HAProxy : High Availability + Load Balancing

<http://www.haproxy.org/>

HAProxy est une solution gratuite, très rapide et fiable offrant la haute disponibilité, équilibrage de charge et proxy pour les applications TCP et HTTP. Il convient particulièrement aux sites Web à très fort trafic et alimente bon nombre des sites les plus visités au monde. Au fil des années, il est devenu l'équilibreur de charge open-source standard. Il est désormais livré avec la plupart des distributions Linux classiques et est souvent déployé par défaut sur des plates-formes cloud.

Contexte :

Attention : il y'a deux segments LAN sur VMWare "LAN" et "WAN"

Une VM HAProxy de répartition de charge :

```
hostname : HAProxy (apt install haproxy openssh-server openssh-client)
network adapter : carte 1 : en segment LAN "LAN": static 172.20.0.20/24
network adapter 1 : carte 2 : en segment LAN "WAN": static 192.168.0.20/24
.
```

Deux serveurs Web (apt install apache2 php openssh-server openssh-client) :

srvWeb1 :

```
· hostname : SRV-WEB1
· network adapter : carte 1 : 172.20.0.21/24 : en segment LAN "LAN": static
```

srvWeb2 :

```
· hostname : SRV-WEB2
· network adapter : carte 1 : 172.20.0.22/24 : en segment LAN "LAN": static
```

Un machine client (Windows) :

- Carte réseau1 : en segment LAN "WAN" : static : 192.168.0.100/24

Validation 1 :

Au niveau de toutes les machine, vérifiez la configuration IP :
"ip a" et "ipconfig"

Depuis la machine HAProxy :

```
· ping 172.20.0.21 -> ok
· ping 172.20.0.22 -> ok
· service ssh status -> Active running
.
```

Depuis les deux serveurs Web :

```
service apache2 status -> Active runing
```

```
service ssh status -> Active running
```

Depuis la machine client :

```
ping 192.168.0.20 -> OK
```

Personnalisez le page d'accueil d'apache2 :

La page d'accueil d'apache2 correspond au fichier `/var/www/html/index.html` :

Vérifiez à l'aide :

```
ls /var/www/html
```

Supprimer le fichier index.html des deux serveurs :

```
rm /var/www/html/index.html
```

Pour personnaliser les pages d'apache, créez un fichier index.php

Au niveau des deux serveurs Web :

```
nano /var/www/html/index.php
```

```
<?php
echo "<center><h1> Un coucou de ".gethostname()
?>
```

05/10/2020

Configuration de HAPROXY :

Nous allons configurer le service haproxy pour assurer le balancement (load balancing) entre les serveurs web du cluster.

Le fichier de configuration de haproxy est `/etc/haproxy/haproxy.cfg`, éditez ce dernier (nano) pour ajouter à la fin du fichier les lignes suivantes :

```
nano /etc/haproxy/haproxy.cfg
```

Ajouter les lignes suivantes à la fin du fichier :

```
# Configuration du balancement
listen clusterWeb
bind 192.168.0.20:80

# mode d'écoute
mode http

# mode du balancement (roundrobin (50%-50%))
balance roundrobin

# Option
```

```
option httpclose
option forwardfor
```

```
# Liste des serveurs impliqués pas le balancement
server SRV-WEB1 172.20.0.21:80 check
server SRV-WEB2 172.20.0.22:80 check
```

```
# Pour les statistiques
stats enable
stats hide-version
stats refresh 30s
stats show-node
stats auth admin:password
stats uri /statistique
```

Enregistrez votre fichier.

Validation :

```
service haproxy restart
service haproxy status -> Active running
```

Pour vérifier le balancement : RDV machine client-> IE -> 192.168.0.20, vous devez lire Un coucou de srvWeb 1, actualisez, vous devez lire Un coucou de srvWeb2, le balancement est bien fonctionnel.

Pour vérifier les stats : RDV machine client -> IE -> 192.168.0.20/statistique, le login c'est admin:password, vous devez voir une page de statistique avec les deux serveurs Web srvWeb1 et srvWeb2 up en vert.

c

Pour tester, vous pouvez "service apache2 stop" sur l'un des serveurs, attendre 30 secondes et vérifiez que le serveur en question passe en état down en rouge dans la page des statistiques.

Explications :

listen cluster_web adresseIPHaProxy:80: Cette directive nous permet de spécifier sur quelle adresse IP HaProxy va fonctionner, nous allons accéder au contenu web depuis cette adresse IP.

mode http: permet de spécifier que le balancement de charge est utilisé pour du contenu web http, dans le cas contraire on peut utiliser le mode tcp (pour du mysql par exemple ;)

balance roundrobin: permet de spécifier l'algorithme de répartition de charge. Il en existe plusieurs

RoundRobin: La méthode Round-robin est une répartition équitable de la charge entre les serveurs d'un cluster. Chaque serveur traite le même nombre de requêtes, mais cela nécessite d'avoir des serveurs homogènes en termes de capacité de traitement.

Source : Le mode de balancement « source » signifie qu'un client en fonction de son adresse IP sera toujours dirigé vers le même serveur web. Cette option est nécessaire lorsque les sites Internet utilisent des sessions PHP.

Least connection: Le serveur renvoie vers le serveur le moins chargé. Si en théorie il semble le plus adapté, en réalité dans le cadre du Web dynamique, un serveur peut être considéré comme chargé alors que les processus sont en attente d'une requête vers une base de données.

First Response: Les requêtes clients sont envoyées simultanément à tous les serveurs et le premier qui répond sera chargé de la connexion. Difficile à mettre en oeuvre et rarement employé.