

Activité 3 : HAProxy : High Availability + Load Balancing

1- Introduction :

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 opensource standard. Il est désormais livré avec la plupart des distributions Linux classiques et est souvent déployé par défaut sur des plates-formes cloud.

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

2- Besoins pour la réalisation :

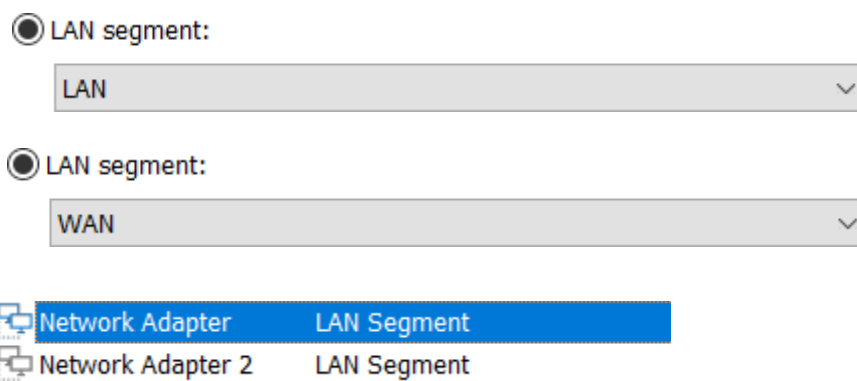
VMWare Workstation 16 avec 3 machines Debian 11 dont une VM HAProxy de répartition de charges et deux serveurs Web et une machine cliente Windows 10 pour valider la réalisation.

3- Réalisation et prérequis

1- Paramétrer l'ensemble des machines :

○ Sur le serveur HAProxy :

Depuis les settings de la machine HAProxy, mettre son interface réseau en segment LAN que l'on appellera « LAN » et ajouter une seconde interface réseau que l'on nommera « WAN » :



Nous allons maintenant attribuer des adresses IP statiques à ces interfaces réseaux et installer notre service HAProxy.

Vérifier le nom des interfaces avec la commande « ip a »

```
2: ens33: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state UP group default qlen 1000
3: ens37: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state UP group default qlen 1000
```

Ici nous avons les interfaces « ens33 » et « ens37 ».

Configurer les interfaces avec la commande :

`nano /etc/network/interfaces`

Attribuer une adresse statique sur la première interface réseau « ens33 » :

`allow-hotplug ens33`

`iface ens33 inet static`

`address 172.20.0.20/24`

Attribuer une adresse statique sur la seconde interface réseau « ens37 » :

`allow-hotplug ens37`

`iface ens37 inet static`

`address 192.168.0.20/24`

```
GNU nano 5.4 /etc/network/interfaces
# This file describes the network interfaces available on your system
# and how to activate them. For more information, see interfaces(5).

source /etc/network/interfaces.d/*

# The loopback network interface
auto lo
iface lo inet loopback

# The primary network interface
allow-hotplug ens33
iface ens33 inet static
address 172.20.0.20/24

allow-hotplug ens37
iface ens37 inet static
address 192.168.0.20/24
```

Changer le nom de la machine en « HAProxy » :

`nano /etc/hostname`

```
GNU nano 5.4 /etc/hostname
HAProxy
```

Installer le service HAProxy avec la commande :

`apt install haproxy openssh-server openssh-client`

Redémarrer la machine avec « reboot ».

- **Pour les serveurs web Debian 11 :**

nano /etc/hostname

Changer le nom des machines en SRV-WEB1 et SRV-WEB2

Depuis les settings des machines serveurs, mettre leur interface réseau en segment LAN « LAN ».

Changer l'ip des interfaces réseaux des deux machines :

nano /etc/network/interfaces

Pour le serveur web 1 : **172.20.0.21/24**

```
GNU nano 5.4 /etc/network/interfaces
# This file describes the network interfaces available on your system
# and how to activate them. For more information, see interfaces(5).

source /etc/network/interfaces.d/*

# The loopback network interface
auto lo
iface lo inet loopback

# The primary network interface
allow-hotplug ens33
iface ens33 inet static
address 172.20.0.21/24
```

Pour le serveur web 2 : **172.20.0.22/24**

```
GNU nano 5.4 /etc/network/interfaces
# This file describes the network interfaces available on your system
# and how to activate them. For more information, see interfaces(5).

source /etc/network/interfaces.d/*

# The loopback network interface
auto lo
iface lo inet loopback

# The primary network interface
allow-hotplug ens33
iface ens33 inet static
address 172.20.0.22/24
```

Installer apache2 sur les deux serveurs :

apt install apache2 php openssh-server openssh-client

- **Pour la machine Windows 10 :**

Dans les paramètres de la machine, mettre l'interface réseau en segment « WAN »

