[FE] 배포 및 openVidu

환경 설정

- Amazon EC2 - Ubuntu 20.04
- Nginx - 1.18.0
- docker - 20.10.17
- openvidu
  - kurento/kurento-media-server - 6.16.0
  - openvidu/openvidu-coturn - 2.22.0
  - openvidu/openvidu-proxy - 2.22.0
  - openvidu/openvidu-server - 2.22.0
  - openvidu/openvidu-call - 2.22.0

배포 방법

> 아래 ec2 서버 내에 모든 어플들을 설치 및 설정 후 빌드하면 된다

[FE] 빌드 후 배포

💡 openvidu 설치 전에 nginx를 먼저 깔지 않거나, nginx를 정지시킨 다음 진행 하는 것을 권장한다.

NGINX 설치

```
sudo apt update
# Nginx 설치
sudo apt install nginx
# 설치된 Nginx 버전확인
sudo nginx -v
```

NGINX 설정

> [FE] nginx 수동 배포(프로젝트 진행 당시)

```
cd /etc/nginx/site-available/

파일수정명령어 default
ex) nano default
```

```
server {
        listen 80 default_server;
        listen [::]:80 default_server;

        # SSL configuration
        # https포트(443) 사용하기
        listen 443 ssl default_server;
```

```
        listen [::]:443 ssl default_server;
        #
        # Note: You should disable gzip for SSL traffic.
        # See: https://bugs.debian.org/773332
        #
        # Read up on ssl_ciphers to ensure a secure configuration.
        # See: https://bugs.debian.org/765782
        #
        # Self signed certs generated by the ssl-cert package
        # Don't use them in a production server!
        #
        # include snippets/snakeoil.conf;

                #프론트 빌드 파일을 가져올 경로
        root /var/www/html/build;

        # Add index.php to the list if you are using PHP
                # 경로에 존재하는 보여줄 프론트 페이지 파일 이름
        index index.html;

                #도메인 이름
        server_name my.domain.url.com;
                #https 보안 포트 접속에 사용할 인증서 위치
                #certbot nginx설정 포함 인증서 발급시 자동으로 기입되지만
                #나는 이때 5회 발급초과여서 zerossl이라는 또다른 무료 인증서 발급사이트를
사용함
        ssl_certificate /home/ubuntu/zerossl/certificate.crt;
        ssl_certificate_key /home/ubuntu/zerossl/private.key;

                #해당 도메인(location) 뒤에 올 경로에 따라 분기
        location / {
                # First attempt to serve request as file, then
                # as directory, then fall back to displaying a 404.
                try_files $uri $uri/ /index.html;
        }
                #백엔드 경로
        location /honjaya {
                proxy_pass http://localhost:8080;
                proxy_redirect off;
                                #stomp채팅 관련되서 sockJS의 버전이 맞지않아 http 버
전을 재설정함
                proxy_http_version 1.1;
                charset utf-8;

                proxy_set_header X-Readl-IP $remote_addr;
                proxy_set_header X-Forwarded-for @proxy_add_x_forwarded_for;
                proxy_set_header X-Forwarded-Proto $scheme;
                proxy_set_header X-NginX-Proxy true;
                                #sockJS에 의한 오류 해결을 위한 추가 해더
                proxy_set_header Upgrade $http_upgrade;
                proxy_set_header Connection "upgrade";
        }

        # pass PHP scripts to FastCGI server
        #
```

```
        #location ~ \.php$ {
        #       include snippets/fastcgi-php.conf;
        #
        #       # With php-fpm (or other unix sockets):
        #       fastcgi_pass unix:/var/run/php/php7.4-fpm.sock;
        #       # With php-cgi (or other tcp sockets):
        #       fastcgi_pass 127.0.0.1:9000;
        #}

        # deny access to .htaccess files, if Apache's document root
        # concurs with nginx's one
        #
        #location ~ /\.ht {
        #       deny all;
        #}
}
```

