

Exec

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

[백엔드 & 데이터]

<Spring Boot>

- JVM : openjdk:8-jdk
- IntelliJ : 2022.02.01

<Database>

- DB : MySQL Server 8.0.22

<배포>

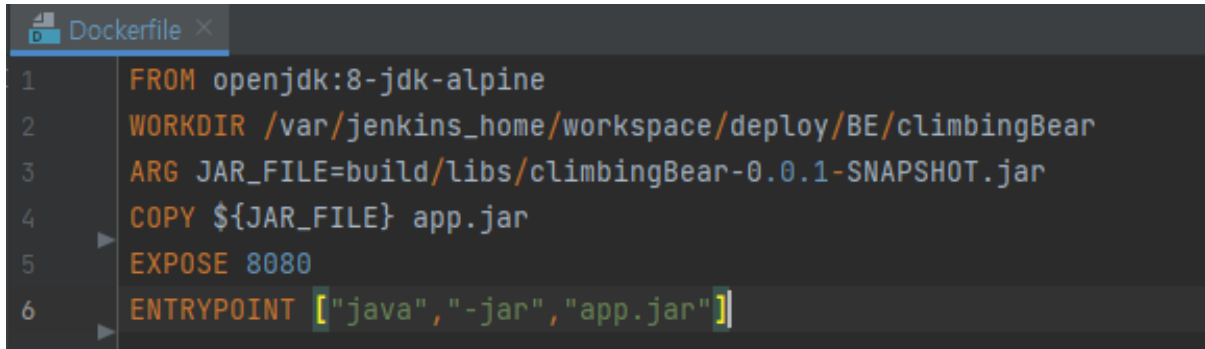
- Docker : 20.10.21
- Jenkins :

[프론트]

- ReactNative 2.0.1
- Node.js 16.17.0
- Jdk 14
- Python 2.7.18
- JavaScript ES6
- Axios 0.27.2
- Redux-Toolkit 1.8.6
- Visual Studio Code 1.70.0

2. 빌드 시 사용되는 환경 변수 등의 주요 내용 상세 기재

- Spring Boot

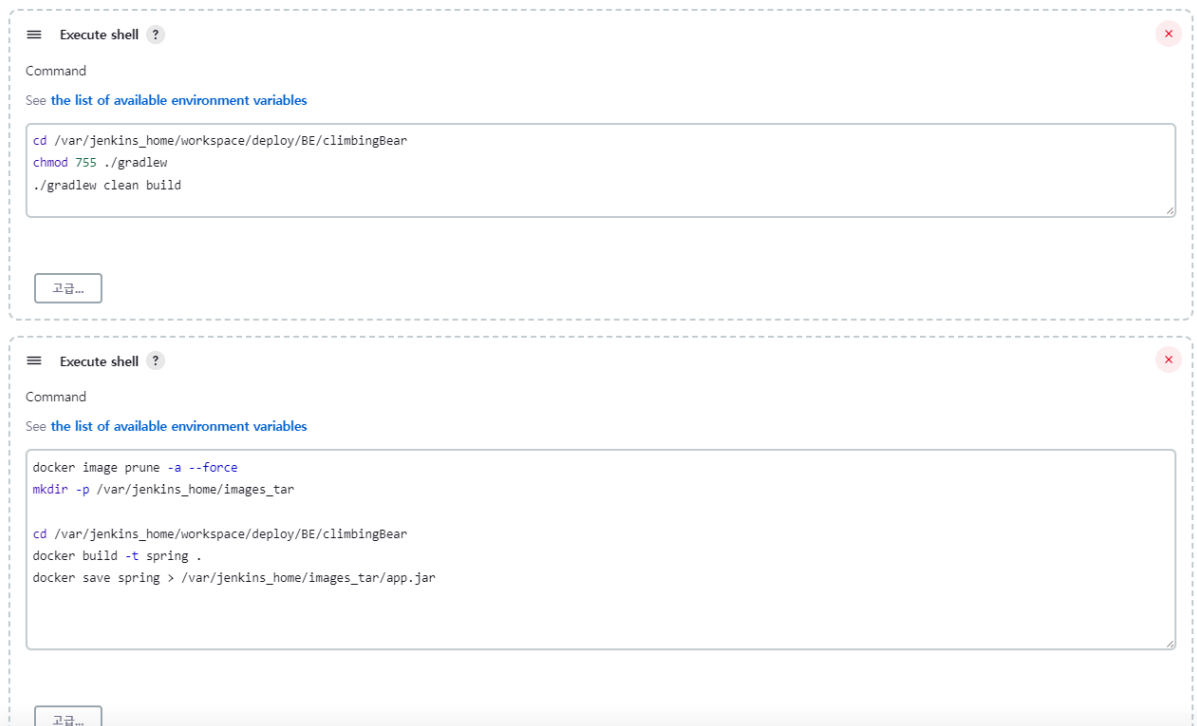


```
1 FROM openjdk:8-jdk-alpine
2 WORKDIR /var/jenkins_home/workspace/deploy/BE/climbingBear
3 ARG JAR_FILE=build/libs/climbingBear-0.0.1-SNAPSHOT.jar
4 COPY ${JAR_FILE} app.jar
5 EXPOSE 8080
6 ENTRYPOINT ["java", "-jar", "app.jar"]
```

