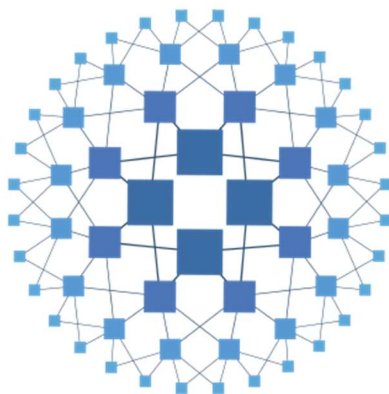


Documentation HAProxy



HAPROXY

Tables des matières:

1. Introduction	2
2. Besoins pour la réalisation	2
3. Réalisation	2
1- Une VM HAProxy de répartition de charge	2
2- Deux serveurs Web (srvWeb1 et srvWeb2) :	2
3- Changer le document Root d'apache2 :	3
4- Vérifier les adresses ip des serveurs et du client :	3
5- Dans le fichier /var/www/html/thegrill-master/index.html,	4
6- Validation n°1 :	5
7- Configuration de HAPROXY :	5
8- Validation n°2 :	7
9- Pour vérifier le balancement :	7
10- Pour vérifier les stats :	8
11- Pour tester, vous pouvez :	8
4. Explications	9

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 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.

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

2. Besoins pour la réalisation :

Poste de travail VMWare Workstation, iso Debian 12, iso Windows 10.

3. Réalisation :

1- Une VM HAProxy de répartition de charge

Créer une nouvelle machine Debian HAPROXY

Hostname : HAProxy

Carte 1 : en segment LAN "**LAN**": **static 172.20.0.20/24**

Carte 2 : en segment LAN "**WAN**": **static 192.168.0.20/24**

Login : root

Mdp : root

Sur cette machine installer HAPROXY, avec la commande suivante :

apt install haproxy

2- Deux serveurs Web (srvWeb1 et srvWeb2) :

Network adapter : carte 1 : en segment **NAT**, **pour avoir accès à internet et installer les paquets.**

Sur les deux VM Debian, installer :

apt install apache2 php wget unzip

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

```
cd /var/www/html
```

```
wget https://github.com/technext/thegrill/archive/master.zip
```

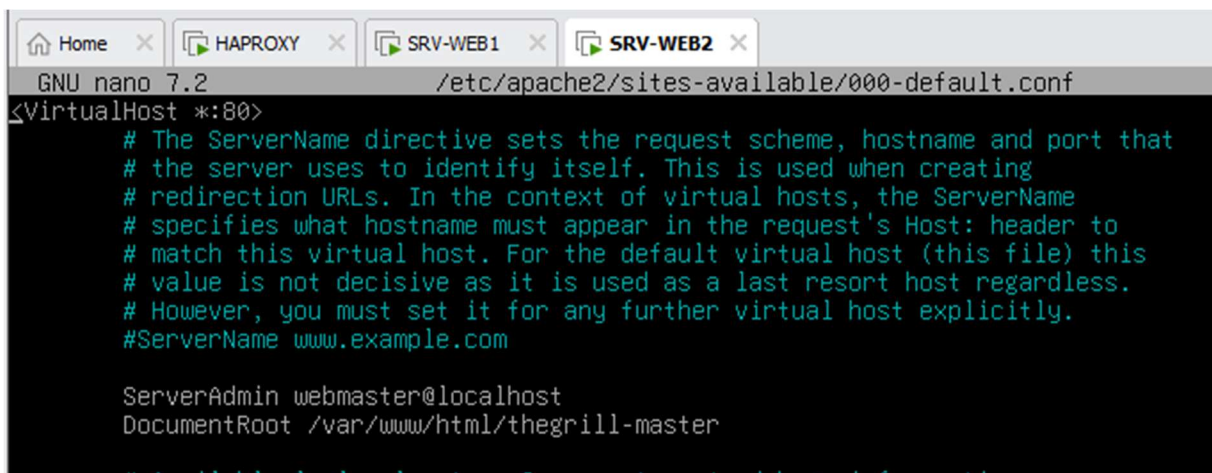
```
unzip master.zip
```

3- Changer le document Root d'apache2 :

```
nano /etc/apache2/sites-available/000-default.conf
```

