목차

Let's Encrypt(CertBot) SSL with HAProxy

Let's Encrypt에서 제공하는 Certbot을 이용한 SSL사용하는경우가 많다. 매우 편리하게 SSL인증서를 발급받아 사용할수 있어 많은 곳에서 사용중이다. 다만 HAProxy 를 사용하는 경우 발급/갱신시 haproxy로 인해 바로 처리가 되지 않는 상황이 있을수 있는데 이경우 아래와 같이 해결하도록 한다.

HAProxy 설정

새 인증서를 요청하면 LetsEncrypt가 인증 파일 (URI: /.well-known/acme-challenge/random-hash-here)을 요청합니다. 인증서 설정이 아직 없기 때문에 이 요청은 포트 80을 통해 발생합니다.

만약 HAProxy가 포트 443에서 수신 대기 중이면 LetsEncrypt가 해당 포트에 대해 권한 부여를 시도할 수 있습니다.(80포트를 사용하지 않기 때문에) 따라서 새 인증서를 만들 때 포트 80에서만 수신 대기하는 HAProxy가 필요합니다.

또 다른 문제 : HAProxy가 포트 80에서 수신 대기 중입니다. 그러나 권한 부여 요청을 수신하도록 독립형 서버를 설정하려면 LetsEncrypt가 필요합니다. 기본적으로 포트 80도 설정되어 한 번에 하나의 프로세스 만 포트에서 수신 할 수 있으므로 충돌이 발생합니다. 따라서 다른 포트에서 수신하도록 LetsEncrypt에 알려야합니다!

HAProxy 내에서 들어오는 HTTP 요청에 /.well-known/acme-challenge 문자열이 포함되어 있는지 물어볼 수 있습니다. 아래 구성에서 HAProxy가 요청에 해당 URI가 포함되어 있음을 확인하면 요청을 LetsEncrypt로 라우팅합니다. 그렇지 않으면 요청이 정상적으로로드 밸런서 회전의 서버로 라우팅됩니다.

```
# The frontend only listens on port 80
# If it detects a LetsEncrypt request, is uses the LE backend
# Else it goes to the default backend for the web servers
frontend fe-scalinglaravel
    bind *:80

    # Test URI to see if its a letsencrypt request
    acl letsencrypt-acl path_beg /.well-known/acme-challenge/
    use_backend letsencrypt-backend if letsencrypt-acl

    default_backend be-scalinglaravel

# LE Backend
backend letsencrypt-backend
    server letsencrypt 127.0.0.1:8888

# Normal (default) Backend
# for web app servers
backend be-scalinglaravel
    # Config omitted here
```

HAProxy 내에서 설정 한 후에는 다시로드 (sudo service haproxy reload) 한 다음 LetsEncrypt 실행으로

넘어갈 수 있습니다.

새로운 인증서

LetsEncrypt에서 새 인증서를 가져 오는 명령은 다음과 같습니다.

```
sudo certbot certonly --standalone -d demo.scalinglaravel.com \
    --non-interactive --agree-tos --email admin@example.com \
    --http-01-port=8888
```

http-01-port=8888 옵션을 사용함으로 인해서 이후 갱신시에도 certbot이 8888로 수신 대기 상태를 시도합니다.

인증서 HAProxy용 통합파일 구성

인증서 파일들이 생성된 이후 haproxy에 등록해서 사용할때 각각의 파일이 아닌 하나로 통합하여 등록해 사용하는경우 파일을 하나로 합쳐주는 작업이 필요하다. 본인은 아래 스크립트를 만들어서 활용한다.

```
#!/bin/bash
# 갱신된 인증서 haproxy로 복제
cat /etc/letsencrypt/live/plex.koov.net/cert.pem
/etc/letsencrypt/live/plex.koov.net/privkey.pem
/etc/letsencrypt/live/plex.koov.net/chain.pem >
/etc/haproxy/ssl/plex.koov.net.pem
cat /etc/letsencrypt/live/allthatlinux.com/cert.pem
/etc/letsencrypt/live/allthatlinux.com/privkey.pem
/etc/letsencrypt/live/allthatlinux.com/chain.pem >
/etc/haproxy/ssl/allthatlinux.com.pem
cat /etc/letsencrypt/live/linuxdata.kr/cert.pem
/etc/letsencrypt/live/linuxdata.kr/privkey.pem
/etc/letsencrypt/live/linuxdata.kr/chain.pem >
/etc/haproxy/ssl/linuxdata.kr.pem
cat /etc/letsencrypt/live/nas.koov.net/cert.pem
/etc/letsencrypt/live/nas.koov.net/privkey.pem
/etc/letsencrypt/live/nas.koov.net/chain.pem >
/etc/haproxy/ssl/nas.koov.net.pem

# 생성한 후 haproxy 서비스 재기동
systemctl restart haproxy
```

만들어진 스크립트는 인증서가 갱신될때 실행되어야 하기 때문에 아래와 같이 기본 설치된 certbot cron 스크립트 내에 추가하도록 한다.

```
# /etc/cron.d/certbot: crontab entries for the certbot package
```

```
#
# Upstream recommends attempting renewal twice a day
#
# Eventually, this will be an opportunity to validate certificates
# haven't been revoked, etc.  Renewal will only occur if expiration
# is within 30 days.
#
# Important Note!  This cronjob will NOT be executed if you are
# running systemd as your init system.  If you are running systemd,
# the cronjob.timer function takes precedence over this cronjob.  For
# more details, see the systemd.timer manpage, or use systemctl show
# certbot.timer.
SHELL=/bin/sh
PATH=/usr/local/sbin:/usr/local/bin:/sbin:/bin:/usr/sbin:/usr/bin

0 */12 * * * root test -x /usr/bin/certbot -a \! -d /run/systemd/system &&
perl -e 'sleep int(rand(43200))' && certbot -q renew

### 이곳에 추가한다.
10 */12 * * * root /etc/letsencrypt/copy_ssl.sh
```

참조링크

- https://stevenwestmoreland.com/2017/11/renewing-certbot-certificates-using-a-systemd-timer.html
- https://wiki.debianusers.or.kr/index.php?title=Certbot