포팅메뉴얼(클론 이후 빌드 및 배포)

1. 개발환경

사용한 제품 종류 및 버전

분류	환경	버전
Database	MySQL	8.0.x
	Redis	5.0.7
Back	JAVA	Open JDK 1.8.x
	Eclipse	- Eclipse IDE 2020-06 R Package > Eclipse IDE for Enterprise Java Developers
	STS3(Spring Tools 3)	sts3 3.9.14.RELEASE
	IntelliJ	2021.3.1 (Community Edition)
	lombok	1.18.22
Front	React.js	5.0.0
	Node.js	16.13.2 LTS
	VS Code	1.64.2

2. 프로젝트 빌드 전 설치요소(특이사항)

2.1. 방화벽 설정

서비에서 사용할 포트를 미리 열어줍니다.

22/TCP	SSH
80/TCP	nginx(react)
443/TCP	OpenVidu
1443/TCP	OpenVidu
3306/TCP	MySQL
3478/TCP+UDP	OpenVidu
6379/TCP	Redis
8443/TCP	Spring Server
40000 ~ 57000/TCP+UDP	Kurento Media Server
57001 ~ 65535/TCP+UDP	OpenVidu

2.2. MySQL

```
# 설치
sudo apt-get update
sudo apt-get install mysql-server

# 실행
sudo systemctl start mysql.service

# 접속
sudo mysql

# 사용할 유저 생성 및 권한 부여
CREATE USER 계정이름@'%' IDENTIFIED BY 비밀번호;
GRANT ALL PRIVILEGES ON *.* TO 계정이름@'%' WITH GRANT OPTION;
FLUSH PRIVILEGES
```

2.3. Redis

```
# 설치 및 자동 시작
sudo apt-get update
sudo apt install redis-server

# 상태 확인
sudo system status redis-server
```

2.4. 인증서

```
# letsencrypt 설치
sudo apt-get update
sudo apt-get install letsencrypt

# 인증서 발급
# sudo letsencrypt certonly --standalone -d 도메인
sudo letsencrypt certonly --standalone -d i6e201.p.ssafy.io

# root 계정 로그인
sudo su

# 인증서 위치 폴더 이동
#cd /etc/letsencrypt/live/도메인
cd /etc/letsencrypt/live/i6e201.p.ssafy.io

# 인증서 형식 변경 pem -> PKCS12
# export용 암호 입력
openssl pkcs12 -export -in fullchain.pem \
-inkey privkey.pem \
-out key.p12 \
-name airpageserver \
-CAfile chain.pem \
-caname root

# 인증서 복사
# sudo cp [파일이름] [인증서를 보관 할 docker volume 폴더]
sudo cp fullchain.pem /home/ubuntu/docker-volume/ssl
sudo cp privkey.pem /home/ubuntu/docker-volume/ssl
sudo cp key.p12 /home/ubuntu/docker-volume/ssl
```

2.5. Docker

```
# 도커 설치
sudo apt-get update
sudo apt-get install docker-ce docker-ce-cli containerd.io
```

2.6. Docker-compose

```
# 설치
sudo apt-get update && sudo apt-get install docker-ce docker-ce-cli containerd.io
```

2.7. OpenVidu

```
# 공식 가이드를 따릅니다.
# https://docs.openvidu.io/en/stable/deployment/ce/on-premises/
```

2. 빌드 시 사용되는 환경 변수 등

2.1. Docker-compose

```
version: '3.2'

services:
  frontend:
    image: frontend-vue
    build:
      context: frontend/
      dockerfile: Dockerfile
    ports:
      - "80:80"
      - "443:443"
    # [인증서 파일 저장 경로]:/var/www/html
    volumes:
      - /home/ubuntu/docker-volume/ssl:/var/www/html
    container_name: "frontend"

  backend:
    image: backend-spring
    build:
      context: backend/
      dockerfile: Dockerfile
    ports:
      - "8443:8443"
    # [인증서 파일 저장 경로]:/root
    volumes:
      # - /home/ubuntu/WISH/ssl:/root
      - /home/ubuntu/docker-volume/ssl:/root
    container_name: "backend"
```

주목해서 봐야하는 것은 port와 volumes 입니다.
실제 사용할 포트와 인증서 경로에 맞춰 콜론(:) 기준 왼쪽 값을 변경하셔야합니다.

- port
  - 각 서비스(프론트/백)에 접근하기 위한 외부포트와 내부 포트를 의미합니다.
- volumes
  - 해당 서비스는 https연결을 요구하기 때문에 인증서가 필요합니다.
  - volumes 설정을 통해 서버(ec2)에 설치된 인증서를 사용할 수 있도록 디렉터리를 마운트합니다.

프론트엔드와 백엔드의 Dockerfile은 파일 마운트나 외부포트 설정이 없기 때문에 생략합니다.

