**1. 프로젝트 버전**

(1) 앱 서버

a. Apache Tomcat : 9.0.76

b. Nginx : 1.18.0

(2) 분산 서버 : Apache Kafka 3.1.2

(3) 가상 서버 : Docker

a. Docker : 24.0.6

b. Docker Compose : 1.25.0

(4) 배포 : Jenkins 2.414.1

(5) JVM : 11.0.19

(6) Python : 3.9.13

(7) FE

a. Android Studio : 2022.03

b. Flutter : 3.12.0

c. Dart : 3.1.0

d. DevTools : 2.24.0

(8) BE

a. Spring boot : 2.7.13

b. Fast API : 0.103.1

(9) DB : MySQL 8.0.34

**2. 환경 변수**

(1) Port

a. Nginx : 80(http)

b. 앱 서버 : 8000

c. 크롤러 및 분산 서버 : 8000

d. MySQL : 3306

e. Zookeeper : 2181

f. Kafka : 9092

(2) Spring DB Setting

spring.datasource.hikari.driver-class-name=com.mysql.cj.jdbc.Driver  
spring.datasource.url=jdvc:mysql://j9d202.p.ssafy.io:3306/fitter?useUniCode=yes&characterEncoding=UTF-8

(3) JPA

spring.jpa.hibernate.ddl-auto=validate  
spring.jpa.database=mysql  
spring.jpa.show-sql=true  
spring.jpa.database-platform=org.hibernate.dialect.MySQL8Dialect  
spring.jpa.properties.hibernate.format\_sql=true  
spring.jpa.properties.hibernate.globally\_quoted\_identifiers=true

(4) JWT

jwt.secretKey=JWT 시크릿 키 값  
# 1 hour  
jwt.access.expiration=3600000  
jwt.access.header=Authorization  
# 2 week  
jwt.refresh.expiration 1209600000  
jwt.refresh.header=Authorization-refresh

(5) File

spring.servlet.multipart.location=/app/file  
spring.servlet.multipart.max-file-size=50MB  
spring.servlet.multipart.max-request-size=50MB  
file.folder.profile=/profileImg/

(6) Naver News Search API

naver.api.url=https://openapi.naver.com  
naver.api.search.url=/v1/search/news.json  
naver.api.client.id=클라이언트 ID  
naver.api.client.secret=클라이언트 secret key 값

(7) Spring Security

spring.security.oauth2.client.registration.kakao.client-id=카카오 id  
spring.security.oauth2.client.registration.kakao.client-secret=카카오 pw  
spring.security.oauth2.client.registration.kakao.redirect-uri=카카오 리다이렉트 URI   
spring.security.oauth2.client.registration.kakao.client-authentication-method=POST  
spring.security.oauth2.client.registration.kakao.authorization-grant-type=카카오 인증코드   
spring.security.oauth2.client.registration.kakao.scope=profile\_nickname  
spring.security.oauth2.client.registration.kakao.client-name=Kakao  
spring.security.oauth2.client.registration.kakao.prompt=login  
spring.security.oauth2.client.provider.kakao.authorization-uri= 카카오 인증 URI  
spring.security.oauth2.client.provider.kakao.token-uri=카카오 토큰 URI spring.security.oauth2.client.provider.kakao.user-info-uri=카카오 유저정보 URI spring.security.oauth2.client.provider.kakao.user-name-attribute=id

**3. 배포**

- 앱 서버

(1) Server Name : <http://j9d202.p.ssafy.io:8000>

(2) Jenkins

|  |
| --- |
| version: '3'  services:  jenkins:  container\_name: jenkins  image: jenkins/jenkins:lts  restart: on-failure  ports:  - "8080:8080"  volumes:  - /var/jenkins\_home:/var/jenkins\_home  - /var/run/docker.sock:/var/run/docker.sock  - /usr/bin/docker:/usr/bin/docker  - /var/www/backend:/var/www/backend  - /var/www/frontend:/var/www/frontend  user: root |

- 크롤러 및 분산 서버

(1) Server Name : http://j9d202a.p.ssafy.io:8000

(2) Jenkins

|  |
| --- |
| version: '3'  services:  jenkins:  container\_name: jenkins  image: jenkins/jenkins:lts  restart: on-failure  ports:  - "8080:8080"  volumes:  - /var/jenkins\_home:/var/jenkins\_home  - /var/run/docker.sock:/var/run/docker.sock  - /usr/bin/docker:/usr/bin/docker  - /var/www/backend:/var/www/backend  - /var/www/frontend:/var/www/frontend  user: root |

(3) Kafka

|  |
| --- |
| version: '2'  services:  zookeeper:  image: wurstmeister/zookeeper:latest  container\_name: zookeeper  ports:  - "2181:2181"  kafka:  image: wurstmeister/kafka:latest  container\_name: kafka  depends\_on:  - zookeeper  ports:  - "9092:9092"  environment:  KAFKA\_ADVERTISED\_HOST\_NAME: j9d202a.p.ssafy.io  KAFKA\_ZOOKEEPER\_CONNECT: zookeeper:2181  KAFKA\_ADVERTISED\_PORT: 9092  volumes:  - /var/run/docker.sock:/var/run/docker.sock |

**4. 주요 접속 계정**

(1) DB

ID : ssafy

PW : ssafy