Aller dans le panneau de configuration :

Ajuster les paramètres de l'ordinateur



Système et sécurité

Consulter l'état de votre ordinateur
Enregistrer des copies de sauvegarde de vos fichiers à l'aide de l'Histoire des fichiers
Sauvegarder et restaurer (Windows 7)



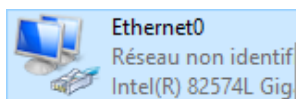
Réseau et Internet

Connexion à Internet
Afficher l'état et la gestion du réseau

Page d'accueil du panneau de configuration

Modifier les paramètres de la carte

Modifier les paramètres de partage avancés



Ethernet0

Réseau non identifié
Intel(R) 82574L Gig

- Désactiver
- Statut**
- Diagnostiquer
- Connexions de pont
- Créer un raccourci
- Supprimer
- Renommer
- Propriétés**

Propriétés de Ethernet0

Gestion de réseau

Connexion en utilisant :

Intel(R) 82574L Gigabit Network Connection

Configurer...

Cette connexion utilise les éléments suivants :

- ☒ Client pour les réseaux Microsoft
- ☒ Partage de fichiers et imprimantes Réseaux Microsoft
- ☒ Planificateur de paquets QoS
- ☒ **Protocole Internet version 4 (TCP/IPv4)**
- ☐ Protocole de multiplexage de carte réseau Microsoft
- ☒ Pilote de protocole LLDP Microsoft
- ☒ Protocole Internet version 6 (TCP/IPv6)

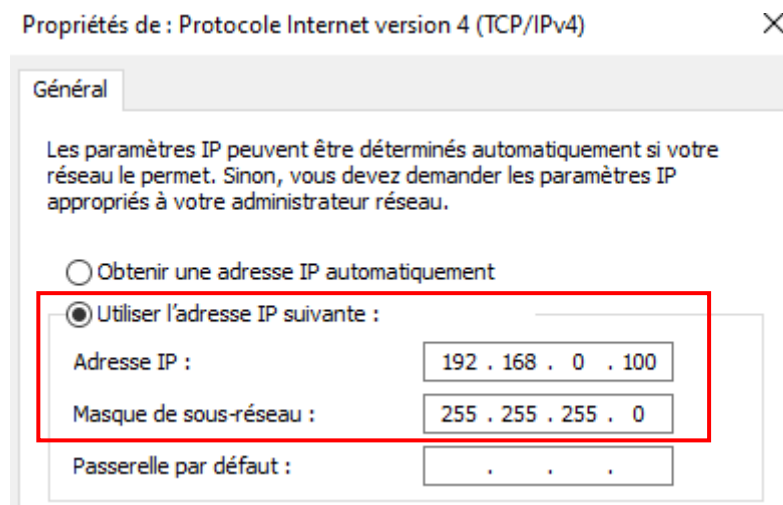
Installer... Désinstaller Propriétés

Description

Protocole TCP/IP (Transmission Control Protocol/Internet Protocol). Protocole de réseau étendu par défaut permettant la communication entre différents réseaux interconnectés.

OK Annuler

Mettre l'adresse IP de **192.168.0.100/24** :



Vérifier le paramétrage des machines :

Au niveau de la configuration IP : « ip a » pour Debian et « ipconfig » pour Windows

La machine HAProxy devrait pouvoir ping les deux serveurs web :

ping 172.20.0.21

ping 172.20.0.22

« service ssh status » devrait afficher Active Running

```
root@HAProxy:~# ping 172.20.0.21
PING 172.20.0.21 (172.20.0.21) 56(84) bytes of data.
64 bytes from 172.20.0.21: icmp_seq=1 ttl=64 time=0.252 ms
64 bytes from 172.20.0.21: icmp_seq=2 ttl=64 time=0.284 ms
64 bytes from 172.20.0.21: icmp_seq=3 ttl=64 time=0.583 ms

root@HAProxy:~# ping 172.20.0.22
PING 172.20.0.22 (172.20.0.22) 56(84) bytes of data.
64 bytes from 172.20.0.22: icmp_seq=1 ttl=64 time=0.299 ms
64 bytes from 172.20.0.22: icmp_seq=2 ttl=64 time=0.312 ms
64 bytes from 172.20.0.22: icmp_seq=3 ttl=64 time=0.455 ms

root@HAProxy:~# service ssh status
• ssh.service - OpenBSD Secure Shell
  Loaded: loaded (/lib/systemd/s
  Active: active (running) since
```

Depuis les deux serveurs Web :

« service apache2 status » devrait afficher Active Running

« service ssh status » devrait afficher Active Running

```
root@SRV-WEB1:~# service apache2 status
• apache2.service - The Apache HTTP Serv
  Loaded: loaded (/lib/systemd/system
  Active: active (running) since Sun
```

```
root@SRV-WEB1:~# service ssh status
• ssh.service - OpenBSD Secure Shell
  Loaded: loaded (/lib/systemd/sy
  Active: active (running) since
```

Depuis le serveur Windows 10 :

ping 192.168.0.20

```
C:\Users\user>ping 192.168.0.20

Envoi d'une requête 'Ping' 192.168.0.20 avec 32 octets de données :
Réponse de 192.168.0.20 : octets=32 temps=15 ms TTL=64
Réponse de 192.168.0.20 : octets=32 temps<1ms TTL=64
Réponse de 192.168.0.20 : octets=32 temps=1 ms TTL=64
```

Nous avons bien paramétré les machines.

