[SSL] Nginx + Certbot을 이용한 SSL인증서 발급

nginx와 certbot을 docker container로 띄워서 https 설정을 위한 ssl 인증서 발급받기

1. 기본적으로 docker와 docker compose가 설치되어 있어야 한다.

```
docker compose version
docker --version
```

2. `docker-compose.yml` 파일을 작성해 nginx와 certbot 컨테이너 설정을 해준다.

```
version : '3.8'

services:
  nginx:
    container_name: webserver-nginx
    image: nginx:latest
    restart: always
    #외부와 소통하는 경우에는 ports로 포트 설정.
    ports:
      - "80:80"
      - "443:443"
    volumes:
      # 현재 경로 기준 proxy 폴더에 존재하는 nginx.conf
      - ./proxy/nginx.conf:/etc/nginx/nginx.conf
      - ./proxy/dev-nginx.conf:/etc/nginx/nginx.conf
      - ./proxy/prod-nginx.conf:/etc/nginx/nginx.conf
      # certbot의 volum 경로와 맞춰준다.
      - ./data/certbot/conf:/etc/letsencrypt
      - ./data/certbot/www:/var/www/certbot
```

```
    environment:
      - TZ=Asia/Seoul
    # 인증서 갱신을 위한 명령어
    command: "/bin/sh -c 'while :; do sleep 6h & wait $${!}; nginx -s reload; done & nginx
-g \"daemon off;\"'"


  certbot:
    container_name: webserver-certbot
    image: certbot/certbot
    restart: unless-stopped
    volumes:
      - ./data/certbot/conf:/etc/letsencrypt
      - ./data/certbot/www:/var/www/certbot
    # 인증서 갱신을 위한 명령어
    entrypoint: "/bin/sh -c 'trap exit TERM; while :; do certbot renew; sleep 12h & wait
 $${!}; done;'"
```

- certbot을 통해 발급 받은 ssl 인증서는 기본적으로 3개월마다 갱신해야 하므로 자동 갱신을 위한 명령어를 추가해주었다.

3. proxy 디렉토리에 nginx.conf 파일 만들어서 작성

```
mkdir proxy

cd proxy

sudo vim nginx.conf
```

```
# nginx.conf 파일포함 내용

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

     server_name 도메인_주소; # 등록한 도메인으로 변경

    # 인증서 발급 요청시 도메인_주소/well-known/acme-challenge으로 요청
     location /.well-known/acme-challenge/ {
             allow all;
             root /var/www/certbot;
     }
}
```

3. docker compose 실행

```
docker-compose -f docker-compose.yml up -d // -f 파일 옵션 -d daemon 실행
docker ps # nginx와 certbot 컨테이너가 살아있는지 확인
```

4. 인증서 발급을 위한 `init-letsencrypt.sh` 쉘 스크립트 실행

```
# git에서 init-letsencrypt.sh 파일 내려받기
# 아래 이슈사항에서 공유된 파일 내용으로 init-letsencrypt.sh 작성 후 실행 가능
curl -L https://raw.githubusercontent.com/wmnnd/nginx-certbot/master/init-letsencrypt.sh >
init-letsencrypt.sh
# 실행 권한 부여
sudo chmod +x init-letsencrypt.sh
# 쉘 스크립트 실행
sudo ./init-letsencrypt.sh
```

🚨 ***** 이슈 사항 공유 *****

- **init-letsencrypt.sh 문서에서 `docker-compose`라고 명기되어 있는 부분은 모두 `docker compose`로 수정을 해주어야 정상 작동함**

