

포팅 매뉴얼

✨ 주요 기능

- 서비스 설명: 싸피인들을 위한 원스톱 채용 솔루션
- 주요 기능:
 - webRTC를 통한 실시간 면접
 - 자기소개서 관리
 - 각 회사의 리뷰 (채용 프로세스를 진행한 싸피인들이 느낀 후기들)
 - 지원자 카드 (인사 채용자가 지원자들을 한 눈에 파악할 수 있도록 지원자 정보를 카드로 정리)
 - 서류 전형 피드백 기능 (서류 전형 탈락 시 어떤 이유로 탈락되었는지 인사 담당자가 적시할 수 있음)

🖥 개발 환경

🌳 Backend

- IntelliJ
- spring boot 2.7.7
- spring-boot-jpa
- Spring Security
- Querydsl
- Java 11
- AWS EC2
- mysql 5.7.35

🌳 Frontend

- Visual Studio Code
- Vue.js 3 , Vite
- Node.js 16.19.0
- Vuetify
- SweetAlert
- Vuex

🌳 Web RTC

- openvidu 2.25

🌳 CI/CD

- aws ec2
- docker
- nginx
- jenkins

협업 툴

- Git
- Jira
- Mattermost
- Discord

기타 편의 툴

- postman

배포 설정

Git Lab과 Jenkins를 이용한 CI/CD 구축 및 Nginx와 SSL 인증서 설정

- GitLab
 - 레포지토리에 Jenkins 프로젝트 Webhook 등록

Project Hooks (1)

http://54.180.146.30:8080/project/springboot-vuejs/vuejs

Test ▼ Edit Delete

Push Events SSL Verification: enabled

- Jenkins
 - 접근 url: <http://54.180.146.30:8080/>
 - 아이디: admin, 비밀번호: 1177bcf210974e1e8bda71b843653615
- Backend
 - 젠킨스와 백엔드 gitlab 레포지토리 연결

Repository URL ?

https://lab.ssafy.com/gobuig00/back_test.git

Credentials ?

gobuig00@naver.com/***** (final)

+ Add

- 빌드 이벤트를 등록

☒ Build when a change is pushed to GitLab. GitLab webhook URL: <http://54.180.146.30:8080/project/springboot-vuejs/springboot> ?

Enabled GitLab triggers

☒ Push Events

☐ Push Events in case of branch delete

☒ Opened Merge Request Events

☐ Build only if new commits were pushed to Merge Request ?

☒ Accepted Merge Request Events

☒ Closed Merge Request Events

◦ 빌드 설정

Build Steps

≡

Invoke Gradle script ?

×

☐ Invoke Gradle ?
 ☒ Use Gradle Wrapper ?

☒ Make gradlew executable

Wrapper location ?

Tasks ?

▼

◦ 빌드된 파일을 ec2 서버 내의 도커 컨테이너로 ssh 프로토콜을 사용해 전달

SSH Publishers

SSH Server
Name ?

Spring-server

고급...

Transfers

Transfer Set
Source files ?

build/libs/*.jar

Remove prefix ?

build/libs

Remote directory ?

/spring_project/target

Exec command ?

sh ./command.sh

All of the transfer fields (except for Exec timeout) support substitution of [Jenkins environment variables](#)

고급...

- spring-server 도커에서 젠킨스를 통해 전달된 jar 파일을 실행하는 쉘스크립트

```
#!/bin/bash
echo "PID Check..."
CURRENT_PID=$(ps -ef | grep java | grep project | awk '{print $2}')
echo "Running PID: ${CURRENT_PID}"

kill -9 $CURRENT_PID
sleep 3

echo "Deploy Project...."
nohup java -jar /home/serve/spring_project/target/project-0.0.1-SNAPSHOT.jar >> /home/serve/spring_project/logs/jenkins_test_spring.log &
echo "Done"
```

- Frontend
 - 젠킨스와 프론트엔드 gitlab 레포지토리 연결

Repositories ?

