



MilkMarket

포팅 매뉴얼

프로젝트 기간 : 2022.2.21.~4.8

삼성SW청년아카데미 서울캠퍼스 6기

윤희영 이상우 이다예 이상훈 권도혁

1. 기술스택

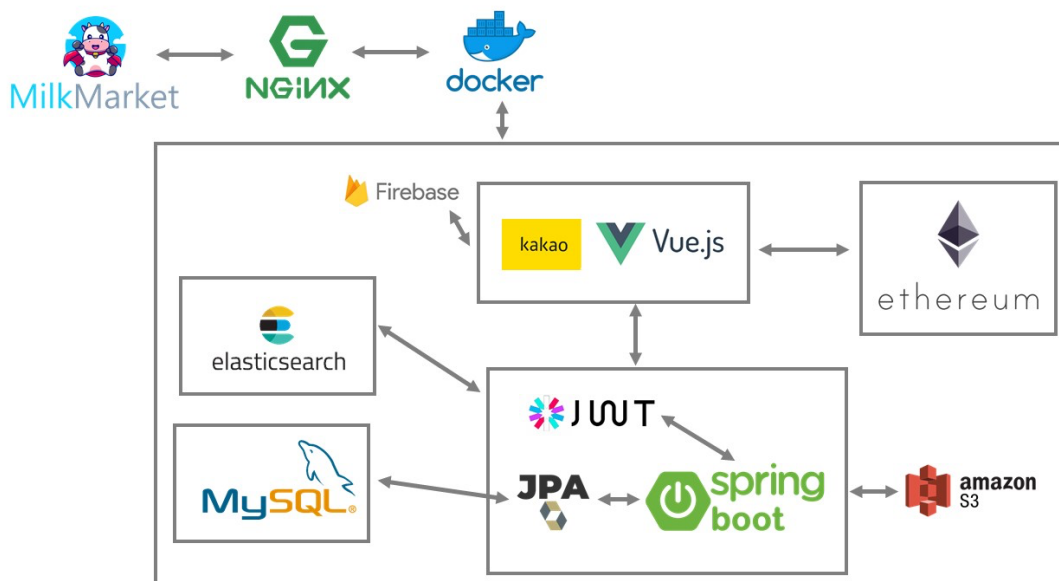
구분	기술스택	상세내용	버전
공통	형상관리	Gitlab	-
	이슈관리	Jira	-
	커뮤니케이션	Mattermost, Notion	-
BackEnd	DB	MySQL	5.7
		JPA	-
	Java	JavaSE	11
	Spring	Spring	5.3.6
		Spring Boot	2.4.5
	IDE	Eclipse	JEE 2020-06
		IntelliJ	2021.3.1
	클라우드 스토리지	AWS S3	-
	Build	Maven	4.0.0
	Elastic	ElasticSearch	7.15.1
	API Docs	Swagger2	3.0.0
FrontEnd	HTML5		-
	CSS3		-
	JavaScript(ES6)		-
	Vue	React	17.0.2
		Redux	7.2.6
		Redux-thunk	2.4.1
	Firebase	Firebase	8.10.0
	IDE	Visual Studio Code	1.63.2
BlockChain	Web3	Solidity	0.8.12
		Web3.js	1.5.3
		truffle	5.5.5
		Ganache	7.0.3
		openzeppelin	4.5.0
Server	서버	AWS EC2	-
	플랫폼	Ubuntu	20.04.3 LTS
	배포	Docker	20.10.12

2. 상세내용

□ 개요

아래 그림은 “우유마켓” 서비스의 서비스 흐름도입니다. 팀원들이 각자 작성한 프로젝트를 Docker 이미지로 빌드하여 배포합니다. Elastic Search는 EC2에서 설치하여 구동합니다.

<서비스 흐름도>



각 프로젝트를 빌드 후에는 Docker 이미지를 만들고 이를 Docker Hub에 push 합니다. Docker Hub로부터 서비스에 필요한 이미지를 받아와 컨테이너로 띄웁니다.

SSAFY에서 지원받은 EC2 싱글 인스턴스로 인프라를 구축했습니다. 추후 서비스화를 위해 Nginx는 리버스 프록시 서버로 설정하였습니다.

□ FrontEnd

- Docker 이미지 생성을 위해 Dockerfile을 작성합니다.
(해당 파일은 프로젝트 내에 이미 작성되어있습니다.)

```
# build stage
FROM node:lts-alpine as build-stage
WORKDIR /app
COPY package*.json ./
RUN npm install
COPY . .
RUN npm run build
EXPOSE 8083

# production stage
FROM nginx:stable-alpine as production-stage
COPY --from=build-stage /app/dist /usr/share/nginx/html
EXPOSE 80
CMD ["nginx", "-g", "daemon off;"]
```

- docker 이미지 생성

```
# 이미지 빌드
docker build . -t front:0.1

# 태그에 버전 작성
docker tag front:0.1 front:latest

# 사용자 name으로 tag 작성
docker tag front:latest <DOCKER_HUB_ID>/front:latest

# 이미지 푸시
docker push <DOCKER_HUB_ID>/front:latest
```