---

docker 설치

> 도커 설치 공식 홈페이지
> https://docs.docker.com/engine/install/ubuntu/

1. Set up the repository
    - 도커 이전 버전 삭제 및 확인

```
$ sudo apt-get remove docker docker-engine docker.io containerd runc
```

    - repo구성 및 GPG 키 받아오기

```
$ sudo apt-get update
$ sudo apt-get install \
    ca-certificates \
    curl \
    gnupg \
    lsb-release
```

    - curl이 없다면, 검색해서 다운 받아서 사용

```
$ sudo mkdir -p /etc/apt/keyrings
$ curl -fsSL https://download.docker.com/linux/ubuntu/gpg | sudo gpg --
dearmor -o /etc/apt/keyrings/docker.gpg
```

```
$ echo \
  "deb [arch=$(dpkg --print-architecture) signed-
by=/etc/apt/keyrings/docker.gpg] https://download.docker.com/linux/ubuntu \
  $(lsb_release -cs) stable" | sudo tee /etc/apt/sources.list.d/docker.list
> /dev/null
```

2. Install Docker Engine
    - 설치
    - 특정 버전 설치는 공식 홈페이지 참조

```
 $ sudo apt-get update
 $ sudo apt-get install docker-ce docker-ce-cli containerd.io docker-compose-
plugin
```

---

openVidu 설치

> 추가 참조 (진행 당시 기록)
>
> [FE] OpenVidu (프로젝트 진행 당시)

> openVidu 서버
> https://docs.openvidu.io/en/2.22.0/deployment/ce/on-premises/
> openVidu 사용 코드
> https://docs.openvidu.io/en/2.22.0/tutorials/openvidu-insecure-react/

💡 도커 설치 선행 필수이며, nginx 미설치 상태 또는 정지 상태에서 깔아야한다
💡 WebRTC의 경우, local에서는 인증서 없이 영상 및 소리가 주고 받아지지만, 배포된 openVidu 서버 또는 배포된 홈페이지에서 접근할 경우 접근이 불가능할 수 있다.

포트 열기

- **22 TCP**: to connect using SSH to admin OpenVidu.
- **80 TCP**: if you select Let's Encrypt to generate an SSL certificate this port is used by the generation process.
- **443 TCP**: OpenVidu server and application are published by default in standard https port.
- **3478 TCP+UDP**: used by STUN/TURN server to resolve clients IPs.
- **40000 - 57000 TCP+UDP**: used by Kurento Media Server to establish media connections.
- **57001 - 65535 TCP+UDP**: used by TURN server to establish relayed media connections.

- 위에 해당되는 포트를 전부 열어준다.
    - 포트를 여는 명령어 : sudo ufw allow portnum
      연속된 포트를 여는 명령어 : sudo ufw allow portn1:portn2/tcp 또는 udp

설치

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

openVidu 설정

💡 배포하여 사용하려면 certbot을 통한 letsencrypt의 인증서 발급 받아야 함

```
# 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
```

```
# 사용하는 도메인
DOMAIN_OR_PUBLIC_IP=my.domain.url.com

# OpenVidu SECRET used for apps to connect to OpenVidu server and users to access
to OpenVidu Dashboard
# 오픈비두 서버와 프론트의 연결 시 비밀번호 같은 거
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.
# 로컬 개발 당시에는 selfsigned로 놓고 사용하면 경고문구만 뜨고 사용이 가능하다.
# 배포하게된다면 letsencrypt 인증서를 사용한다는 방식으로 변경해야한다.
CERTIFICATE_TYPE=letsencrypt

# If CERTIFICATE_TYPE=letsencrypt, you need to configure a valid email for
notifications
# letsencrypt 인증서 발급시 사용되는 이메일 입력
# 실질적으로 작동하는 이메일을 적으면 된다. asdf1234@naver.com 같은 것도 사용가능하지만,
# 혹시나 모를 에러를 방지하기위해 추후 발급할 certbot에서 사용하는 인증서와 같은 이메일을
쓰자
LETSENCRYPT_EMAIL=bds01088@naver.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
# openvidu 내부에 존재하는 nginx가 받아오는 포트일 듯함
# 프론트에서 실제 오픈비두 서버를 접속할때 사용되는 포트
# 주로 ssl을 사용한다면 4443포트를 사용한다.
HTTP_PORT=4442

# Changes the port of all services exposed by OpenVidu.
# SDKs, REST clients and browsers will have to connect to this port
HTTPS_PORT=4443

# Old paths are considered now deprecated, but still supported by default.
# OpenVidu Server will log a WARN message every time a deprecated path is called,
indicating
```