Repository URL ? ✕

https://lab.ssfy.com/gobuig00/f_test.git

Credentials ?

gobuig00@naver.com/***** (final) ▼

+ Add

고급...

Add Repository

Branches to build ?

Branch Specifier (blank for 'any') ? ✕

*/master

Add Branch

Repository browser ?

(자동) ▼

Additional Behaviours

빌드 설정

빌드 환경

☐ Delete workspace before build starts

☐ Use secret text(s) or file(s) ?

☐ Provide Configuration files ?

☐ Send files or execute commands over SSH before the build starts ?

☐ Send files or execute commands over SSH after the build runs ?

☐ Add timestamps to the Console Output

☐ Inspect build log for published build scans

☒ Provide Node & npm bin/ folder to PATH

NodeJS Installation

Specify needed nodejs installation where npm installed packages will be provided to the PATH

node_16.19.0 ▼

npmrc file

- use system default - ▼

Cache location

Default (~/.npm or %APP_DATA%\npm-cache) ▼

☐ Terminate a build if it's stuck

☐ With Ant ?

빌드된 파일을 scp를 통해 전달

Build Steps

```

Execute shell ?
Command
See the list of available environment variables

cd /var/jenkins_home/workspace/springboot-vuejs/vuejs
rm -rf .git
cd ..
scp -r vuejs ubuntu@i8c208.p.ssafy.io:/home/ubuntu
  
```

• Nginx 설정

- Nginx와 letsencrypt를 이용하여 ssl 인증서 적용, 프론트엔드는 /경로, 백엔드는 /api경로로 프록시를 걸어주었음

```

server {
    location /{
        proxy_pass http://127.0.0.1:3000;
    }

    location /api {
        rewrite ^/api(.*)$ $1?$args break;
        proxy_pass http://127.0.0.1:8081;
        # proxy_pass http://localhost:8081/api;
    }

    listen 443 ssl;
    ssl_certificate /etc/letsencrypt/live/i8c208.p.ssafy.io/fullchain.pem; # managed by Certbot
    ssl_certificate_key /etc/letsencrypt/live/i8c208.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
}

server {
    if ($host = i8c208.p.ssafy.io) {
        return 301 https://$host$request_uri;
    } #managed by Certbot

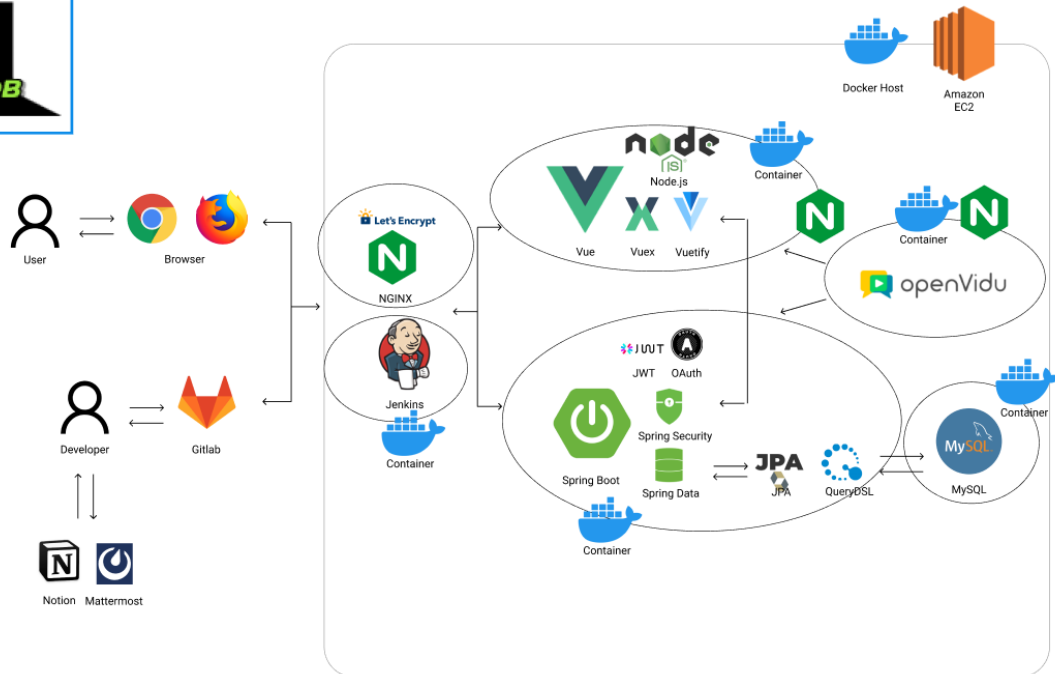
    listen 80;
    server_name i8c208.p.ssafy.io;

    return 404; # managed by Certbot
}
  
