**포팅 메뉴얼**

**프로젝트 개발 환경**

* SpringBoot 3.5.4
* Gradle 8.14.3
* Java 17 LTS
* **Python 3.10.11**
* Openvidu 설치

<https://openvidu.io/latest/docs/self-hosting/single-node/on-premises/install/>

openvidu 443 port -> 8443 port로 변경

* Gitignore 처리한 핵심 키들

React : .env (최상단 위치)

SpringBoot : .env (최상단 위치), application.propertise (\src\main\resource에 위치)

FastAPI: .env (최상단 위치)

### Front-End

.env

REACT\_APP\_NAME=Malmoon

REACT\_APP\_VERSION=1.0.0

.env.development

REACT\_APP\_API\_URL=http://localhost:8080/api/v1

.env.production

REACT\_APP\_API\_URL=https://www.malmoon.store/api/v1

### A**pplication.properties**

spring.application.name=malmoon

# ========= DATABASE =========

spring.datasource.url=${DB\_URL}

spring.datasource.username=${DB\_USERNAME}

spring.datasource.password=${DB\_PASSWORD}

spring.datasource.driver-class-name=org.postgresql.Driver

# ========= S3 =========

cloud.aws.credentials.access-key=${AWS\_ACCESS\_KEY}

cloud.aws.credentials.secret-key=${AWS\_SECRET\_KEY}

cloud.aws.region.static=${AWS\_REGION}

cloud.aws.s3.bucket=${AWS\_S3\_BUCKET}

cloud.aws.s3.url-prefix=${AWS\_S3\_URL\_PREFIX}

# ========= JWT =========

jwt.secret=${JWT\_SECRET}

jwt.expiration=${JWT\_EXPIRATION}

# ========= Fast API =========

external.fastapi.url=${FASTAPI\_URL}

# ========= Openvidu(Livekit) =========

livekit.host=${LIVEKIT\_HOST}

livekit.api.key=${LIVEKIT\_API\_KEY}

livekit.api.secret=${LIVEKIT\_API\_SECRET}

# ========= Redis =========

spring.data.redis.host=${REDIS\_HOST}

spring.data.redis.port=${REDIS\_PORT}

# ========= Filter Lens API Token =========

filter.lens.api.token=${FILTER\_LENS\_API\_TOKEN}

# ========= 테이블 업데이트 시에 추가 =========

spring.jpa.hibernate.ddl-auto=update

spring.jpa.show-sql=false

spring.jpa.properties.hibernate.format\_sql=false

# ========= swagger =========

springdoc.swagger-ui.path=/swagger-ui.html

springdoc.swagger-ui.tags-sorter=alpha

springdoc.swagger-ui.operations-sorter=alpha

springdoc.api-docs.path=/api/v1

springdoc.default-consumes-media-type=application/json;charset=UTF-8

springdoc.default-produces-media-type=application/json;charset=UTF-8

spring.servlet.multipart.max-file-size=10MB

spring.servlet.multipart.max-request-size=20MB

spring.jackson.serialization.write-dates-as-timestamps=false

# ========= Pre-Signed 업로드용 추가 =========

cloud.aws.s3.presign-exp-seconds=1200

cloud.aws.s3.key-prefix=files

# ========= 업로드 검증(서버측) =========

file.max-size-bytes=20971520

file.allowed-content-types=image/jpeg,image/png,image/webp,image/avif,image/gif,application/pdf,video/mp4,audio/mpeg,audio/wav

### .env

* root 디렉토리에 생성, backend 폴더 안에

# ========= DB =========

#DB\_HOST=172.17.0.1

DB\_URL=jdbc:postgresql://도메인또는localhost:5432/malmoon

DB\_USERNAME=malmoon\_dev

DB\_PASSWORD=communet

# ========= AWS S3 =========

AWS\_ACCESS\_KEY=

AWS\_SECRET\_KEY=

AWS\_REGION=ap-northeast-2

AWS\_S3\_BUCKET=malmoon-file-bucket-dev

AWS\_S3\_URL\_PREFIX=https://

# ========= JWT =========

JWT\_SECRET=

# ========= Fast API =========

FASTAPI\_URL=http://localhost:8000

JWT\_EXPIRATION=46800000

# ========= Openvidu =========

LIVEKIT\_HOST=http://i13c107.p.ssafy.io:7880/

LIVEKIT\_API\_KEY=

LIVEKIT\_API\_SECRET=

# ========= Redis =========

REDIS\_HOST=127.0.0.1

REDIS\_PORT=6379

# ========= Filter Lens Api Token =========

FILTER\_LENS\_API\_TOKEN=

FastAPI

### .env

# AWS S3 설정

AWS\_ACCESS\_KEY=

AWS\_SECRET\_KEY=

AWS\_REGION=ap-northeast-2

AWS\_S3\_BUCKET=

AWS\_S3\_URL\_PREFIX=https://

# GMS API

GMS\_API\_KEY=

GEMINI\_IMAGE\_API\_URL=

GMS\_FEEDBACK\_URL=

GMS\_STT\_URL=

* 포트

TCP 80 HTTP

TCP 443 HTTPS

TCP 1935 Needed if you want to ingest RTMP streams using Ingress service. openVidu

TCP 3000 React ours

TCP 5432 PostgreSQL ours

TCP 6379 Redis ours

TCP 7881 Needed if you want to allow WebRTC over TCP. openVidu

UDP 7885 Needed if you want to ingest WebRTC using WHIP protocol. openVidu

TCP 8000 FastAPI ours

TCP 8443 Allows access to the following: LiveKit API OpenVidu Dashboard OpenVidu Call (Default Application) WHIP API TURN with TLS Custom layouts openVidu

TCP 8080 Spring Boot ours

TCP Jenkins ours

TCP 9000 Needed if you want to expose MinIO publicly. openVidu

UDP 50000 - 60000 WebRTC Media traffic openVidu

로컬 또는 빌드 환경에서

docker compose build

docker compose push

로 dockerhub에 push하고

Ec2 환경에서

docker compose pull

docker compose up -d