

Porting Manual

목차

1. 개발 환경
2. Properties
3. 빌드 및 배포

1. 개발 환경

Frontend

- React 18.2.0
- React-query 5.27.5
- node.js npm 10.2.5
 - Vite 5.1.3
- Typescript 5.2.2
- Axios 1.6.5
- Styled Components 6.1.8
- Zustand 4.5.2

Backend

- OpenJDK 17.0.9
- SpringBoot 3.2.3
- JPA (Hibernate)
- Gradle 8.5

Infra

- Docker 25.0.4
- MySQL 8.0.34
- Nginx Proxy Manger 2.11.1
- Jenkins 2.448

AI

- PyTorch 2.2.1
- GCP

2. Properties

Backend

- `db` : database 관련 주소
 - `host` : Database host 서버 주소 및 포트
 - `username` : Database에 로그인 하기 위한 user 이름
 - `password` : Database user 이름에 해당하는 password
- `deepl.api-key` : deepl 번역기 api 호출을 위한 key
- `oci.object-storage` : OCI Object Storage 관련 환경 변수
 - `bucket-name` : 버킷 명
 - `bucket-name-space` : 버킷 namespace
 - `region-url` : region 주소
 - `read-only-url` : Object Storage 읽기 전용 주소
- `gemini.api-key` : Gemini api 호출을 위한 key

AI

- GCP Key : GCP 환경에 배포할 key

3. 빌드 및 배포

배포 서버 구성

1. Docker 설치
2. Nginx Proxy Manger

- a. Pull Docker Image

```
jc21/nginx-proxy-manager:latest
```

- b. Run Docker Image

```
docker run jc21/nginx-proxy-manager:latest -d (+옵션 값)
```

- c. 배포 서버 주소 81번 port 접속
- d. 할당할 Domain을 Proxy Host에 등록
- e. SSL 인증서 발급

3. MySQL

- a. Pull Docker Image

```
docker pull mysql:latest
```

- b. Run Docker Image

```
docker run mysql:latest -d (+옵션 값)
```

- c. MySQL root 권한 계정 생성

- i. `docker exec -itu 0 mysql:latest bash`

- ii. `mysql -u root -p (비밀번호)`

- iii. root 권한의 계정 생성

- d. Database, Table 생성

4. Jenkins

a. Pull Docker Image

```
docker pull jenkins:jdk17
```

b. Run Docker Image

```
docker run jenkins:jdk17 -d (+옵션 값)
```

c. 최초 Unlock Jenkins 후 CI/CD Pipeline 작성

Frontend

1. 의존성 패키지 설치

```
npm install
```

2. 프로젝트 빌드

```
npm run build
```

3. repository 내에 있는 Dockerfile을 활용하여 Docker Image 빌드

4. Dockerhub Push

Backend

1. 프로젝트 빌드

```
./gradlew clean build
```

2. repository 내에 있는 Dockerfile을 활용하여 Docker Image 빌드

3. Dockerhub Push

AI

1. 의존성 라이브러리 설치 (pip3 install)

2. repository 내에 있는 Dockerfile을 활용하여 Docker Image 빌드

3. GCP Credential 값 설정

4. Dockerhub Push