

DOCUMENTATION

HAPROXY WEB

HAProxy est un logiciel open-source de répartition de charge et de proxy TCP/HTTP. Il est largement utilisé pour améliorer les performances et la fiabilité des applications web en distribuant le trafic entre plusieurs serveurs.

HAProxy offre également des fonctionnalités avancées telles que l'équilibrage de charge, la haute disponibilité et la terminaison SSL.



Sommaire

Définition

Prérequis

Installation d'un Load Balancer (HaProxy)

Configuration pour 2 serveurs web

Définition

HaProxy est un logiciel qui aide les sites web à fonctionner plus rapidement et de manière plus fiable en répartissant intelligemment le trafic des visiteurs sur plusieurs serveurs. Cela garantit que les sites web restent accessibles même en cas de forte affluence ou de problèmes techniques sur l'un des serveurs. En résumé, HaProxy améliore la performance et la disponibilité des sites web. On appelle cela un Load Balancer.

Prérequis

- - Un système d'exploitation supportant HaProxy, dans ce cas j'utilise un serveur Debian
- Un hyperviseur de type 1 afin d'installer nos machines virtuelles. (Dans mon cas j'utilise Proxmox VE)
- Quatre machines virtuelles sur Proxmox (2 serveurs proxy + 2 serveurs web)

Étape 1 : Installation et configuration IP de HaProxy et Nginx

1. Dans un premier temps nous allons installer le service HaProxy sur les serveurs proxy

```
root@serveurProxy:~# apt install haproxy
```

2. Puis installer le service Nginx sur les serveurs web

```
root@web1:~# apt install nginx -y_
```

3. Tout d'abord nous allons configurer nos serveurs proxy en IP static pour cela il faut d'abord s'identifier en root et ensuite faire : `nano /etc/network/interfaces` et dedans rajouter les lignes suivantes en remplaçant les IP et la carte réseau afficher par les vôtres

```
# The primary network interface
allow-hotplug ens18
iface ens18 inet dhcp

auto ens18
iface ens18 inet static
    address 172.16.3.70
    netmask 255.255.255.0
```

```
# The primary network interface
allow-hotplug ens18
iface ens18 inet dhcp

auto ens18
iface ens18 inet static
    address 172.16.3.68
    netmask 255.255.255.0
```

4. Ensuite on fait de même avec les serveurs web

```
# The primary network interface
allow-hotplug ens18
iface ens18 inet static
    address 172.16.3.69
    gateway 172.16.0.1

auto ens18
iface ens18 inet static
    address 172.16.3.64/24
```

```
# The primary network interface
allow-hotplug ens18
iface ens18 inet static
    address 172.16.3.71
    gateway 172.16.0.1

auto ens18
iface ens18 inet static
    adress 172.16.3.64/24
```

5. Par la suite, nous allons modifier les 2 fichier de configuration de HaProxy : nano /etc/haproxy/haproxy.cfg

```
backend http_roundrobin
    balance roundrobin
    server proxy1 172.16.3.70:80 check
    server proxy2 172.16.3.68:80 check

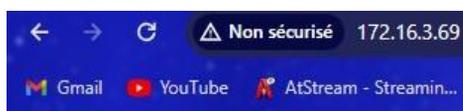
frontend http_haproxy
    mode http
    bind *:80
    default_backend http_roundrobin
```

6. Sur les serveurs web taper cette commande afin de modifier l'interface du site web. Nous pourrons ainsi vérifier le bon fonctionnement du load balancer :

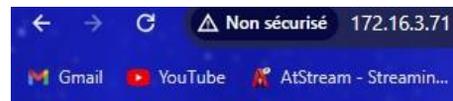
```
root@web1:~# echo serveur 1 > /var/www/html/index.nginx-debian.html
```

```
root@WEB2:~# echo server 2 > /var/www/html/index.nginx-debian.html
```

7. Tapez ensuite systemctl restart nginx pour redémarrer le service et dans un navigateur entrez l'adresse IP de votre serveur proxy, en rafraichissant la page il doit afficher vos 2 serveur web :



server 2



server 1

8. Ensuite nous allons ajouter « listenstats » a notre HaProxy pour voir les statistiques depuis le web. Pour ce faire il faut modifier le fichier de configuration HaProxy :
nano /etc/haproxy/haproxy.cfg puis ajouter ces lignes.

```
listen stats
  bind *:9000
  mode http
  stats enable
  stats hide-version
  stats realm HAproxy-statistics
  stats uri /haproxy_stats
  stats auth user:password
```

Remplacer user:password par votre nom d'utilisateur et votre mot de passe. Une fois fais, redémarrer le service HaProxy (systemctl restart haproxy) puis entrer l'IP de votre proxy sur le port 9000 (ex : 172.16.3.68:9000/haproxy_stats)
 Cela vous redirigera vers une page de connexion, entrez vos identifiant choisis au préalable.