3. 배포 시 특이사항 기재

- Spring Boot 빌드 후 도커라이징 (Jenkins에서 권한설정)

Build Steps



Execute shell ?

Command

See [the list of available environment variables](#)

```
cd /var/jenkins_home/workspace/deploy/BE/climbingBear
chmod 755 ./gradlew
./gradlew clean build
```

고급...

Execute shell ?

Command

See [the list of available environment variables](#)

```
docker image prune -a --force
mkdir -p /var/jenkins_home/images_tar

cd /var/jenkins_home/workspace/deploy/BE/climbingBear
docker build -t spring .
docker save spring > /var/jenkins_home/images_tar/app.jar
```

고급...

- Docker-compose를 이용해 container로 띄움
- Docker-compose file

```
version: '3'

services:
  v
  react:
    container_name: react-dockerize
    build:
      context: .
      dockerfile: dockerfile
    volumes:
      - usr/src/app/node_modules
      - .:/usr/src/app
    ports:
      - "3000:3000"
    environment:
      - NODE_ENV=development
    stdin_open: true
    tty: true
```

4. DB 접속 정보 등 프로젝트(ERD)에 활용되는 주요 계정 및 프로퍼티가 정의된 파일 목록
- Back/demo/src/main/resources/application.properties

```
# jpa
spring.jpa.generate-ddl=true
spring.jpa.hibernate.ddl-auto=update
spring.jpa.show-sql=true

# h2
spring.datasource.driver-class-name=org.h2.Driver
spring.datasource.username=sa
spring.datasource.password=
spring.datasource.url=jdbc:h2:tcp://localhost/~ /test;
spring.h2.console.enabled=true

# mysql
spring.datasource.driver-class-name=com.mysql.cj.jdbc.Driver
spring.datasource.url=jdbc:mysql://k7d109.p.ssafy.io:3306/climbingBear
spring.jpa.database=mysql
spring.datasource.username=root
spring.datasource.password=jyh9524

# jwt
jwt.secret=123456789213456789asdfsdfasdfsdfsdfwerwrersfd
spring.jpa.open-in-view=false
spring.jpa.defer-datasource-initialization=true

# google cloud storage : file upload / download
spring.servlet.multipart.enabled=true
spring.cloud.gcp.credentials.location=classpath:static/climbingbear-367901-07c9a11fe26a.json

# web socket
spring.devtools.livereload.enabled=true
spring.devtools.restart.enabled=false
spring.freemarker.cache=false
spring.jackson.serialization.fail-on-empty-beans=false
```