2.2. 실행

```
docker-compose build
docker-compose up
```

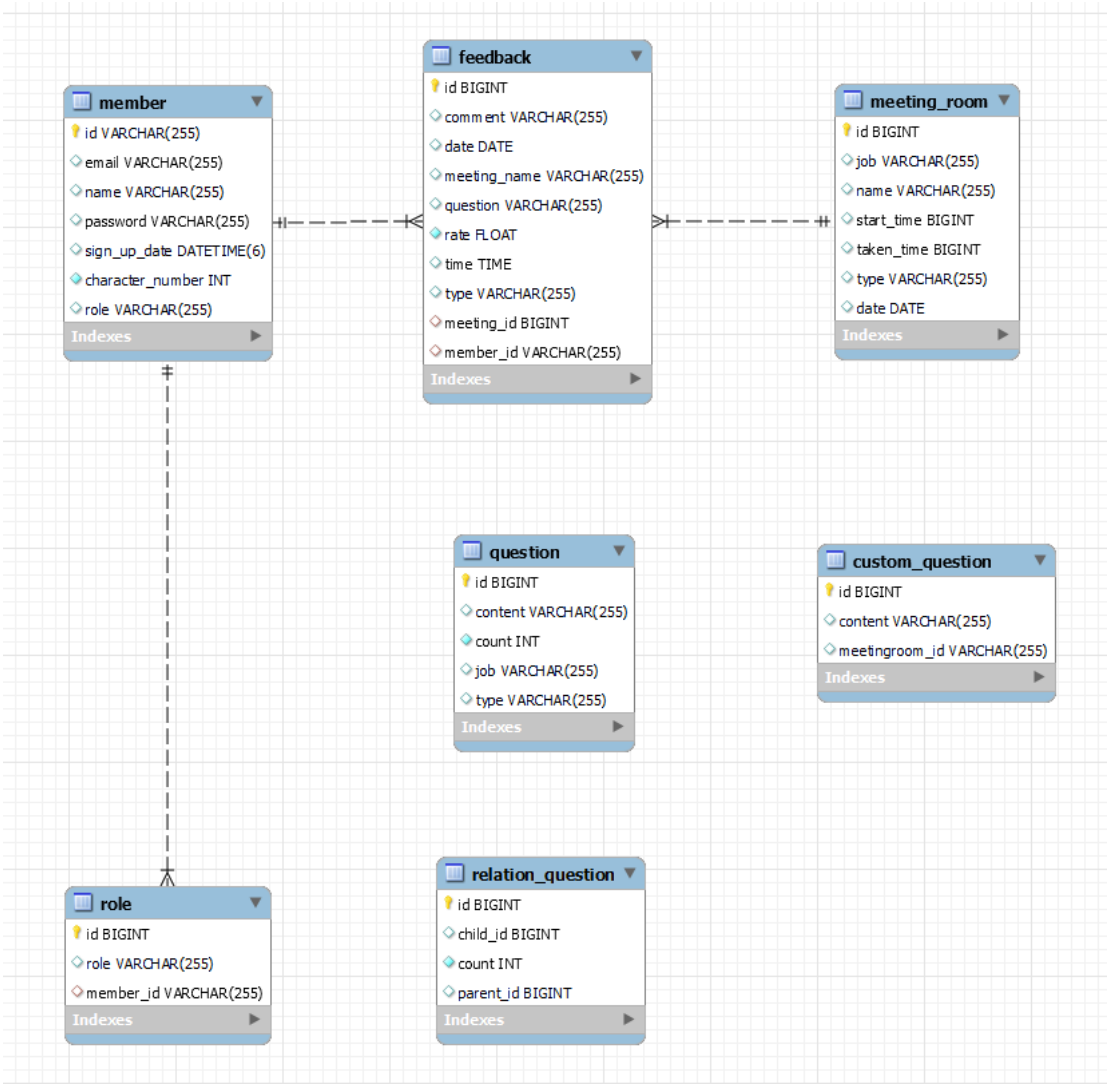
docker-compose를 사용하였기 때문에 실행시 위의 두 명령어만 docker-compose 파일이 있는 최상단 경로에서 실행하면 배포할 수 있습니다.

3. 주요 계정 및 프로퍼티 정의 파일

DB접속 정보등 프로젝트에 활용되는 주요 계정 및 프로퍼티가 정의된 파일

3.1. ERD

작업 기간이 짧은 만큼 테이블간 관계를 최소화하였습니다.



3.2. application.properties

환경변수와 관련된 세팅만 표기합니다.

```
#it will be set build date by gradle. if this value is @build.date@, front-end is development mode
build.date=@build.date@
#server.port=8080
# for SSL
server.port=서버 포트로 사용할 번호
server.http.port=8080

# 인증서
server.ssl.enabled=true
server.ssl.key-store-type=PKCS12
server.ssl.key-store=/root/key.p12
server.ssl.key-store-password=인증서 비밀번호
```

```
#database
spring.jpa.hibernate.naming.implicit-strategy=org.springframework.boot.orm.jpa.hibernate.SpringImplicitNamingStrategy
spring.jpa.hibernate.naming.physical-strategy=org.springframework.boot.orm.jpa.hibernate.SpringPhysicalNamingStrategy
spring.jpa.hibernate.ddl-auto=update
spring.jpa.properties.hibernate.dialect=org.hibernate.dialect.MySQL8Dialect
spring.data.web.pageable.one-indexed-parameters=true
spring.datasource.url=jdbc:mysql://도메인:mysql포트번호/DB이름?useUnicode=true&characterEncoding=utf8&serverTimezone=Asia/Seoul&zeroDateTir
spring.datasource.driver-class-name=com.mysql.cj.jdbc.Driver

spring.datasource.hikari.username=유저이름
spring.datasource.hikari.password=비밀번호

# 비밀번호 찾기를 위한 email
# gmail사용
spring.mail.host=smtp.gmail.com
spring.mail.port=465
spring.mail.username=이메일주소
spring.mail.password=이메일 비밀번호

# OpenVidu
# openvidu.url: https://i6e201.p.ssafy.io:1443
openvidu.url: OpenVidu 배포 주소:포트번호
openvidu.secret: Openvidu비밀번호


#Redis
spring.redis.port=레디스 포트번호
spring.redis.host=도메인
#spring.redis.password=
```