HAProxy
Statistics Report for pid 4931

> General process information

pid = 4931 (process #1, nbproc = 1, nbthread = 1)
 uptime = 0d 0h0m22s
 system limits: rmemmax = unlimited, ulimit.n = 524287
 maxsock = 524287, maxconn = 262123, maxpipes = 0
 current conn = 2, current pipes = 0, conn rate = 0/sec, bit rate = 0 000 kbps
 Running tasks: 0/17, idle = 100 %

active UP, going down | backup UP, going down
 active DOWN, going up | backup DOWN, going up
 active or backup DOWN | not checked
 active or backup DOWN for maintenance (MAINT)
 active or backup SOF1 STOPPED for maintenance
 Note: "NOLEAFDOWN" = UP with load balancing disabled.

Display option:
 • Scope:
 • Hide DOWN servers
 • Refresh now
 • CSV export
 • JSON export (schema)

External resources:
 • Primary site
 • Updates (v2.6)
 • Online manual

Queue		Session rate			Sessions			LibTot		Bytes		Denied		Errors		Warnings		Status		Server									
Cur	Max	Limit	Cur	Max	Limit	Cur	Max	Limit	Total	LibTot	Last	In	Out	Req	Resp	Req	Conn	Resp	Rate	Refds	State	LastChk	Wght	Act	Bck	Chk	Dns	Downtime	Thrtle
Frontend		0	0	-	0	1	262	123	0			0	0	0	0	0						OPEN							

Queue		Session rate			Sessions			LibTot		Bytes		Denied		Errors		Warnings		Status		Server									
Cur	Max	Limit	Cur	Max	Limit	Cur	Max	Limit	Total	LibTot	Last	In	Out	Req	Resp	Req	Conn	Resp	Rate	Refds	State	LastChk	Wght	Act	Bck	Chk	Dns	Downtime	Thrtle
Frontend		0	2	-	2	2	262	123	3			1	963	23	789	0	0	0				OPEN							
Backend		0	0	-	0	0	0	0	26	213	0	0	0s	1	963	23	789	0	0			4m22s UP		00	0	0	0	0	

Queue		Session rate			Sessions			LibTot		Bytes		Denied		Errors		Warnings		Status		Server									
Cur	Max	Limit	Cur	Max	Limit	Cur	Max	Limit	Total	LibTot	Last	In	Out	Req	Resp	Req	Conn	Resp	Rate	Refds	State	LastChk	Wght	Act	Bck	Chk	Dns	Downtime	Thrtle
VM1		0	0	-	0	0	0	0	-	0	0	?	0	0	0	0	0	0	0	0		4m22s UP		1/1	Y	-	0	0	
VM2		0	0	-	0	0	0	0	-	0	0	?	0	0	0	0	0	0	0	0		4m22s UP		1/1	Y	-	0	0	
Backend		0	0	-	0	0	0	0	26	213	0	0	?	0	0	0	0	0	0	0		4m22s UP		2/2	0	0	0	0	

Vous avez maintenant accès aux statistiques de votre HaProxy

Maintenant il faut installer keepalived sur les deux serveurs proxy « apt install keepalived »

Maintenant que cela est fait sur les deux VM nous allons pouvoir commencer il va falloir crée une adresse ip virtuel

qui remontera sur les deux machine car en effet le but ici et de pouvoir aller sur l'un des deux Haproxy quoi qu'il arrive.

Pour cela il faut définir celui qui va être le « Master » et celui qui va être le « slave » Nous allons dans un premier temps configuré le « Master » pour cela il faut crée le fichier `keepalived.conf` pour cela `nano/etc/keepalived/keepalived.conf` et y insérer les lignes suivantes

```
GNU nano 7.2 /etc/keepalived/keepalived.conf
vrrp_script reload_haproxy {
    script "killall -0 haproxy"
    interval 1
}

vrrp_instance VI_1 {
    virtual_router_id 100
    state MASTER
    priority 100
    # Check inter-load balancer toutes les 1 secondes
    advert_int 1
    # Synchro de l'état des connexions entre les LB sur l'interface enp0s3
    lvs_sync_daemon_interface enp0s3
    interface enp0s8
    # Authentification mutuelle entre les LB, identique sur les deux membres
    authentication {
        auth_type PASS
        auth_pass secret
    }
    # Interface réseau commune aux deux LB
    virtual_ipaddress {
        192.168.50.199/32 brd 192.168.50.255 scope global
    }

    track_script {
        reload_haproxy
    }
}
```