▼ **수정 파일 공유 (** 🖱️ **click!)**

```
#!/bin/bash

domains=(도메인 주소)
rsa_key_size=4096
data_path="./data/certbot"
email="이메일 주소" # Adding a valid address is strongly recommended
staging=0 # Set to 1 if you're testing your setup to avoid hitting request
limits

if [ -d "$data_path" ]; then
  read -p "Existing data found for $domains. Continue and replace existing
certificate? (y/N) " decision
  if [ "$decision" != "Y" ] && [ "$decision" != "y" ]; then
    exit
  fi
fi


if [ ! -e "$data_path/conf/options-ssl-nginx.conf" ] || [ ! -e "$data_pat
h/conf/ssl-dhparams.pem" ]; then
  echo "### Downloading recommended TLS parameters ..."
  mkdir -p "$data_path/conf"
  curl -s https://raw.githubusercontent.com/certbot/certbot/master/certbot
-nginx/certbot_nginx/_internal/tls_configs/options-ssl-nginx.conf > "$data
_path/conf/options-ssl-nginx.conf"
  curl -s https://raw.githubusercontent.com/certbot/certbot/master/certbo
t/certbot/ssl-dhparams.pem > "$data_path/conf/ssl-dhparams.pem"
  echo
fi

echo "### Creating dummy certificate for $domains ..."
path="/etc/letsencrypt/live/$domains"
mkdir -p "$data_path/conf/live/$domains"
docker compose run --rm --entrypoint "\
  openssl req -x509 -nodes -newkey rsa:$rsa_key_size -days 1\
    -keyout '$path/privkey.pem' \
    -out '$path/fullchain.pem' \
    -subj '/CN=localhost'" certbot
echo


echo "### Starting nginx ..."
docker compose up --force-recreate -d nginx
```

```
echo

echo "### Deleting dummy certificate for $domains ..."
docker compose run --rm --entrypoint "\
  rm -Rf /etc/letsencrypt/live/$domains && \
  rm -Rf /etc/letsencrypt/archive/$domains && \
  rm -Rf /etc/letsencrypt/renewal/$domains.conf" certbot
echo


echo "### Requesting Let's Encrypt certificate for $domains ..."
#Join $domains to -d args
domain_args=""
for domain in "${domains[@]}"; do
  domain_args="$domain_args -d $domain"
done

# Select appropriate email arg
case "$email" in
  "") email_arg="--register-unsafely-without-email" ;;
  *) email_arg="--email $email" ;;
esac

# Enable staging mode if needed
if [ $staging != "0" ]; then staging_arg="--staging"; fi

docker compose run --rm --entrypoint "\
  certbot certonly --webroot -w /var/www/certbot \
    $staging_arg \
    $email_arg \
    $domain_args \
    --rsa-key-size $rsa_key_size \
    --agree-tos \
    --force-renewal" certbot
echo

echo "### Reloading nginx ..."
docker compose exec nginx nginx -s reload
```

- **docker-compose.yml 파일과 `동일한 위치`에 존재해야 정상 작동함**

5. `nginx.conf` 파일 수정

```
user  nginx;
worker_processes  auto;
error_log  /var/log/nginx/error.log warn;
pid        /var/run/nginx.pid;
```

```
events {
    worker_connections  1024;
}

http{

    include       /etc/nginx/mime.types;
    default_type  application/octet-stream;

    server {
        listen 80;
        server_name 도메인_주소;
        server_tokens off;

        location /.well-known/acme-challenge/ {
            allow all;
            root /var/www/certbot;
        }

        # 만약, 80번 포트(http)로 요청이 오면, https로 리디렉션
        location / {
            return 301 https://$host$request_uri;
        }
    }

    server {
        listen 443 ssl;
        server_name 도메인_주소;
        server_tokens off;

        # 인증서 위치
        ssl_certificate /etc/letsencrypt/live/도메인_주소/fullchain.pem;
        ssl_certificate_key /etc/letsencrypt/live/도메인_주소/privkey.pem;
        include /etc/letsencrypt/options-ssl-nginx.conf;
        ssl_dhparam /etc/letsencrypt/ssl-dhparams.pem;

    }

    log_format  main  '$remote_addr - $remote_user [$time_local] "$request" '
                      '$status $body_bytes_sent "$http_referer" '
                      '"$http_user_agent" "$http_x_forwarded_for"';

    access_log  /var/log/nginx/access.log  main;

    sendfile        on;
    keepalive_timeout  65;
    include /etc/nginx/conf.d/*.conf;

}
```