose 기반 배포
Redis	7.x	Refresh Token 저장소
MongoDB	추후 업데이트 예정	
MySQL	8.0.x	InnoDB, utf8mb4

1-3. 포트 매펑

환경	FE 포트	BE 포트	비고
Dev	3001	8081	
Prod	3000	8082	
OpenVidu -		4443 (WSS) TURN/STUN 포트 포함	

1-4. 빌드 환경 변수 / 프로퍼티

Backend (application.properties)

- DB 접속 정보

ini

복사편집

```
spring.datasource.url=jdbc:mysql://host.docker.internal:3306/stalk_db
?serverTimezone=Asia/Seoul&characterEncoding=UTF-8
spring.datasource.username=root
spring.datasource.password=****
```

- OpenVidu

ini

복사편집

```
openvidu.url=https://i13e205.p.ssafy.io:8443
openvidu.secret=****
```

- Redis

ini

복사편집

```
spring.data.redis.host=i13e205.p.ssafy.io
spring.data.redis.port=6379
spring.data.redis.password=****
```

- JWT

ini

복사편집

```
jwt.secret=****
jwt.access-token-validity=900000
jwt.refresh-token-validity=604800000
```

- 외부 API

- KRX API: krx.api.key=****
- KIS API: kis.api.app-key, kis.api.app-secret
- Toss Payments: payment.toss.test_client_api_key,
payment.toss.test_secret_api_key
- GCP/Vertex AI: gcp.project.id, gcp.credentials.location,
gcp.storage.bucket.name

Frontend (.env)

- API Base URL

```
ini  
복사편집  
VITE_API_BASE_URL=https://i13e205.p.ssafy.io/api  
VITE_OPENVIDU_URL=https://i13e205.p.ssafy.io:4443
```

- 외부 SDK/API 키: Kakao Map API Key 등
-

1-5. 빌드 방법

Backend

```
bash  
복사편집  
# GitLab 소스 클론  
git clone https://lab.ssafy.com/s13-webmobile1-sub1/S13P11E205.git  
cd S13P11E205/Stalk_BE  
  
# Gradle 빌드 (Wrapper 사용)  
.gradlew clean build -x test  
  
# Docker 이미지 빌드 및 실행  
docker build -t stalk-be:latest .  
docker run -d --name stalk-be ¶  
--env-file .env ¶  
-p 8081:8081 stalk-be:latest
```

Frontend

```
bash  
복사편집  
cd ../Stalk_FE  
  
# 패키지 설치  
npm install  
  
# 빌드  
npm run build  
  
# 배포 (예: Nginx / Docker)  
docker build -t stalk-fe:latest .  
docker run -d --name stalk-fe ¶  
-p 3000:80 stalk-fe:latest
```

1-6. 배포 특이사항

- Nginx
 - Reverse Proxy 설정 (FE/BE, WebSocket 포함)
 - SSL 인증서(Let's Encrypt) 적용
 - Jenkins
 - 브랜치별 Job 분리(dev → 테스트 서버, master → 운영 서버)
 - Jenkins Secret File을 사용하여 application.properties, .env 주입
 - Docker Compose
 - docker-compose.dev.yml, docker-compose.prod.yml 분리
 - OpenVidu, MySQL, Redis, BE, FE 통합 관리
-

1-7. 주요 프로퍼티/계정 정의 파일 목록

- Stalk_BE/src/main/resources/(배포용) application.properties
- Stalk_FE/.env.production
- nginx.conf (dev/prod)
- docker-compose.{dev,prod}.yml
- Jenkinsfile (FE, BE)