```

📀 EC2 포트 정리

| PORT | 이름 |
|------|-----------------------------------|
| 80 | HTTP-HTTPS로 리다이렉트(프론트 페이지로 리다이렉트) |
| 8080 | Jenkins |
| 3306 | MySQL |
| 8081 | Spring boot Docker Container |
| 3000 | Vue, Nginx Docker Container |
| 3478 | Openvidu |

프로젝트 EC2 인프라 구조



• Backend

- 스프링부트 서버가 동작할 도커 컨테이너(컨테이너명: spring-server)를 만든 후, 젠킨스에서 빌드한 jar 파일을 ssh 통신을 통해 보내. 새로 빌드한 파일을 컨테이너에서 받은 후, 기존 실행 중이던 spring 서버를 중단시키고 새롭게 빌드된 jar 파일을 실행하도록 해주는 `command.sh` 파일을 실행함.

MySQL 도커 컨테이너를 올린 후, spring-server와 같은 네트워크로 묶어주어 서로 통신이 가능하도록 구성.

```

Containers: {
  "1a5eac34785109be4de111d489bf03e7fb89c2bf2c381bbef9e7dc7b3ba7c13": {
    "Name": "db-mysql",
    "EndpointID": "b12190152a7aaf948657b93374a8a61dbdd1e7fa26d7199bccf20b882221098c",
    "MacAddress": "02:42:ac:12:00:03",
    "IPv4Address": "172.18.0.3/16",
    "IPv6Address": ""
  },
  "91ceccf96e9b3344979fc58cf42b7e52bd109adac88e001391b5955681e8d231": {
    "Name": "spring-server",
    "EndpointID": "04f08919f10ba56c745f4fbc5ffbad3a0005e5db6d2f09a6bf8b655376alc6e4",
    "MacAddress": "02:42:ac:12:00:02",
    "IPv4Address": "172.18.0.2/16",
    "IPv6Address": ""
  }
}

```

- 스프링 서버 도커 파일

```

FROM ubuntu:20.04

# default user
ENV USER serve

# packages install
RUN apt-get update && apt-get upgrade -y
RUN apt-get install -y sudo vim net-tools ssh openssh-server openjdk-11-jdk-headless

# Access Option
RUN sed -i 's/PermitRootLogin prohibit-password/PermitRootLogin yes/' /etc/ssh/sshd_config
RUN sed -i 's/UsePAM yes/#UserPAM yes/g' /etc/ssh/sshd_config

#user add & set
RUN groupadd -g 999 $USER
RUN useradd -m -r -u 999 -g $USER $USER

RUN sed -ri '20a'$USER'    ALL=(ALL) NOPASSWD:ALL' /etc/sudoers

#set root & user passwd
RUN echo 'root:root' | chpasswd
RUN echo $USER':serve123' | chpasswd

# java 환경 변수
ENV JAVA_HOME=/usr/lib/jvm/java-11-openjdk-amd64

ENTRYPOINT sudo service ssh restart && bash

USER $USER
~

