Problem statement :

Install haproxy with self signed SSL certificate at Linux

Solution High level steps-

1. Update and install openssl
2. Create HAProxy Repository
3. install HAProxy
4. Generate self sign certificate and private key
5. Create SSL pem file by containing both the key and the certificate
6. configure haproxy.cfg file
7. Restart haproxy service
8. Troubleshoot

Implementation

Below step applied on ubuntu 18.04 platform

Update and install openssl::

apt-get update

apt-get -y install openssl

Create HAProxy Repository

apt install curl -y

curl https://haproxy.debian.net/bernat.debian.org.gpg | apt-key add -

echo "deb http://haproxy.debian.net $(lsb_release -cs)-backports-2.0 main" | tee /etc/apt/sources.list.d/haproxy.list

apt install software-properties-common

add-apt-repository ppa:vbernat/haproxy-2.0

Once the repos are created on each system, perform system update and install HAProxy.

apt update

apt install haproxy=2.0.*

haproxy -v //check version

Generating Self-Signed SSL Certificates for HAProxy Begin with generating private key::

openssl genrsa -out /etc/ssl/private/haproxy.key 2048

Next, generate the Certificate signing request (CSR)::

openssl req -new -key /etc/ssl/private/haproxy.key -out /etc/ssl/certs/haproxy.csr

Create the Self Signed Certificate (CRT)::

openssl x509 -req -days 365 -in /etc/ssl/certs/haproxy.csr -signkey /etc/ssl/private/haproxy.key -out /etc/ssl/certs/haproxy.crt

Create SSL pem file by containing both the key and the certificate::

cat /etc/ssl/private/haproxy.key /etc/ssl/certs/haproxy.crt >> /etc/ssl/certs/haproxy.pem

configure haproxy.cfg file:::

here at bind section specify the pem cert file location as showing yellow marked. Haprozy cfg file other file as usual

vi /etc/haproxy/haproxy.cfg

```
global
        log /dev/log    local0
        log /dev/log    local1 notice
        chroot /var/lib/haproxy
        stats socket /run/haproxy/admin.sock mode 660 level admin expose-fd
listeners
        stats timeout 30s

defaults
        log     global
        mode    http
        option  httplog
        option  dontlognull
        timeout connect 5000
```

timeout client 50000
timeout server 50000

frontend website
bind :80
bind :443 ssl crt /etc/ssl/certs/haproxy.pem
default_backend servers

backend servers
balance roundrobin
server serv1 192.168.0.1:8080
server serv2 192.168.0.2:8080
save and exit

Running HAProxy:::

When installed, HAProxy is set to run by default. To restart and enable HAProxy to run on system boot;

```
systemctl restart haproxy
```

```
systemctl enable haproxy
```

To check the status;

```
systemctl status haproxy
```

● haproxy.service - HAProxy Load Balancer

Loaded: loaded (/lib/systemd/system/haproxy.service; enabled; vendor preset: enabled)

Active: active (running) since Wed 2020-11-04 11:39:39 UTC; 32s ago

Docs: man:haproxy(1)

file:/usr/share/doc/haproxy/configuration.txt.gz

Main PID: 12979 (haproxy)

Tasks: 2 (limit: 1140)

CGroup: /system.slice/haproxy.service

├─12979 /usr/sbin/haproxy -Ws -f /etc/haproxy/haproxy.cfg -p /run/haproxy.pid -S /run/haprox

└─12981 /usr/sbin/haproxy -Ws -f /etc/haproxy/haproxy.cfg -p /run/haproxy.pid -S /run/haprox

Nov 04 11:39:39 ip-10-0-0-101 systemd[1]: Starting HAProxy Load Balancer...

Now browse loadbalancer URL IP to check https://IP

Troubleshoot :

run the command below to check the HAProxy configuration for any error.

haproxy -c -f /etc/haproxy/haproxy.cfg

Future work:

Install this setup at docker container