Ensuite il faut redémarrer le service `keepalived` « `service keepalived restart` » puis faire un « `ip a` » pour voir si l'ip virtuel remonte bien

```
: enp0s8: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default qlen 1
    link/ether 08:00:27:6b:5d:fd brd ff:ff:ff:ff:ff:ff
    inet 192.168.50.5/24 brd 192.168.50.255 scope global enp0s8
        valid_lft forever preferred_lft forever
    inet 192.168.50.199/32 brd 192.168.50.255 scope global enp0s8
        valid_lft forever preferred_lft forever
    inet6 fe80::a00:27ff:fe6b:5dfd/64 scope link
        valid_lft forever preferred_lft forever
```

Nous pouvons voir que l'ip virtuel sue nous avons mis le 192.168.30.199 remonte bien. Il faut faire la configuration aussi sur le « slave » Pareil que l'autre il faut créer le fichier keepalived.conf Avec la commande nano /etc/keepalived/keepalived.conf Et y insérer les lignes suivante

```
vrrp_script reload_haproxy {
    script "killall -0 haproxy"
    interval 1
}

vrrp_instance VI_1 {
    virtual_router_id 100
    state BACKUP
    priority 100
    # Check inter-load balancer toutes les 1 secondes
    advert_int 1
    # Synchro de l'état des connexions entre les LB sur l'interface enp0s3
    lvs_sync_daemon_interface enp0s3
    interface enp0s3
    # Authentification mutuelle entre les LB, identique sur les deux membres
    authentication {
        auth_type PASS
        auth_pass secret
    }
    # Interface réseau commune aux deux LB
    virtual_ipaddress {
        192.168.50.199/32 brd 192.168.50.255 scope global
    }

    track_script {
        reload_haproxy
    }
}
```

Il faut ensuite encore redémarrer le service et taper « ip a »

```
root@serveurproxy:~# ip a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
        valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host noprefixroute
        valid_lft forever preferred_lft forever
2: enp0s3: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default qlen 1000
    link/ether 08:00:27:9b:23:77 brd ff:ff:ff:ff:ff:ff
    inet 10.0.2.15/24 brd 10.0.2.255 scope global dynamic enp0s3
        valid_lft 86187sec preferred_lft 86187sec
    inet6 fe80::a00:27ff:fe9b:2377/64 scope link
        valid_lft forever preferred_lft forever
3: enp0s8: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default qlen 1000
    link/ether 08:00:27:6a:9d:31 brd ff:ff:ff:ff:ff:ff
    inet 192.168.50.6/24 brd 192.168.50.255 scope global enp0s8
        valid_lft forever preferred_lft forever
    inet6 fe80::a00:27ff:fe6a:9d31/64 scope link
        valid_lft forever preferred_lft forever
root@serveurproxy:~#
```

L'adresse Ip virtuel remonte bien nous allons donc pouvoir aller sur les deux Haproxy avec la même Ip.

On va ensuite installer un certificat SSL

Pour commencer on va installer OpenSSL « apt install openssl »

Ensuite générer une clé

```
openssl req -x509 -days 365 -out mycert.crt -nodes -newkey rsa:4096 -keyout mykey.key
```

```
Generating a RSA private key
.....+++++
.....+++++
writing new private key to 'mykey.key'
-----
You are about to be asked to enter information that will be incorporated
into your certificate request.
What you are about to enter is what is called a Distinguished Name or a DN.
There are quite a few fields but you can leave some blank
For some fields there will be a default value,
If you enter '.', the field will be left blank.
-----
Country Name (2 letter code) [AU]:FR
State or Province Name (full name) [Some-State]:RHONE
Locality Name (eg, city) []:maville
Organization Name (eg, company) [Internet Widgits Pty Ltd]:CONTACTIT
Organizational Unit Name (eg, section) []:IT
Common Name (e.g. server FQDN or YOUR name) []:web.it.fr
Email Address []:contactit.yarka@slmail.me
```

On va ensuite fusionner notre clé et certificat en un .pem

« mkdir /etc/ssl/nom_de_fichier »

« cat nom.key nom.crt > /etc/ssl/nom_du_fichier/nom_clé.pem »

Puis changer le fichier de config Haproxy

```
frontend http_haproxy
  mode http
  bind *:80
  bind *:443 ssl crt /etc/ssl/certs/ssl.pem
  http-request redirect scheme https unless { ssl_fc }
  default_backend http_roundrobin
```

Redémarrer ensuite le service Haproxy et aller sur l'interface web du proxy et vous serez en mesure voir https