```

- application.properties

```

#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=8081
server.address=0.0.0.0
server.servlet.contextPath=/
# Charset of HTTP requests and responses. Added to the "Content-Type" header if not set explicitly.
server.servlet.encoding.charset=UTF-8
# Enable http encoding support.
server.servlet.encoding.enabled=true
# Force the encoding to the configured charset on HTTP requests and responses.
server.servlet.encoding.force=true

server.ssl.enabled: false

OPENVIDU_URL: https://54.180.146.30:8000/
OPENVIDU_SECRET: MY_SECRET

# for SPA
#spring.web.resources.static-locations=classpath:/dist/
#spa.default-file=/dist/index.html
#spring.mvc.throw-exception-if-no-handler-found=true
#spring.web.resources.add-mappings=false

# Swagger
springfox.documentation.swagger.use-model-v3=false

#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.show-sql=true

```



```

spring.jpa.properties.hibernate.dialect=org.hibernate.dialect.MySQL5InnoDBDialect
spring.data.web.pageable.one-indexed-parameters=true
spring.datasource.url=jdbc:mysql://db-mysql:3306/mydb?useSSL=false
spring.datasource.driver-class-name=com.mysql.cj.jdbc.Driver
spring.datasource.hikari.username=ssafy
spring.datasource.hikari.password=ssafy

spring.profiles.include=secret, oauth

spring.devtools.livereload.enabled=true

#gzip compression
server.compression.enabled=true
server.compression.mime-types=application/json,application/xml,text/html,text/xml,text/plain,application/javascript,text/css

#for health check
management.servlet.context-path=/manage
management.health.db.enabled=true
management.health.default.enabled=true
management.health.diskspace.enabled=true

```

- Frontend
 - Frontend 빌드 파일을 실행시킬 Nginx 서버를 도커 컨테이너로 만들어 준 후, -v 옵션을 통해 /home/ubuntu/dist 폴더를 Nginx 컨테이너에서 사용할 수 있도록 설정. Jenkins에서 빌드된 vue 파일을 scp를 통해 보내주면 Nginx 서버가 이를 동작시킴.
- Reverse Proxy
 - EC2에 Nginx를 설치한 후, Let's Encrypt 방식으로 SSL 적용
- Openvidu server

```
curl https://s3-eu-west-1.amazonaws.com/aws.openvidu.io/install_openvidu_latest.sh | bash
```

위의 명령어를 통해 Openvidu를 위한 모든 파일을 다운받음.

```
# 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=i8c208.p.ssafy.io

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

# 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=gkatldus1@gmail.com

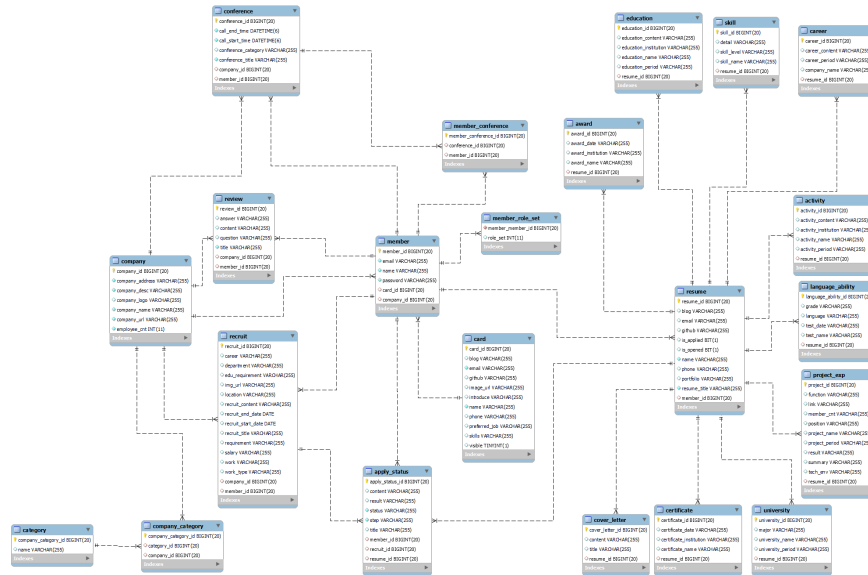
# Proxy configuration
# If you want to change the ports on which openvidu listens, uncomment the following lines

# 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=8001
```

