Let's Encrypt

- Follow Automate HAProxy Let's Encrypt certificates with acme.sh
- You can use my sample HAProxy Configuration and remove double-comments ## after replacing the account thumbprint

Install acme.sh

1. First, on the HAProxy server, create the acme user:

```
$ sudo adduser \
   --system \
   --disabled-password \
   --disabled-login \
   --home /var/lib/acme \
   --quiet \
   --force-badname \
   --group \
   acme
```

2. Add the acme user to the HAProxy group:

```
$ sudo adduser acme haproxy
```

3. Create the directory where we will install the acme.sh program:

```
$ sudo mkdir /usr/local/share/acme.sh/
```

4. Then install acme.sh:

```
$ git clone https://github.com/acmesh-official/acme.sh.git
$ cd acme.sh/
$ sudo ./acme.sh \
   --install \
   --no-cron \
   --no-profile \
   --home /usr/local/share/acme.sh
$ sudo ln -s /usr/local/share/acme.sh/acme.sh /usr/local/bin/
$ sudo chmod 755 /usr/local/share/acme.sh/
```

Generate your ACME account

1. To generate your ACME account, switch to the acme user so the ACME account info will be stored in this user's home directory:

```
$ sudo -u acme -s
```

```
$ acme.sh --register-account \
   --server letsencrypt_test \
   -m youremail@example.com

[Mon Apr 24 01:28:14 PM UTC 2023] Create account key ok.
[Mon Apr 24 01:28:14 PM UTC 2023] Registering account:
https://acme-staging-v02.api.letsencrypt.org/directory

[Mon Apr 24 01:28:14 PM UTC 2023] Registered
[Mon Apr 24 01:28:14 PM UTC 2023]
ACCOUNT_THUMBPRINT='lCufto4sDRTHdmWL0EugFywGV54hBCuTTXvwifi65R4'

$ exit
```

2. Take a note of ACCOUNT_THUMBPRINT as you need this value when setting up certificate generation.
3. Note that we have set the server where we'd like to register an account to be `letsencrypt_test`, which is the Let's Encrypt staging server. It's a good idea to use this value while you test your setup. Once you've verified that everything is working, register again using a server value of `letsencrypt`.

Configure HAProxy to Respond to HTTP Challenges

1. Start by creating the directory for certificates:

```
$ sudo mkdir /etc/haproxy/certs
$ sudo chown haproxy:haproxy /etc/haproxy/certs
$ sudo chmod 770 /etc/haproxy/certs
```

2. Edit `/etc/haproxy/haproxy.cfg` to add the challenge response and the thumbprint:

```
global
    stats socket /var/run/haproxy/admin.sock level admin mode 660
    setenv ACCOUNT_THUMBPRINT
'lCufto4sDRTHdmWL0EugFywGV54hBCuTTXvwifi65R4'

frontend web
    bind :80
    bind :443 ssl crt /etc/haproxy/certs/ strict-sni
    http-request return status 200 content-type text/plain lf-string
"%[path,field(-1,/)].${ACCOUNT_THUMBPRINT}\n" if { path_beg '/.well-
known/acme-challenge/' }
```

3. Restart the HAProxy service:

```
$ sudo systemctl restart haproxy
```

Generate and deploy certificate

1. Generate a certificate (once tested use `letsencrypt` instead of `letsencrypt_test`):

```
$ sudo -u acme -s
$ acme.sh --issue \
   -d example.com \
   --stateless \
   --server letsencrypt_test

[Mon Apr 24 01:36:03 PM UTC 2023] Using CA:
https://acme-staging-v02.api.letsencrypt.org/directory

[Mon Apr 24 01:36:03 PM UTC 2023] Single domain=example.com'
[Mon Apr 24 01:36:03 PM UTC 2023] Getting domain auth token for each
domain
[Mon Apr 24 01:36:04 PM UTC 2023] Getting webroot for
domain='example.com'
[Mon Apr 24 01:36:04 PM UTC 2023] Verifying: example.com
[Mon Apr 24 01:36:04 PM UTC 2023] Stateless mode for domain:example.com
[Mon Apr 24 01:36:06 PM UTC 2023] Success
[Mon Apr 24 01:36:06 PM UTC 2023] Verify finished, start to sign.
[Mon Apr 24 01:36:06 PM UTC 2023] Lets finalize the order.
[Mon Apr 24 01:36:06 PM UTC 2023]
Le_OrderFinalize='https://acme-staging-v02.api.letsencrypt.org/acme/fin
alize/12345'
[Mon Apr 24 01:36:06 PM UTC 2023] Downloading cert.
[Mon Apr 24 01:36:06 PM UTC 2023]
Le_LinkCert='https://acme-staging-v02.api.letsencrypt.org/acme/cert/abc
12345'
[Mon Apr 24 01:36:06 PM UTC 2023] Cert success.
-----BEGIN CERTIFICATE-----
[...]
-----END CERTIFICATE-----
[Mon Apr 24 01:36:06 PM UTC 2023] Your cert is in:
/var/lib/acme/.acme.sh/example.com_ecc/example.cer
[Mon Apr 24 01:36:06 PM UTC 2023] Your cert key is in:
/var/lib/acme/.acme.sh/example.com_ecc/example.key
[Mon Apr 24 01:36:06 PM UTC 2023] The intermediate CA cert is in:
/var/lib/acme/.acme.sh/example.com_ecc/ca.cer
[Mon Apr 24 01:36:06 PM UTC 2023] And the full chain certs is there:
/var/lib/acme/.acme.sh/example.com_ecc/fullchain.cer
```

