빌드 및 배포

1. 환경

Frontend : React, Javascript, Redux, Node.js 18.17.1 LTS, Yarn 3.6.1

Backend: Azul-Zulu version 17.0.9, Spring Boot 3.1.1, JPA

DB: PostgreSQL, Redis, MongoDB

Infra: Ubuntu 20.04 LTS, AWS EC2, AWS S3 Docker, Docker Compose, GitLab CICD, Jenkins:jdk17, Gradle, nginx

기타: Figma, Jira, GitLab, Mattermost, Webex, Notion

2. 환경 변수 형태

1. .env.development (frontend)

```
프론트 주소
네이버 API Client ID
구글 API Client ID

REACT_APP_LOGIN_SERVER_URL=https://i9b108.p.ssafy.io
REACT_APP_NAVER_CLIENT_ID=sOLMKlOcIu3pqoFbtMey
REACT_APP_GOOGLE_CLIENT_ID=953911532873-0ve3ob0gtc2eq0fdp8ui67mue02pufpr.apps.googleusercontent.com
```

2. AWS, AWS-KEY

```
AWS 설정정보

spring:
  data:
    couchbase:
      bucket-name: aquh
cloud:
  aws:
    stack:
      auto: false
    region:
      static: ap-northeast-2

AWS SECRET KEY (.properties)
cloud.aws.credentials.access-key=
cloud.aws.credentials.secret-key=
```

3. Mail

```
spring:
  mail:
    host: smtp.naver.com
    port: 465
    properties:
```

```
      mail:
        smtp:
          auth: true
          starttls:
            enable: true
          ssl:
            enable: enable
    username: 아이디
    password: 비밀번호
front:
  server:
```

4. MongoDB

```
spring:
  data:
    mongodb:
      uri: 접속 주소
      database: 데이터 베이스
```

5. Naver

```
naver:
  client:
    id: 네이버 api 클라이언트 id
    secret: 네이버 api 클라이언트 secret
```

6. Openvidu, https

```
server:
  ssl:
    enabled: true
    key-store: classpath:keystore.p12
    key-store-type: PKCS12
    key-alias: spring
OPENVIDU_URL: openvidu 서버 주소
OPENVIDU_SECRET: openvidu 서버 비밀번호

server:
  ssl:
    key-store-password:
```

7. PostgreSQL

```
spring:
  datasource:
    hikari:
      maximum-pool-size: 4
    url: DB 주소
    username: DB 아이디
    password: DB 패스워드
```

8. Redis

```
spring:
  data:
    redis:
      lettuce:
        pool:
          max-active: 5
          max-idle: 5
          min-idle: 2
      host: 디비 주소
      port: 6379
```

9. docker-compose.yml 설정 (/home/ubuntu/docker-compose.yml)

```
version: "3"

services:
        database:
                container_name: postgres
                image: postgres:latest
                restart: always
                ports:
                        - 5432:5432
                environment:
                        TZ: Asia/Seoul
                        POSTGRES_DB: AQuh
                        POSTGRES_USER: postgres
                        POSTGRES_PASSWORD: 2208
                volumes:
                        - ./postgres/:/var/lib/postgresql/data
        redis:
                container_name: redis
                image: redis
                ports:
                        - 6379:6379
                volumes:
                        - ./redis/data:/data
                        - ./redis/conf/redis.conf:/usr/local/conf/redis.conf
                labels:
                        - "name=redis"
                        - "mode=standalone"
                restart: always
                command: redis-server /usr/local/conf/redis.conf
                environment:
                        TZ: Asia/Seoul
        mongo:
                image: mongo
                container_name: mongo
                ports:
                        - 27017:27017
                environment:
                        MONGO_INITDB_DATABASE: AQuh
                        MONGO_INITDB_ROOT_USERNAME: mongo
                        MONGO_INITDB_ROOT_PASSWORD: 2208
                        TZ: Asia/Seoul
                volumes:
                        - ./mongodb:/data/db
        jenkins:
                image: jenkins/jenkins:jdk17
                container_name: jenkins
                volumes:
                        - /var/run/docker.sock:/var/run/docker.sock
                        - /jenkins:/var/jenkins_home
                ports:
```

```
                        - 9090:8080
                privileged: true
                user: root
```

