

Moving248 포팅 메뉴얼

Gitlab 소스 클론 이후 빌드 및 배포할 수 있도록 정리한 문서 오

1. nginx설치

```
sudo apt install nginx
```

2. ssl

Zerossl 발급 (HTTP File Upload방식)

3. 발급후 nginx ssl 적용

```
server{
    listen 80;
    server_name {도메인네임};
    return 301 https://$host$request_uri;
}

server{
    listen 443 ssl;
    server_name {도메인네임};

    ssl_certificate {crt경로};
    ssl_certificate_key {key경로};

    // 리버스프록시 설정 //
}
```

4. 도커, 도커컴포즈

```
// 도커 설치
sudo apt install docker-ce

// 도커 컴포즈 설치
sudo curl -L "https://github.com/docker/compose/releases/download/1.28.2/docker-compose-$(uname -s)-$(uname -m)" -o /usr/local/bin/docker-compose
```

5. 오픈비두 → 오픈비두 포트변경전 한번 실행 포트바꾸고 nginx로 접근해서 ssl적용

```
curl https://s3-eu-west-1.amazonaws.com/aws.openvidu.io/install_openvidu_2.28.0.sh | bash
```

```
=====
Openvidu Platform successfully installed.
=====
```

```
1. Go to openvidu folder:
$ cd openvidu

2. Configure DOMAIN_OR_PUBLIC_IP and OPENVIDU_SECRET in .env file:
$ nano .env

3. Start OpenVidu
$ ./openvidu start
```

```
docker exec -it openvidu-nginx-1 bash
```

```
etc/nginx/conf.d - default.conf
```

```
ssl_certificate /디렉토리 certificate.crt
ssl_certificate_key /디렉토리 private.key
```

docker 로컬 /opt/openvidu/certificate 아래에 볼륨으로 연결
ssl인증서를 certificate 디렉토리 아래에 넣어야
openvidu-nginx-1 컨테이너하고 볼륨을 통해 파일을 사용할 수 있다.

```
nginx reload
nginx -s reload
```

6. nodejs 최신버전으로 설치 → 리액트 빌드를 위함

```
curl -sL https://deb.nodesource.com/setup_18.x | sudo -E bash -
sudo apt install nodejs
```

7. mysql설치

```
sudo apt-get install mysql-server
```

8. redis설치

```
sudo apt-get install redis-server

bind 0.0.0.0 ::1
...(생략)
requirepass [비밀번호 지정]
...(생략)
port [포트지정]
```

9. 리액트 ec2에 올리고 빌드 → 두개올리면 프록시 location에 맞춰서 basename 설정

- 리액트 메인페이지, openvidu-cam 페이지 2개 빌드 필요
- 오픈비두 리액트 package.json 파일 "react-scripts": "5.0.1" 설정 변경 필요
- location path 와 basename이 일치해야 라우터와 정확한 연결이 가능
 - location path 가 /api2 라면 리액트 브라우저 라우터 basename = '/api2'로 일치 시켜야한다.

```
<BrowserRouter basename='/api2'>
  <Routes>
    <Route path="/" element={<VideoRoomComponent />} index />
  </Routes>
</BrowserRouter>
```

10. nginx : nginx.conf 설정 파일 리버스 프록시 설정시 root 위치 빌드 된 프로젝트 경로로 지정

```
server{
    listen 80;
    server_name i9b301.p.ssafy.io;
    return 301 https://$host$request_uri;
}

server{
    listen 443 ssl;
    server_name i9b301.p.ssafy.io;

    ssl_certificate {crt경로};
    ssl_certificate_key {key경로};

    location /api2{
        alias {리액트 빌드경로};
        index index.html index.htm;
        try_files $uri $uri/ /index.html;
        sub_filter_types text/xml text/css text/javascript;
        sub_filter static/js/ api2/static/js/;
        sub_filter href="/favicon href="api2/favicon;
        sub_filter static/css/ api2/static/css/;
        sub_filter static/media/ api2/static/media/;
        sub_filter manifest.json/ api2/manifest.json/;
        sub_filter_once off;
        sub_filter_types *;
    }
}
```

```

proxy_set_header Host $host;
proxy_set_header X-Real-IP $remote_addr;
proxy_set_header X-Forwarded-For $proxy_add_x_forwarded_for;
proxy_set_header X-Original-URI $request_uri;

}

location /{
    root    {리액트 빌드경로};
    index   index.html index.htm;
    try_files $uri $uri/ /index.html;
    proxy_set_header Host $host;
    proxy_set_header X-Real-IP $remote_addr;
    proxy_set_header X-Forwarded-For $proxy_add_x_forwarded_for;
}

location /api/{
    proxy_pass http://localhost:8080/;
    proxy_set_header Host $host;
    proxy_set_header X-Real-IP $remote_addr;
    proxy_set_header X-Forwarded-For $proxy_add_x_forwarded_for;
    proxy_set_header X-Original-URI $request_uri;
}

}

