1. 사용한 JVM, 웹서버, WAS, 제품 등 종류와 설정 값, 버전

- Build Tool : Gradle
- Back-End 언어 : Java 11
- FrameWork : Spring Boot 2.7.15
- Front-End : React
- Docker-compose.yml script

```
version: '3'

services:
    nginx:
        image: nginx:latest  # Nginx 이미지를 사용합니다. 버전은 필요에 따라 조정할 수 있습니다.
        container_name: nginx
        environment:
            - TZ=Asia/Seoul
        networks:
            - front
        expose:
            - "80"
            - "443"
        ports:
            - "80:80"
            - "443:443"
        volumes:
                #- ./data/nginx/app.conf:/etc/nginx/conf.d/default.conf
            - ./data/nginx:/etc/nginx/conf.d
            - ./data/certbot/conf:/etc/letsencrypt
            - ./data/certbot/www:/var/www/certbot
        restart: unless-stopped  # 컨테이너가 종료되면 항상 다시 시작합니다.
        # 컨테이너가 만들어진 후에 실행할 명령어
        command: "/bin/sh -c 'while :; do sleep 6h & wait $${!}; nginx -s reload; done & nginx -g \"daemon off;\"'"

    certbot:
        image: certbot/certbot
        container_name: certbot
```

```
-         networks:
-             - front
-         restart: unless-stopped
-         volumes:
-             - ./data/certbot/conf:/etc/letsencrypt
-             - ./data/certbot/www:/var/www/certbot
-         entrypoint:  "/bin/sh -c 'trap exit TERM; while :; do
  certbot renew; sleep 12h & wait $${!}; done;'"
-
-     jenkins:
-         image: jenkins/jenkins:lts
-         container_name: jenkins
-         networks:
-             - back
-             - front
-         volumes:
-             - /var/run/docker.sock:/var/run/docker.sock
-             - ./data/jenkins:/var/jenkins_home
-             - ./data/nginx:/var/nginx
-
-         ports:
-             - "9090:8080"
-         environment:
-             - JENKINS_SLAVE_AGENT_PORT=3000 # 에이전트와의 통신 포트
  설정
-         user: root
-
-     mysql:
-         image: mysql:latest
-         container_name: mysql
-         networks:
-             - back
-         environment:
-             MYSQL_ROOT_PASSWORD: rlarhkdtjrdmlvuswl123
-             MYSQL_DATABASE: i9
-             MYSQL_USER: i9B309
-             MYSQL_PASSWORD: dkdlrnB309
-         command: --character-set-server=utf8mb4 --collation-
  server=utf8mb4_unicode_ci
-         volumes:
-             - ./data/mysql:/var/lib/mysql
-         ports:
```

```
-             - "3306:3306"
-
-     redis:
-         image: redis:latest
-         container_name: redis
-         networks:
-             - back
-         volumes:
-             - ./data/redis:/data
-         ports:
-             - "6379:6379"
-
-     mongodb-config1:
-         image: mongo
-         command: mongod --configsvr --replSet configRepl --port
  20001
-         container_name: mongo-config1
-         ports:
-             - 20001:20001
-         volumes:
-             - ./mongo-dir/mongodb-config1:/data/db
-
-
-
-     mongodb-config2:
-         image: mongo
-         command: mongod --configsvr --replSet configRepl --port
  20002
-         container_name: mongo-config2
-         ports:
-             - 20002:20002
-         volumes:
-             - ./mongo-dir/mongodb-config2:/data/db
-
-
-     mongo:
-         image: mongo
-         container_name: mongo-root
-         command: mongos --configdb
  configRepl/j9b309.p.ssafy.io:20001,j9b309.p.ssafy.io:20002 --
  bind_ip_all
-         ports:
```

```
-             - 20000:27017
-         volumes:
-             - ./mongo-dir/mongodb:/data/db
-         environment:
-             MONGO_INITDB_ROOT_USERNAME: i9B309
-             MONGO_INITDB_ROOT_PASSWORD: dkdlrnB309
-             MONGO_INITDB_DATABASE: SSAFY
-
-     mongo-shard11:
-         image: mongo
-         container_name: mongo-shard11
-         command: mongod --shardsvr --replSet shardRep1 --port 30011
-         ports:
-             - 30011:30011
-         volumes:
-             - ./mongo-dir/mongodb-shard11:/data/db
-
-
-     mongo-shard12:
-         image: mongo
-         container_name: mongo-shard12
-         command: mongod --shardsvr --replSet shardRep1 --port 30012
-         ports:
-             - 30012:30012
-         volumes:
-             - ./mongo-dir/mongodb-shard12:/data/db
-
-
-     mongo-shard21:
-         image: mongo
-         container_name: mongo-shard21
-         command: mongod --shardsvr --replSet shardRep2 --port 30021
-         ports:
-             - 30021:30021
-         volumes:
-             - ./mongo-dir/mongodb-shard21:/data/db
-
-     mongo-shard22:
-         image: mongo
-         container_name: mongo-shard22
-         command: mongod --shardsvr --replSet shardRep2 --port 30022
-         ports:
```

```
-             - 30022:30022
-         volumes:
-             - ./mongo-dir/mongodb-shard22:/data/db
-
-
-  networks:
-      front:
-      back:
-
```

2. 환경 변수

- application.yml

```
-  server:
-    port: {포트 번호}
-
-  spring:
      security:
-      oauth2:
-        resourceserver:
-          jwt:
-            jwk-set-uri: {  인증 서버 주소 + "/oauth/token_key" ex: https://j9b309.p.ssafy.io/oauth/token_key }
-
-    datasource:
-      driver-class-name: com.mysql.cj.jdbc.Driver
-      url: { database url }
-      username: { database user name }
-      password: { database password }
-
-    jpa:
-      open-in-view: false
-      hibernate:
-        ddl-auto: update
-  data:
-      mongodb:
-        uri: { mongodb rul }
-        database: { database name }
-  mail:
```

```
-      host: smtp.gmail.com
-      port: 587
-      username: { gmail 이름 }
-      password: { gmail password }
-      properties:
-        mail:
-          smtp:
-            auth: true
-            starttls:
-              enable: true
-
-  url:
-    host: { 서비스 주소 이름 ex}https://j9b309.p.ssafy.io }
-
-  jwt:
-      secretKey: { jwt secret key }
```

3. 배포 시 특이사항
   - MSA 구조 배포