4- Service HAProxy

- 1- Vérifier la présence de la page d'accueil d'apache2 dans /var/www/html/index.html
ls /var/www/html/index.html (sur les deux serveurs Web)
- 2- Supprimer le fichier sur les deux serveurs Web :
rm /var/www/html/index.html

Nous allons personnaliser les pages d'apache :
Au niveau des deux serveurs Web :

nano /var/www/html/index.php

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

```
GNU nano 5.4 /var/www/html/index.php
<?php
echo "<center><h1>Un coucou de ".gethostname()
?>
```

Enregistrer le fichier.

- 3- Configurer HAProxy pour assurer le balancement (load balancing) entre les serveurs web du cluster.

Le fichier de configuration de HAProxy est /etc/haproxy/haproxy.cfg

nano /etc/haproxy/haproxy.cfg

Ajouter les lignes suivantes :

```
# 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
```

```
GNU nano 5.4 /etc/haproxy/haproxy.cfg
option httplog
option dontlognull
timeout connect 5000
timeout client 50000
timeout server 50000
errorfile 400 /etc/haproxy/errors/400.http
errorfile 403 /etc/haproxy/errors/403.http
errorfile 408 /etc/haproxy/errors/408.http
errorfile 500 /etc/haproxy/errors/500.http
errorfile 502 /etc/haproxy/errors/502.http
errorfile 503 /etc/haproxy/errors/503.http
errorfile 504 /etc/haproxy/errors/504.http

listen clusterWebStadiumCompany
bind 192.168.0.20:80

mode http

balance roundrobin

option httpclose
option forwardfor

server SRV-WEB1 172.20.0.21:80 check
server SRV-WEB2 172.20.0.22:80 check

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

Les paramètres* sont expliqués à la fin de la procédure.

4- Vérifier que le service HAProxy est bien mis en place :

service haproxy restart

service haproxy status" devrait afficher Active Running

```
root@HAProxy:~# service haproxy status
• haproxy.service - HAProxy Load Balance
   Loaded: loaded (/lib/systemd/systemd; vendor preset: enabled)
   Active: active (running) since Sun 2023-10-01 10:00:00 CEST; 1min 1s ago
```

Sur la machine cliente Windows 10 :

Ouvrir un navigateur et taper 192.168.0.20

On devrait lire « Un coucou de srvWeb1 »

Actualiser la page et on devrait lire « Un coucou de srvWeb2 », le balancement est bien fonctionnel.

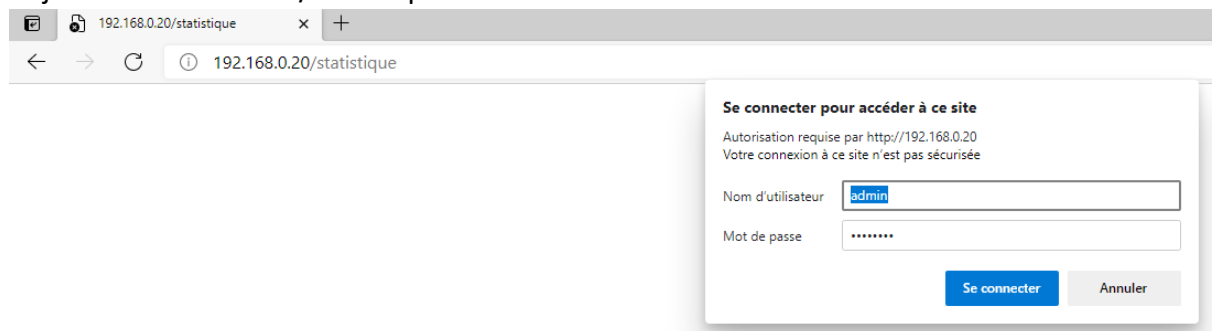


On actualise :



Nous allons maintenant vérifier les statistiques :

Rajouter 192.168.0.20/statistiques



Se connecter avec « admin » en nom d'utilisateur et « password » en mot de passe.

Notre service HAProxy est donc bien mis en place.

**Explication des paramètres :*

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.