2. I have written a small bash file so I don't need to enter the command parameters every time

```
sudo vim /var/lib/acme/batch/generate.sh
#!/bin/bash
#
# generate certificate for $1 domain and $2...$n alternative domains
if [ $# -eq 0 ]; then
```

```
    echo "usage: $0 <domain> <alternative domain 1>...<alternative
domain n>"
else
    COMMAND=(acme.sh "--issue")
    for arg in "$@"; do
        COMMAND+=("-d" ${arg})
    done
    COMMAND+=("--stateless" "--server letsencrypt")
    "${COMMAND[@]}"
fi
:wq
sudo chown acme:acme /var/lib/acme/batch/generate.sh
ln -s /var/lib/acme/batch/generate.sh /usr/local/bin/generate.sh
```

3. Once the certificate is generated, you can deploy it to the /etc/haproxy/certs directory and enable it in HAProxy without needing to reload:

```
$ sudo -u acme -s
$ DEPLOY_HAPROXY_HOT_UPDATE=yes \
  DEPLOY_HAPROXY_STATS_SOCKET=/run/haproxy/admin.sock \
  DEPLOY_HAPROXY_PEM_PATH=/etc/haproxy/certs \
  acme.sh --deploy -d example.com --deploy-hook haproxy

[Mon Apr 24 02:17:31 PM UTC 2023] The domain 'example.com' seems to
have a ECC cert already, lets use ecc cert.
[Mon Apr 24 02:17:31 PM UTC 2023] Deploying PEM file
[Mon Apr 24 02:17:31 PM UTC 2023] Moving new certificate into place
[Mon Apr 24 02:17:31 PM UTC 2023] Creating new certificate
'/etc/haproxy/certs/example.com.pem' over HAProxy stats socket.
[Mon Apr 24 02:17:31 PM UTC 2023] Success
```

4. I have written a small bash file so I don't need to enter the command parameters every time

```
sudo vim /var/lib/acme/batch/deploy.sh
#!/bin/bash
#
# deploy certificate for $1 domain
if [ $# -eq 0 ]; then
    echo "usage: deploy.sh <domain>"
else
    DEPLOY_HAPROXY_HOT_UPDATE=yes \
    DEPLOY_HAPROXY_STATS_SOCKET=/run/haproxy/admin.sock \
    DEPLOY_HAPROXY_PEM_PATH=/etc/haproxy/certs \
    acme.sh --deploy -d $1 --deploy-hook haproxy
fi
:wq
sudo chown acme:acme /var/lib/acme/batch/deploy.sh
ln -s /var/lib/acme/batch/deploy.sh /usr/local/bin/deploy.sh
```

5. Check if the certificate was added in HAProxy with socat:

```
$ echo "show ssl cert /etc/haproxy/certs/example.com.pem" |\
```

```
   socat /var/run/haproxy/admin.sock -

Filename: /etc/haproxy/certs/example.com.pem
Status: Used
Serial: 6C058BA29A03DF58BFA9CF0ABD0C0AB8
notBefore: Apr 24 13:35:04 2023 GMT
notAfter: Apr 25 13:36:04 2023 GMT
Subject Alternative Name: DNS:example.com
Algorithm: EC256
SHA1 FingerPrint: 7A6A64B2D3BB1E548DBF383FDB3A7CA8434B3F5A
[...]
```

6. If the dry run works go back to account registration and do everything with `letsencrypt` production server.

Certificate renewal

1. Install a crontab directive for the acme user:

```
$ sudo -u acme -s
$ crontab -e
$ acme.sh --install-cronjob
```

Check certificates

- List All Certificates:

```
acme.sh --list
```

- View Details for a Specific Domain:

```
acme.sh --info -d domain.com
```

- Check Live Certificate Expiration (OpenSSL):

```
openssl x509 -in ~/.acme.sh/domain.com_ecc/domain.com.cer -noout -text
| grep -A 2 "Validity"
```

Links

- ACME Protocol Explained
- Automate HAProxy Let's Encrypt certificates with acme.sh

From:
https://wiki.condrau.com/ - **Bernard's Wiki**

Permanent link:
https://wiki.condrau.com/deb13:acme

Last update: **2026/09/20 12:26**