**배포 conf 파일 정리**

**Dockerfile**

* Backend dockerfile 보기
* # 1. OpenJDK 이미지를 사용하여 기반 이미지 설정
* FROM openjdk:17-jdk-alpine as build
* # 2. 작업 디렉토리 설정
* WORKDIR /workspace/app
* # 3. 의존성 파일만 먼저 복사하여 캐시 이점 활용
* COPY gradlew .
* COPY gradle gradle
* COPY build.gradle .
* COPY settings.gradle .
* # 4. 프로젝트 내 모든 소스 코드 복사
* COPY src src
* # 5. 실행 권한 부여 및 프로젝트 빌드
* # gradlew에 실행 권한이 없는 경우 아래 RUN 명령을 통해 부여
* RUN chmod +x ./gradlew
* RUN ./gradlew build -x test
* # 6. 애플리케이션 실행을 위한 새로운 단계
* FROM openjdk:17-jdk-alpine
* WORKDIR /app
* EXPOSE 8080
* # 빌드 단계에서 생성된 JAR 파일 복사
* COPY --from=build /workspace/app/build/libs/\*.jar app.jar
* # 애플리케이션 실행
* ENTRYPOINT ["java","-jar","app.jar"]
* Frontend dockerfile 보기
* # Node.js 버전을 20.11.0로 설정
* FROM node:20.11.0
* # 작업 디렉토리 설정
* WORKDIR /user/src/app
* # package.json 및 package-lock.json (있는 경우) 복사
* COPY package\*.json ./
* # npm 대신 npm@10.2.4 버전을 명시적으로 설치
* RUN npm install -g npm@10.2.4
* # 프로젝트 의존성 설치
* RUN npm install
* # 나머지 프로젝트 파일 복사
* COPY ./ ./
* # EXPOSE 5173
* # 애플리케이션 실행
* CMD ["npm", "run", "dev", "--", "--host"]

**Nginx 설정 파일**

* etc/nginx/sites-available
* 설정 파일 보기