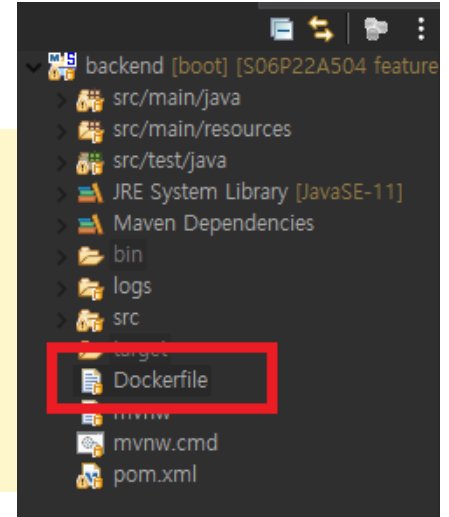
- 생성해놓은 FE 이미지 컨테이너 실행

```
docker run --name front -d -p 8083:80 <DOCKER_HUB_ID>/front
```

□ BackEnd

- Eclipse에서 backend 폴더를 import 합니다.
- Docker 이미지 생성을 위해 Dockerfile을 작성합니다.
(해당 파일은 프로젝트 내에 이미 작성되어있습니다.)

```
FROM openjdk:11-jdk  
ARG JAR_FILE=target/*.jar  
COPY ${JAR_FILE} app.jar  
EXPOSE 8080  
ENTRYPOINT ["java","-jar","/app.jar"]
```



- Maven 빌드

```
mvn install
```

- docker 이미지 생성

```
# 이미지 빌드  
docker build . -t back:0.1  
  
# 태그에 버전 작성  
docker tag front:0.1 back:latest  
  
# 사용자 name으로 tag 작성  
docker tag back:latest <DOCKER_HUB_ID>/back:latest  
  
# 이미지 푸시  
docker push <DOCKER_HUB_ID>/back:latest
```

- mysql 이미지 다운로드

```
docker pull mysql:5.7.37
```

- mysql 컨테이너 실행

```
# 컨테이너 실행
docker run --name mysql -e MYSQL_ROOT_PASSWORD=<password> -d -p 3306:3306 mysql:5.7.37 --character-set-server=utf8mb4 --collation-server=utf8mb4_unicode_ci

# mysql 클라이언트 명령어를 실행하여 DB 접속 테스트
docker exec -it mysql mysql -u root -p
```

- 생성해놓은 BE 이미지 컨테이너 실행

```
docker run --name back -d -p 8080:8080 <DOCKER_HUB_ID>/back
```

- 서버 컨테이너 로그 확인

```
docker logs <containerID>
```

□ Smart Contract

- truffle, contract 설치

```
cd smart-contract
npm init -ya
npm install -g truffle # 트러플 설치 (스마트 컨트랙트 지원 tool)
```

- truffle 설정(truffle-config.js) 수정

```
module.exports = {
  networks: {
    development: {
      host: "127.0.0.1", // Localhost (default: none)
      port: 8545, // Standard Ethereum port (default: none)
      network_id: "*", // Any network (default: none)
    },
    mocha: {
      // timeout: 100000
    },
    compilers: {
      solc: {
        version: "0.8.12",
      },
    },
  },
};
```

- Ganache 실행


```
ganache-cli
```

- cmd창에 출력 된 내용 중 `mnemonic` 에 해당하는 문장을 `.secret` 폴더에 입력
- smart-contract 위치에서 `truffle migrate --reset` 명령어 실행
- 생성된 contract-address를 프론트엔드 `config` 폴더에서 `index.js` 파일에 입력

□ Elastic Search

- Elastic Search 공식 홈페이지에서 7.15.1 버전을 다운받고 압축풀기를 합니다.

<Elastic Search 다운로드>

 elastic 제품 고객 사례 정보 회사 소개 요금제 보기 문의하기 로그인 무료로 시작 Q

Download Elasticsearch

1 Download and unzip Elasticsearch

Choose platform:

Windows

Windows

sha

asc

플랫폼 선택 후 다운로드

Summary

버전 선택

Version: 8.1.2

View past releases →

Release date: April 01, 2022

Detailed release notes →

License: Elastic License 2.0

Elastic License 2.0 →

- 압축을 푼 폴더에서 config > elasticsearch.yml 파일을 아래와 같이

<Elastic Search 다운로드>

```
cluster.name: "es-cluster-1"
node.name: "node-1"

xpack.security.enabled: false

network.bind_host: 사용자 주소
http.port: 9200
node.master: true
node.data: true
transport.host: localhost
transport.tcp.port: 9300
```

- Elastic Search 실행(압축 풀 폴더에서 실행)

```
bin/elasticsearch -d
```


3. 특이사항

☐ 개요

- Milk Market 서비스는 Docker 이미지 컨테이너를 기반으로 서비스를 배포하고 있습니다. 서비스에 문제가 발생 시, 아래 명령어를 확인하여 상태를 확인할 수 있습니다. 또한 상품조회 서비스에 문제가 발생 시에는 elastic search가 정상적으로 구동되고 있는지 확인이 필요합니다. BackEnd, FrontEnd, Mysql 프로젝트의 상태를 확인하기 위해선 각 컨테이너의 로그를 확인하는 명령어를 사용하여 log 확인이 가능합니다.

