빌드 및 배포 과정 문서

SSAFY 11기 C208 종달새

# 버전 정보

|  |  |
| --- | --- |
| jdk | Liberica 17.0.2+ LTS |
| nginx | 1.27.0 |
| redis | 7.4.0 |
| mysql | 8.4.1 |
| openVidu | 3.0.0-beta2 |
| IntelliJ | 2024.1.4 |

# 빌드 환경변수 정보

Frontend .env

|  |
| --- |
| REACT\_APP\_KakaoMap\_api=[]  REACT\_APP\_OPENAI\_API\_KEY=[]  REACT\_APP\_YOUTUBE\_API\_KEY=[]  REACT\_APP\_URL=https://i11c208.p.ssafy.io  REACT\_APP\_WEB\_STOMP=wss://i11c208.p.ssafy.io/tarotbom/ws-stomp |

Backend .env

|  |
| --- |
| * S3\_BUCKET\_NAME=joosas-very-awesome-bucket * S3\_ACCESS\_KEY=[] * S3\_SECRET\_KEY=[] * S3\_CLOUD\_FRONT\_DOMAIN=d2ovihsqke74ur.cloudfront.net * SPRING\_DATASOURCE\_URL=jdbc:mysql://mysql:3306/tarotbom?useSSL=false&allowPublicKeyRetrieval=true&useUnicode=true&serverTimezone=Asia/S> * SPRING\_DATASOURCE\_USERNAME=jong * SPRING\_DATASOURCE\_PASSWORD=[] * SPRING\_DATA\_REDIS\_HOST=myredis * SPRING\_DATA\_REDIS\_PORT=6379 * SPRING\_DATA\_REDIS\_PASSWORD=[] * LIVEKIT\_API\_KEY=[] LIVEKIT\_API\_SECRET=[] |

# Dockerfile

Frontend Dockerfile.prod

|  |
| --- |
| GNU nano 4.8 Dockerfile.prod  FROM node:20.15.0 AS build  WORKDIR /app  COPY package.json .  RUN npm install  COPY . .  COPY .env ./  RUN export $(cat .env | xargs)  RUN npm run build  FROM nginx  COPY --from=build /app/build /usr/share/nginx/html |

Backend Dockerfile

|  |
| --- |
| FROM bellsoft/liberica-openjdk-debian:17.0.12 AS builder  COPY gradlew .  COPY gradle gradle  COPY build.gradle .  COPY settings.gradle .  COPY src src  RUN chmod +x ./gradlew  RUN ./gradlew bootJar -x test  # build end  FROM bellsoft/liberica-openjdk-debian:17.0.12  COPY --from=builder build/libs/\*.jar app.jar  COPY .env ./  RUN export $(cat .env | xargs)  ENTRYPOINT ["java","-jar","/app.jar"] |

# Compose.yml

|  |
| --- |
| services:  mysql:  container\_name: mysql  environment:  - MYSQL\_ROOT\_PASSWORD=${MYSQL\_ROOT\_PASSWORD}  - MYSQL\_DATABASE=${MYSQL\_DATABASE}  - MYSQL\_USER=${MYSQL\_USER}  - MYSQL\_PASSWORD=${MYSQL\_PASSWORD}  - LC\_ALL=C.UTF-8  ports:  - '127.0.0.1:3306:3306'  volumes:  - './mysql\_data:/var/lib/mysql'  image: 'mysql:8.4.1'  redis:  container\_name: myredis  volumes:  - './myredis/conf/redis.conf:/etc/redis/redis.conf'  - './myredis/redis\_data:/data'  command: 'redis-server /etc/redis/redis.conf --save 60 1 --loglevel warning'  image: 'redis:latest'  ports:  - '6380:6379'  hostname: myredis  tarotbom-server:  container\_name: server  hostname: server  build:  context: ./S11P12C208/backend  dockerfile: Dockerfile  ports:  - '18080:80'  depends\_on:  - mysql  environment:  - SPRING\_DATASOURCE\_URL=${SPRING\_DATASOURCE\_URL}  - SPRING\_DATASOURCE\_USERNAME=${SPRING\_DATASOURCE\_USERNAME}  - SPRING\_DATASOURCE\_PASSWORD=${SPRING\_DATASOURCE\_PASSWORD}  - SPRING\_DATA\_REDIS\_HOST=${SPRING\_DATA\_REDIS\_HOST}  - SPRING\_DATA\_REDIS\_PORT=${SPRING\_DATA\_REDIS\_PORT}  tarotbom-client:  container\_name: client  hostname: client  build:  context: ./S11P12C208/frontend  dockerfile: Dockerfile.prod  args:  - REACT\_APP\_NAME=tarotbom  - REACT\_APP\_TITLE=타로봄  ports:  - '3010:80' |

# Compose용 .env

|  |
| --- |
| MYSQL\_ROOT\_PASSWORD=[]  MYSQL\_DATABASE=tarotbom  MYSQL\_USER=jong  MYSQL\_PASSWORD=[]  SPRING\_DATASOURCE\_URL=jdbc:mysql://mysql:3306/tarotbom?useSSL=false&allowPublicKeyRetrieval=true&useUnicode=true&serverTimezone=Asia/S>  SPRING\_DATASOURCE\_USERNAME=jong  SPRING\_DATASOURCE\_PASSWORD=[]  SPRING\_DATA\_REDIS\_HOST=myredis  SPRING\_DATA\_REDIS\_PORT=6379 |

# 기타 특이사항

openVidu v3 내에 포함된 caddy.yml을 이하와 같이 수정(Caddy를 이용한 리벅스 프록시를 위함)

|  |
| --- |
| .. (중략) …  # Grafana  - handle:  - handler: subroute  routes:  - handle:  - handler: reverse\_proxy  upstreams:  - dial: "localhost:3000"  match:  - path:  - /grafana/\*  - /grafana  # Tarotbom Backend  - handle:  - handler: subroute  routes:  - handle:  - handler: reverse\_proxy  upstreams:  - dial: "localhost:18080"  match:  - path:  - /tarotbom/\*  - /tarotbom  # Tarotbom Frontend  - handle:  - handler: subroute  routes:  - handle:  - handler: reverse\_proxy  upstreams:  - dial: "localhost:3010"  match:  - path:  - /\*  … (중략) … |