- 프론트에서 서버 연결 시 주소 == https://DOMAIN_OR_PUBLIC_IP:HTTPS_PORT/

인증서 발급

> certbot 공식 홈페이지
> https://certbot.eff.org/instructions?ws=nginx&os=ubuntufocal
> nginx-ubuntu20 일 때 설치 방법

💡 하나의 도메인에는 일주일에 5번만 발급이 가능하다

certbot 설치

- $ `sudo snap install core; sudo snap refresh core`
- $ `sudo snap install --classic certbot`

certbot을 통한 인증서 발급

- `sudo certbot --nginx`
  - nginx 이용 시 자동으로 https를 사용할 수 있도록 nginx에 설정을 기입해줌
- `sudo certbot renew --dry-run`
  - 단순히 인증서만 받아올 때

인증서 발급 확인

```
$ sudo su
$ cd /etc/letsencrypt/live/
```

- 해당 폴더 안에 인증서를 발급 받은 도메인의 이름으로 폴더가 존재한다면 성공한 것임

---

openVidu 해결하기 어려웠던 문제점

💡 letsencrypt를 통한 인증서 발급 완료 및 openvidu의 인증 방식을 letsencrypt로 설정하였음에도 인증서 오류로 인한 배포된 환경에서 실행이 불가능할 경우
💡 또한 docker-compose logs nginx를 통해 실행 중인 openvidu nginx의 로그를 보고 그 중 인증서를 찾지 못했다는 에러를 발견했을 경우

> 해결해주신 안XX님 감사합니다

```
cd /opt/openvidu/
nano docker-compose.yml
```

```
nginx:
        image: openvidu/openvidu-proxy:2.22.0
        restart: always
        network_mode: host
        volumes:
            - /etc/letsencrypt:/etc/letsencrypt
            - ./owncert:/owncert
            - ./custom-nginx-vhosts:/etc/nginx/vhost.d/
            - ./custom-nginx-locations:/custom-nginx-locations
            - ${OPENVIDU_RECORDING_CUSTOM_LAYOUT}:/opt/openvidu/custom-layout
        environment:
            - DOMAIN_OR_PUBLIC_IP=${DOMAIN_OR_PUBLIC_IP}
            - CERTIFICATE_TYPE=${CERTIFICATE_TYPE}
            - LETSENCRYPT_EMAIL=${LETSENCRYPT_EMAIL}
```

```
      - PROXY_HTTP_PORT=${HTTP_PORT:-}
      - PROXY_HTTPS_PORT=${HTTPS_PORT:-}
      - PROXY_HTTPS_PROTOCOLS=${HTTPS_PROTOCOLS:-}
      - PROXY_HTTPS_CIPHERS=${HTTPS_CIPHERS:-}
      - PROXY_HTTPS_HSTS=${HTTPS_HSTS:-}
```

- 위 구조에서 volumes에 letsencrypt를 위와 같이 수정해주면 될 듯하다
  현재 수정되어서 이전에 어떤 형태로 기입되어 있는지 기억은 안난다.
- 해당 openvidu nginx가 ec2서버 내에 인증서를 docker 내부까지 매핑을 하지 못해서 발생하는 에러이므로 강제로 주소를 하드코딩? 했던 것으로 기억한다.