10. Nginx 설정 (/etc/nginx/sites-available/my_site)

```
server {
   listen 80;
        location / {
                if ($request_method = 'OPTIONS') {
                      add_header 'Access-Control-Allow-Origin' '*';
                      add_header 'Access-Control-Allow-Methods' 'GET, POST, DELETE, PATCH, OPTIONS';
                      add_header 'Access-Control-Allow-Headers' 'Content-Type, Authorization';
                      add_header 'Access-Control-Max-Age' 86400;
                      return 204;
                }
                add_header 'Access-Control-Allow-Origin' '*' always;
                proxy_pass https://localhost:3000;
        }

        location /api {
                proxy_pass https://localhost:8080/api;
        }

        location /apiv2 {
                proxy_pass https://localhost:5000/apiv2;

        }

    server_name i9b108.p.ssafy.io;
    listen 443 ssl; # managed by Certbot
    ssl_certificate /etc/letsencrypt/live/i9b108.p.ssafy.io/fullchain.pem; # managed by Certbot
    ssl_certificate_key /etc/letsencrypt/live/i9b108.p.ssafy.io/privkey.pem; # managed by Certbot
    # include /etc/letsencrypt/options-ssl-nginx.conf; # managed by Certbot
    # ssl_dhparam /etc/letsencrypt/ssl-dhparams.pem; # managed by Certbot

    client_max_body_size 1G;
}
```

11. openvidu 설정 (/etc/openvidu/.env)

```
# OpenVidu configuration
# ----------------------
# Documentation: https://docs.openvidu.io/en/stable/reference-docs/openvidu-config/

# NOTE: This file doesn't need to quote assignment values, like most shells do.
# All values are stored as-is, even if they contain spaces, so don't quote them.

# Domain name. If you do not have one, the public IP of the machine.
# For example: 198.51.100.1, or openvidu.example.com
DOMAIN_OR_PUBLIC_IP=i9b108.p.ssafy.io

# OpenVidu SECRET used for apps to connect to OpenVidu server and users to access to OpenVidu Dashboard
OPENVIDU_SECRET=JAEWON

# Certificate type:
# - selfsigned:  Self signed certificate. Not recommended for production use.
#                Users will see an ERROR when connected to web page.
# - owncert:     Valid certificate purchased in a Internet services company.
#                Please put the certificates files inside folder ./owncert
#                with names certificate.key and certificate.cert
```

```
# - letsencrypt: Generate a new certificate using letsencrypt. Please set the
#                required contact email for Let's Encrypt in LETSENCRYPT_EMAIL
#                variable.
CERTIFICATE_TYPE=letsencrypt

# If CERTIFICATE_TYPE=letsencrypt, you need to configure a valid email for notifications
LETSENCRYPT_EMAIL=kjwkjw1104@gmail.com

# Proxy configuration
# If you want to change the ports on which openvidu listens, uncomment the following lines

# Allows any request to http://DOMAIN_OR_PUBLIC_IP:HTTP_PORT/ to be automatically
# redirected to https://DOMAIN_OR_PUBLIC_IP:HTTPS_PORT/.
# WARNING: the default port 80 cannot be changed during the first boot
# if you have chosen to deploy with the option CERTIFICATE_TYPE=letsencrypt
HTTP_PORT=7081

# Changes the port of all services exposed by OpenVidu.
# SDKs, REST clients and browsers will have to connect to this port
HTTPS_PORT=7082
```

3. 배포 시 특이사항

Frontend:

1. root 디렉토리에 .cert 디렉토리 생성
2. cert.pem, key.pem 파일 생성
3. 명령어 입력

```
yarn install
yarn start
```

Backend:

1. AWS-KEY, OPENVIDU-KEY.application.yml 생성 및 작성
2. `./gradlew clean bootJar` 실행
3. `cd ./build/libs` 실행
4. `java -jar *.jar` 실행

Docker Container:

1. docker-compose.yml 작성
2. `sudo docker-compose up -d` 실행

4. DB 접속 정보 및 ERD에 활용되는 주요 계정 및 프로퍼티 정의

1. 계정

2.

- URL : jdbc:postgresql://i9b108.p.ssafy.io:5432/AQuh
- 주소 : http://i9b108.p.ssafy.io:5432
- 아이디 : postgres
- 비번 : 2208