Ajouter le nom du dossier "**thegrill-master**" derrière **/var/www/html/**

Comme ci-dessous :



```
GNU nano 7.2 /etc/apache2/sites-available/000-default.conf
<VirtualHost *:80>
    # The ServerName directive sets the request scheme, hostname and port that
    # the server uses to identify itself. This is used when creating
    # redirection URLs. In the context of virtual hosts, the ServerName
    # specifies what hostname must appear in the request's Host: header to
    # match this virtual host. For the default virtual host (this file) this
    # value is not decisive as it is used as a last resort host regardless.
    # However, you must set it for any further virtual host explicitly.
    #ServerName www.example.com

    ServerAdmin webmaster@localhost
    DocumentRoot /var/www/html/thegrill-master
```

Enregistrer puis redémarrez Apache2

```
service apache2 restart
```

4- Vérifier les adresses ip des serveurs et du client :

- SRV-WEB1 :
 - Nom d'hôte : **SRV-WEB1**
 - Adaptateur réseau carte 1 : En segment LAN "LAN"
 - Statique : **172.20.0.21/24**

Comme ci-dessous :

```
root@SRV-WEB1:~# service apache2 restart
root@SRV-WEB1:~# 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: ens33: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default qlen
    link/ether 00:0c:29:88:7f:90 brd ff:ff:ff:ff:ff:ff
    altname enp2s1
    inet 172.20.0.21/24 brd 172.20.0.255 scope global ens33
        valid_lft forever preferred_lft forever
    inet6 fe80::20c:29ff:fe88:7f90/64 scope link
        valid_lft forever preferred_lft forever
root@SRV-WEB1:~#
```

- SRV-WEB2 :
 - Nom d'hôte : **SRV-WEB2**
 - Adaptateur réseau carte 1 : En segment LAN "**LAN**"
 - Statique : **172.20.0.22/24**

Comme ci-dessous :

```
root@SRV-WEB2:~# service apache2 restart
root@SRV-WEB2:~# 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: ens33: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default qlen
    link/ether 00:0c:29:e7:1b:b3 brd ff:ff:ff:ff:ff:ff
    altname enp2s1
    inet 172.20.0.22/24 brd 172.20.0.255 scope global ens33
        valid_lft forever preferred_lft forever
    inet6 fe80::20c:29ff:fee7:1bb3/64 scope link
        valid_lft forever preferred_lft forever
```

- Un client machine (Windows) :
 - Carte réseau 1, en segment LAN "**WAN**"
 - Statique : **192.168.0.100/24**

5- Dans le fichier /var/www/html/thegrill-master/index.html,

Ajouter respectivement **1** et **2** après le titre **THE GRILL** de SRV-WEB1 et SRV-WEB2, comme ci-dessous :

- SRV-WEB1 :

```
<title>The Grill1 by Distinctive Themes</title>
<!-- Bootstrap Core CSS -->
```

- SRV-WEB2 :

```
<title>The Grill2 by Distinctive Themes</title>
```

6- Validation n°1 :

Au niveau de toutes les machines, vérifiez la configuration IP grâce à la commande "**ip a**".

On ping :

- Depuis la machine HAProxy :
 - ping 172.20.0.21 -> ok
 - ping 172.20.0.22 -> ok
- Depuis les deux serveurs Web :
 - service apache2 status -> En cours d'exécution
- Depuis la machine cliente :
 - ping 192.168.0.20 -> OK

