

- Let's Encrypt(CertBot) SSL with HAProxy** ..... 3
- HAProxy** ..... 3
- ..... 4
- HAProxy** ..... 4
- ..... 5



# Let's Encrypt(CertBot) SSL with HAProxy

Let's Encrypt Certbot SSL 가 . SSL  
haproxy 가 HAProxy /

## HAProxy

hash-here) LetsEncrypt가 (URI: /.well-known/acme-challenge/random-80

HAProxy가 443 LetsEncrypt가 ) 80  
.(80 )  
HAProxy가

: HAProxy가 80  
LetsEncrypt가 80

LetsEncrypt !

HAProxy HTTP /.well-known/acme-challenge  
HAProxy가 URI가

LetsEncrypt

```
# 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 ( sudo service haproxy reload) LetsEncrypt

LetsEncrypt 가

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

http-01-port=8888

certbot 8888

## HAProxy

haproxy

```
#!/bin/bash  
# haproxy  
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  
  
# haproxy  
systemctl restart haproxy
```

가

certbot cron

```
# /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# haven't been revoked, etc.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

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

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

From:  
<https://atl.kr/dokuwiki/> - AllThatLinux!

Permanent link:  
[https://atl.kr/dokuwiki/doku.php/let\\_s\\_encrypt\\_certbot\\_ssl\\_with\\_haproxy](https://atl.kr/dokuwiki/doku.php/let_s_encrypt_certbot_ssl_with_haproxy)

Last update: **2020/02/21 07:01**

