포팅 메뉴얼 - 공통 프로젝트

구미 1반 7조
🤼 김민지, 김지현, 김지훈, 이효진, 이희수, 임수민

개발환경💻

소통채널

- Mattermost
- Webex
- Notion
- Figjam

데이터베이스

- Redis : 7.0.8
- MariaDB : Ver 15.1 Distrib 10.10.2-MariaDB, for debian-linux-gnu (x86_64) using EditLine wrapper

기타 툴

- Postman : v10.9.4

이슈관리

- Jira

UI/UX

- Figma

형상관리

- Gitlab

IDE

- IntelliJ IDEA : 2022.3.1
- Visual Studio Code : 1.74.2

서버

- AWS EC2
- Ubuntu : 20.04 LTS
- Docker : 20.10.23, build 7155243
- Docker Compose : 2.15.1

OpenVidu 설정

WebRTC 화상회의를 위해 OpenVidu를 설치하고 설정한다.

https://docs.openvidu.io/en/2.25.0/deployment/ce/on-premises/

상기 링크의 On-Premise Deployment 메뉴얼을 따른다.

OpenVidu 설치

1. 상기된 개발환경 버전에 맞춰 Docker와 Docker compose를 설치
2. openvidu 구성요소 다운로드

```
cd /opt
curl https://s3-eu-west-1.amazonaws.com/aws.openvidu.io/install_openvidu_latest.sh | b
ash
```

3. /opt/openvidu 하위 폴더의 .env파일을 수정한다.
    a. DOMAIN_OR_PUBLIC_IP에 서버가 사용하는 도메인을 설정
    b. OPENVIDU_SECRET에 사용할 비밀번호를 설정
    c. CERTIFICATE_TYPE = letsencrypt를 사용한다.

        letsencrypt 값을 기입하면 openvidu-nginx 컨테이너 내부에 자동으로 SSL 설정이 이루어진다.

    d. LETSENCRYPT_EMAIL에 유효한 이메일을 설정

openvidu-nginx 파일 커스텀

1. nginx가 사용할 conf 파일을 컨테이너 내부로부터 꺼냄

```
cd /opt/openvidu
docker-compose exec nginx cat /etc/nginx/conf.d/default.conf > custom-nginx.conf
docker-compose exec nginx cat /etc/nginx/nginx.conf > nginx.conf
```

2. custom-nginx.conf 파일 수정

```
...

upstream yourapp {
    # server localhost:5442;
    # 프론트엔드 서버
    server localhost:3000;
}

... 중략 ...

##### 백엔드 요청 HTTPS internal redirect를 막기 위한 proxy_pass
    location /api {
        proxy_pass http://localhost:8815;
}
...
```

3. /opt/openvidu/docker-compose.yml 파일 수정

```
nginx:
        ...
        volumes:
            ...
            - ./custom-nginx.conf:/custom-nginx/custom-nginx.conf
            - ./nginx.conf:/etc/nginx/nginx.conf
```

Openvidu 기동

```
cd /opt/openvidu
./openvidu start
```

프론트엔드 빌드 및 배포

Dockerfile

```
FROM node:lts-alpine as build-stage
WORKDIR /app
COPY . .
RUN npm install --force
RUN npm run build

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

빌드 및 배포

1) Docker 이미지 생성

```
docker build -t pitchit-fe ./frontend
```

2) 현재 돌아가고 있는 프론트엔드 컨테이너가 있다면 삭제

```
docker rm -f pitchit-fe || true
```

3) 컨테이너 생성

```
docker run -d -p 3000:80 --name pitchit-fe pitchit-fe
```

백엔드 빌드 및 배포

application.yml

```
server:
  port: 8080
  servlet:
    context-path: /api

spring:
  redis:
    host: {redis 서버 주소}
    port: {redis 포트}
  datasource:
    driver-class-name: org.mariadb.jdbc.Driver
    url: jdbc:mariadb://MariaDB배포주소:포트/pitchit
    username: {username 입력}
    password: {password 입력}

  servlet:
    multipart:
      max-file-size: 500MB
      max-request-size: 500MB

  jpa:
    hibernate:
      ddl-auto: update
    show-sql: true

  security:
    oauth2:
      client:
        registration:
          kakao:
            client-id: {client-id 입력}
            client-secret: {client-secret 입력}
            redirect-uri: {배포도메인/api/login/oauth2/code/kakao}
            authorization-grant-type: authorization_code
            client-authentication-method: POST
            client-name: Kakao
            scope:
              - profile_nickname
              - profile_image
              - account_email

        provider:
```

```
          kakao:
            authorization-uri: https://kauth.kakao.com/oauth/authorize
            token-uri: https://kauth.kakao.com/oauth/token
            user-info-uri: https://kapi.kakao.com/v2/user/me
            user-name-attribute: id
  activemq:
    broker-url:

file:
  dir: /usr/app/pitchit-dev/

app:
  oauth2:
    authorizedRedirectUri: "{프론트엔드배포주소}/auth" # 임시

jwt:
  secret: {secret 입력}
  issuer: {발행자 입력}

openvidu:
  OPENVIDU_URL: {OpenVidu배포도메인:443/}
  OPENVIDU_RECORDING_PATH: /opt/openvidu/recordings/
  OPENVIDU_SECRET: {secret 입력}

logging:
  level:
    com.alppano.speakon: info
```

Dockerfile

```
FROM adoptopenjdk/openjdk11
WORKDIR /usr/app
COPY build/libs/*.jar pitichit_rest-api.jar
EXPOSE 8815
CMD ["java", "-jar", "pitichit_rest-api.jar"]
```

빌드 및 배포

1) Spring boot 프로젝트 빌드

```
./gradlew clean build
```

2) Docker 이미지 생성

```
docker build -t pitchit-server-v1 ./backend
```

3) 현재 돌아가고 있는 백엔드 컨테이너가 있다면 삭제

```
docker rm -f pitchit-server-v1 || true
```

4) 컨테이너 생성

```
docker run -d -p 8815:8080 -v /home/ubuntu/app/pitchit-v1:/usr/app/pitchit-v1 -v /opt/
openvidu/recordings:/opt/openvidu/recordings -e TZ=Asia/Seoul --name pitchit-server-v1
pitchit-server-v1
```