Openvidu 설정파일을 JJOBssafy 도메인에 맞게 변경해준 후 Openvidu를 실행시켰음.

DB 설정

- username: ssafy, password: ssafy
- db erd



외부 서비스 문서

소셜 로그인

- 카카오
 - <https://developers.kakao.com/product/kakaoLogin>
- 구글
 - <https://developers.google.com/identity/sign-in/web/sign-in?hl=ko>

시연 시나리오

1. 유저, 회사, 관리자 아이디 생성해놓고 회사가 광고를 올린다.

최근 올라온 공고

전체 광고 보기

싸피 교육프로님을 모집합니다.
싸피
경기도 광주시

[삼점삼] 백엔드 엔지니어
자비스엔빌런즈(삼점삼)
서울

서비스 기획자(PM)
스와치온(SwatchOn)
서울 성동구

백엔드 개발자 (Python)
퍼알엔디컴퍼니(헤이딜러)
서울 서초구

2. 유저가 이력서를 작성하고 지원한다. 지원현황에서 지원완료 버튼 보여준다.

휴대폰
010-0000-0000

이메일
email@example.com

Github
github.com/jiang

블로그
blog.com/jiang

自我介绍书

함시연

학력사항

| 기간 | 학 교 명 | 학 과 |
|-----------|-------|--------|
| 2010~2022 | 백경대학교 | 응용언어학 |
| 2010~2022 | 백경대학교 | 중어중문학과 |

교육이수

| 교육 명 | 교육 기간 | 교육 내용 | 교육 기관 |
|-------|-----------|----------|----------|
| SSAFY | 2022~2023 | 소프트웨어 교육 | 삼성,윌티캠퍼스 |

3. 회사 계정이 지원자의 이력서를 확인하면 지원자의 지원현황탭에서 심사중으로 바뀔걸 보여주고합격했을때 결과보기 버튼으로 바뀌는걸 보여준다.

휴대폰
010-8285-2929

Skills
C, JAVA

축하드립니다

함잔나님은 sk텔레콤 구독 커머스 플랫폼 프론트엔드 기획자 서류 전형에 합격하셨습니다.
면접 일정은 추후에 따로 연락드리겠습니다.

나가기

4. 지원자가 카드정보를 기입했을때 지원자 정보 페이지에서 정보가 안뜨는걸 보여준뒤 정보 공유 승낙을 했을때 카드가 보이는걸 확인시킨뒤 호버

Applicant Information

지원자 정보를 볼 수 있는 곳입니다.

이름 보유 기술 선호 직무

| | | | |
|---|---|---|---|
|  <p>교육선</p> <p>Python, TensorFlow ★ 관심해 주십시오.</p> |  <p>김기연</p> <p>Java, Spring, JPA ★ 관심있어 도전하는 개발자 입니다.</p> |  <p>정우연</p> <p>Java, PYTHON, C ★ 관심 있어.</p> |  <p>박소나</p> <p>Vue, JREACT, NODE ★ 관심있어 주십시오.</p> |
|  <p>강아리</p> <p>Python, TensorFlow ★ 관심해 주십시오.</p> |  <p>황나나</p> <p>C, C++, Java ★ 관심있어 주십시오.</p> |  <p>김유리</p> <p>Linux, C ★ 관심있어 주십시오.</p> |  <p>황나나</p> <p>Apache, Linux ★ 관심있어 주십시오.</p> |