☐ Nginx

- 상태 확인

```
sudo service nginx status
```

- 재실행

```
sudo service nginx restart
```

☐ Elastic Search

- 접속 상태 확인(웹브라우저에서 검색)

```
사용자주소:9200
```

☐ Docker

- 컨테이너 확인

```
sudo docker ps -a
```

- 서버 컨테이너 로그 확인

```
docker logs <containerID>
```

- 컨테이너 재실행

```
sudo docker restart <containerID>
```

- 컨테이너 삭제

```
sudo docker rm <containerID>
```

- 이미지 삭제

```
sudo docker rmi <imageID>
```

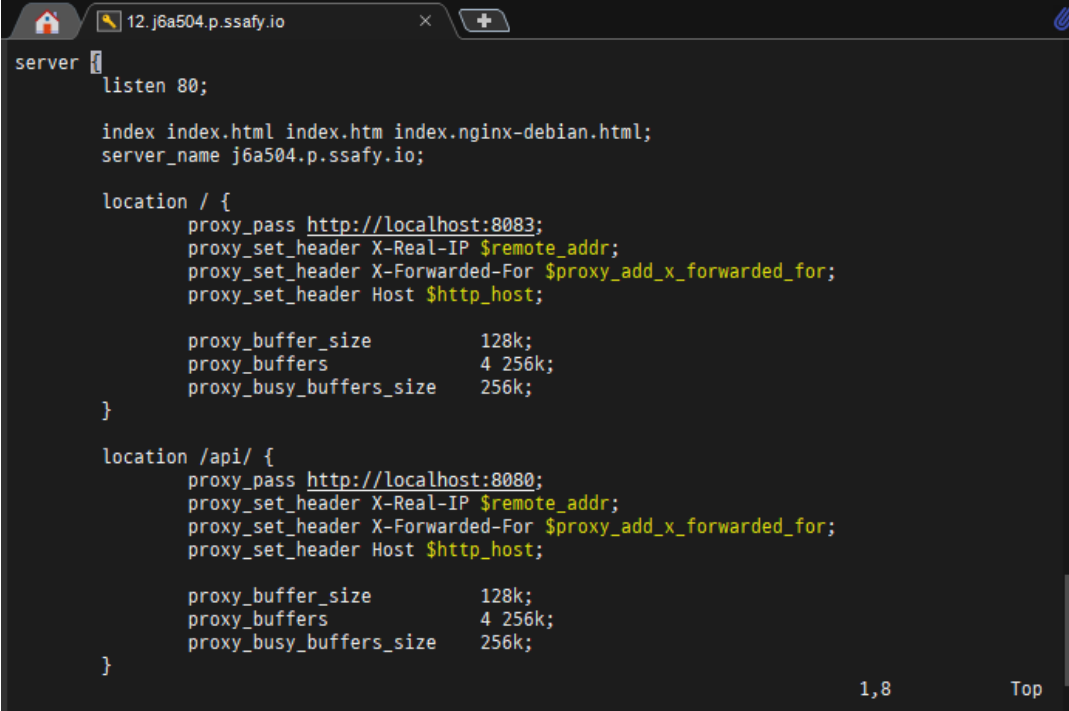
4. 프로퍼티 정의

☐ Nginx

○ 환경설정

```
sudo vi /etc/nginx/sites-available/default
```

<Nginx 설정 화면>



```
server {  
    listen 80;  
  
    index index.html index.htm index.nginx-debian.html;  
    server_name j6a504.p.ssafy.io;  
  
    location / {  
        proxy_pass http://localhost:8080;  
        proxy_set_header X-Real-IP $remote_addr;  
        proxy_set_header X-Forwarded-For $proxy_add_x_forwarded_for;  
        proxy_set_header Host $http_host;  
  
        proxy_buffer_size 128k;  
        proxy_buffers 4 256k;  
        proxy_busy_buffers_size 256k;  
    }  
  
    location /api/ {  
        proxy_pass http://localhost:8080;  
        proxy_set_header X-Real-IP $remote_addr;  
        proxy_set_header X-Forwarded-For $proxy_add_x_forwarded_for;  
        proxy_set_header Host $http_host;  
  
        proxy_buffer_size 128k;  
        proxy_buffers 4 256k;  
        proxy_busy_buffers_size 256k;  
    }  
}
```

☐ MySQL

○ MySQL Docker 컨테이너에서 DB 스키마를 생성해두면 SpringBoot 구동 시 자동으로 Table이 생성됩니다.

○ 계정 생성

```
#계정 생성  
create user milkmarket@% identified by 1234;  
  
# 권한 부여  
grant all privileges on *.* to milkmarket@% with grant option;  
flush privileges;
```

- 스키마 생성

```
create database if not exists milkmarket_db collate utf8mb4_general_ci;
```