

Gitlab 소스 클론 이후 빌드/배포 시 참고사항

1. 개발 환경

- 사용한 JVM
 - BellSoft LibericaJDK-17
- 웹서버/WAS
 - Spring boot 3.4.3
- IDE 버전
 - IntelliJ IDEA 2024.3.1.1

2. 빌드 시 사용되는 환경변수

- 4. 프로퍼티 참고

3. 배포 시 특이사항

- 배포 환경
 - 운영 환경에서는 Jenkins에서 빌드 파일을 생성하여 Docker 컨테이너로 자동 배포함
 - 로컬에서 실행 시, 수동으로 BE 빌드 파일(JAR)을 생성해야 함
 - MySQL 서버는 로컬에 직접 설치함
 - redis 서버를 필요로 하므로 docker로 배포하는 것을 권장함

- 배포 스크립트

```
cd BackEnd
./gradlew clean build
cd ../infra
docker compose down
docker compose up -d
```

- BE 배포 방법

```
# MySQL, redis를 직접 설치하거나 docker로 구동 중이라고 가정한다.  
cd BackEnd  
.Wgradlew clean build  
java -jar build/libs/econimal-0.0.1-SNAPSHOT.jar
```

■ FE 배포 방법

```
# BE가 정상 배포되었다고 가정한다.  
cd FrontEnd  
npm install  
npm run dev  
yarn dev
```

4. DB 접속 정보 등 프로젝트에 활용되는 주요 계정 및 프로퍼티가 정의된 파일 목록

■ DB 접속 정보

- 배포 환경에서는 AWS RDS를 사용하여 MySQL 서버를 구성함
- 로컬에서 배포 시 MySQL을 직접 설치해야 함(localhost:3306)

■ 주요 계정

- ERD 조회는 별첨한 파일 참고

■ 프로퍼티

● 경로

- S12P21A504WBackEndWsrcWmainWresourcesWsecret.yml
- S12P21A504WFrontEndW.env
- 상세 내용은 S12P21A504/exec/ 밑에있는 sample 참고

5. 비교. 분산 시스템 구축 관련

■ Hadoop 에코시스템 구축

```
# BE가 정상 배포되었다고 가정한다.  
cd infra#docker-hadoop-spark  
docker compose up
```

■ 벌크 데이터 HDFS 적재

데이터 경로

```
docker cp climates_202503260931.csv ₩  
namenode:climates_202503260931.csv  
docker exec -it namenode /bin/bash  
  
hdfs dfs -mkdir -p /climate  
hdfs dfs -put climates_202503260931.csv /climate/climates_202503260931.csv
```

- <http://localhost:9870> 에서 데이터가 파티션 단위로 적재된 것을 확인할 수 있다.

Browse Directory

Permission	Owner	Group	Size	Last Modified	Replication	Block Size	Name	
-rw-r--r--	root	supergroup	0 B	Mar 29 01:07	3	128 MB	_SUCCESS	
drwxr-xr-x	root	supergroup	0 B	Mar 29 01:07	0	0 B	year=1931	
drwxr-xr-x	root	supergroup	0 B	Mar 29 01:07	0	0 B	year=1932	
drwxr-xr-x	root	supergroup	0 B	Mar 29 01:07	0	0 B	year=1933	
drwxr-xr-x	root	supergroup	0 B	Mar 29 01:07	0	0 B	year=1934	
drwxr-xr-x	root	supergroup	0 B	Mar 29 01:07	0	0 B	year=1935	
drwxr-xr-x	root	supergroup	0 B	Mar 29 01:07	0	0 B	year=1936	
drwxr-xr-x	root	supergroup	0 B	Mar 29 01:07	0	0 B	year=1937	
drwxr-xr-x	root	supergroup	0 B	Mar 29 01:07	0	0 B	year=1938	
drwxr-xr-x	root	supergroup	0 B	Mar 29 01:07	0	0 B	year=1939	

■ Spark

```
14 spark = SparkSession.builder \  
15     .appName("CalculateTempHumPipeline") \  
16     .master("spark://spark-master:7077") \  
17     .config("spark.hadoop.fs.defaultFS", "hdfs://namenode:9000") \  
18     .config("spark.redis.host", "YOUR_REDIS_HOST") \  
19     .config("spark.redis.port", "6379") \  
20     .config("spark.redis.pipeline.enabled", "true") \  
21     .config("spark.redis.write.batch.size", "1000") \  
22     .getOrCreate()  
23
```

- Line 18에 redis가 설치된 서버의 IP 주소를 변경한다.

Redis가 정상 배포되었다고 가정한다.

```
docker cp DataWglobeWtemphumWcalculate_temphum_pipeline.py W
```

```
spark-master:calculate_temphum_pipeline.py
```

```
docker exec -it spark-master bash /spark/bin/spark-submit W
```

```
--packages com.redislabs:spark-redis_2.12:3.1.0 calculate_temphum_pipeline.py
```