exec

Exec

1. **사용한 JVM, 웹서버, WAS 제품 등의 종류와 설정값, 버전(IDE버전 포함) 기재**

[백엔드 & 데이터]

<Spring Boot>

- **JVM : openjdk:8-jdk**
- **IntelliJ : 2022.2.1**

<Django>

- **VSCode : 1.70.2**
- **Python : 3.9.10**

<Database>

- **DB : MySQL Server 5.7**

<배포>

- **Docker : 20.10.18**
- **Jenkins : 2.371**

[프론트]

- **React : 18.2.0**
- **TypeScript : 4.8.2**
- **d3 : 7.6.1**
- **react-router : 6.3.0**
- **redux-toolkit : 1.8.5**
- **ide : vscode**

2. 빌드 시 사용되는 환경 변수 등의 주요 내용 상세 기재

- 리액트

```
FROM node:16.15.0 as build-stage
WORKDIR /var/jenkins_home/workspace/BeTravelic/Front/be-travelic
COPY package*.json ./
RUN npm install
COPY . .
RUN npm run build


EXPOSE 3000
CMD npm start --port 3000
```

- 스프링 부트

```
FROM openjdk:8-jdk-alpine
ARG JAR_FILE=build/libs/demo-0.0.1-SNAPSHOT.jar
COPY ${JAR_FILE} demo.jar
ENTRYPOINT ["java","-jar","/demo.jar"]
```

- 장고

```
FROM python:3.9.10
WORKDIR /var/jenkins_home/workspace/BeTravelic/backend
COPY requirements.txt ./

RUN pip3 install --upgrade pip
RUN pip3 install --upgrade setuptools
RUN pip3 install django
ENV JAVA_HOME /usr/lib/jvm/java-1.7-openjdk/jre
RUN apt-get update && apt-get install -y g++ default-jdk
RUN pip3 install konlpy

RUN pip3 install -r requirements.txt
COPY . .
CMD ["python3", "manage.py", "runserver", "0.0.0.0:8081"]
EXPOSE 8081
```

3. **배포 시 특이사항 기재**

- React app, Django, Spring boot 빌드 후 도커라이징

- docker-compose를 이용해 container로 띄움

- docker-compose file

```yaml
version: '3.7'

services:
  backend:
    image: backend-spring
    build:
      context: Back/demo
      dockerfile: Dockerfile
    ports:
      - "8443:8443"
    volumes:
      - /home/ubuntu/docker-volume/ssl:/root
    container_name: "backend"

  backendDjango:
    image: backend-django
    build:
      context: Back/backend
      dockerfile: Dockerfile
    ports:
      - "8081:8081"
    volumes:
      - /home/ubuntu/docker-volume/ssl:/root
    container_name: "backendDjango"

  frontend:
    image: frontend-react
    build:
      context: Front/be-travelic/
      dockerfile: Dockerfile
    ports:
      - "3000:3000"
    volumes:
      - /home/ubuntu/docker-volume/ssl:/var/www/html
    container_name: "frontend"
```

4. **DB 접속 정보 등 프로젝트(ERD)에 활용되는 주요 계정 및 프로퍼티가 정의된 파일 목록**

- **Back/demo/src/main/resources/application.properties**