```

11. 스프링 빌드후 ec2에 배포 → ENCODE_KEY : xxxxxx 설정

```

.bashrc ENCODE_KEY export하기

source .bashrc 적용후

jar실행

```

1)사용한 JVM, 웹서버, WAS 제품 등의 종류와 설정값, 버전(IDE버전 포함) 기재

WAS : 톰캣

intelij : 2023.1 Ultimate

```

spring:
  datasource:
    driver-class-name: com.mysql.cj.jdbc.Driver
    url: {url}
    username: {DataSource-Username}
    password: {DataSource-Password}

---
spring:
  profiles:
    group:
      "local": "local, common"
      "development": "development, common"
    active: local
---
spring:
  config:
    activate:
      on-profile: "common"
  servlet:
    multipart:
      max-file-size: 5GB
      max-request-size: 5GB

```

```

jwt:
  key: {Secret-Key}
  live:
    atk: 1800000 #30분
    # atk: 30000
    rtk: 7200000 #2시간

redis:
  host: {host}
  port: {port}
  # password: your-redis-server-password (설정하지 않았다면 생략)

jpa:
  hibernate:
    ddl-auto: none
  properties:
    hibernate:
      dialect: org.hibernate.dialect.MySQLDialect
      # show_sql: true
      format_sql: true
      default_batch_fetch_size: 1000 #최적화 옵션

logging.level:
  org.hibernate.SQL: debug

logging:
  level:
    com:
      amazonaws:
        util:
          EC2MetadataUtils: error

cloud:
  aws:
    region:
      static: ap-northeast-2
  s3:
    bucket: yeonybucket
  stack:
    auto: false
  credentials:
    instance-profile: true
    access-key: ENC(ygiolVwr3vioSlXjuFEa0N3yy14Motmb7E9tZ3ZLEfk=)
    secret-key: ENC(3Ms0ma3Eh0tToXCgljBugcQru5SqMgUbB3ESnjx6PEVt/WtqCvzJzB88MjMqd7ex+Ns6M6ghrSA=)

---
# 로컬 환경
spring:
  environment: "local"
  config:
    activate:
      on-profile: "local"
  file-dir: /src/main/resources/static/files/
---
# 배포 환경
spring:
  environment: "development"
  config:
    activate:
      on-profile: "development"
  file-dir: /home/ubuntu/fileupload/
---
jaspyt:
  encryptor:
    password: ${ENCODE_KEY}

```

```

"react": "^18.2.0",
"react-dom": "^18.2.0",
"react-helmet-async": "^1.3.0",
"react-router-dom": "^6.14.2",
"react-scripts": "5.0.1",
"recoil": "^0.7.7",
"recoil-persist": "^5.1.0",
"openvidu-browser": "2.27.0",

```

2)배포 시 특이사항 기재 openvidu server docker-compose로 실행

국세청 사업자등록정보 진위확인 및 상태조회 서비스

DB 덤프 파일 최신본

Data248

시연 시나리오: 시연 순서에 따른 site 화면별, 실행별(클릭 위치 등) 상세 설명

영상 시나리오.txt

<시연 할 파트너 아이디어>

24thousand@moving248.com

1234abcd@

<시연 할 무버 아이디어>

a6@moving248.com

1234abcd@

[PC]

저희 서비스는 이사하는 사람인 무버와
이사를 도와주는 파트너로 나뉘어 있습니다.

파트너는 웹 화면을, 무버는 모바일 화면을 주로 사용할 것이라고 생각하고 반응형 웹으로 모든 디바이스에서 접근 가능하도록 구현하였습니다.

저희 서비스는 무버 회원가입과 파트너 회원가입이 따로 있는데요.
파트너의 경우 사업자가 등록된 사업장에 대해서만 회원가입이 가능하도록
사업자등록번호 검증 API를 사용하여 검증하게 했습니다.

[illegible]

메시지 화면에서 업체와 무버가 함께 소통할 수 있습니다.

영상이나 신청서에서 이사 시 부족한 점이 있다면 화상을 통해서 상담 및 보완이 가능합니다.
(화상 연결 한다.)

이사가 완료되면 자동으로 신청서가 완료 상태로 변경되고

무버 메인페이지에서 완료된 이사에 한정하여 리뷰를 등록 할 수 있습니다.

[PC]

(업체의 마이페이지로 이동)

업체의 마이 페이지에서 무버들의 리뷰를 확인할 수 있고

다른 무버가 이사 시에도 참고할 수 있습니다