7- 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 :

listen clusterWebM2L

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

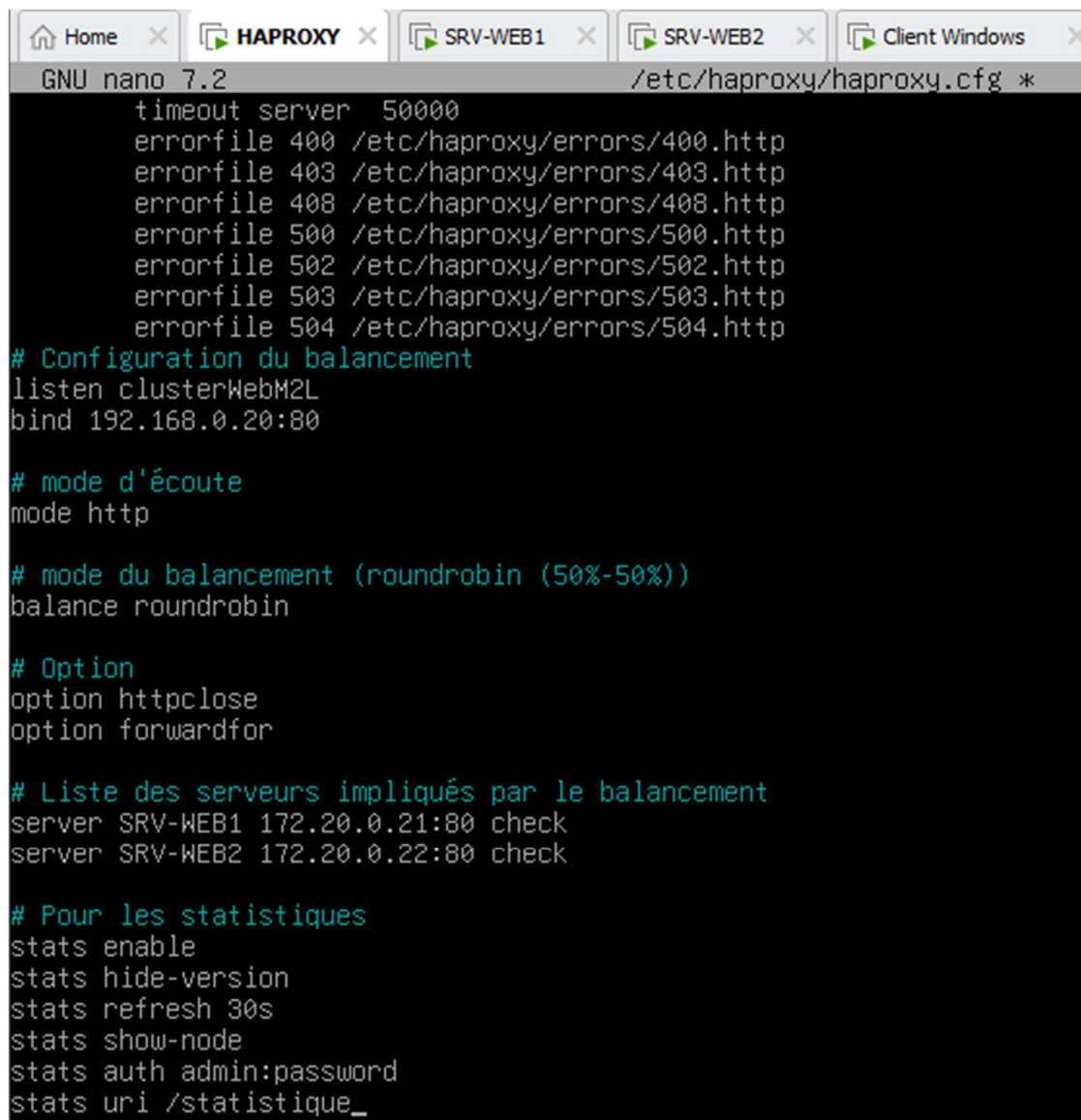
stats refresh 30s

stats show-node

stats auth admin:password

stats uri /statistique

Comme ci-dessous :

A screenshot of a terminal window with multiple tabs at the top: 'Home', 'HAPROXY', 'SRV-WEB1', 'SRV-WEB2', and 'Client Windows'. The active tab is 'HAPROXY', and the file being edited is '/etc/haproxy/haproxy.cfg'. The terminal shows the content of this file, which is a HAProxy configuration. It includes settings for timeouts, error files, listen address (192.168.0.20:80), mode (http), balance method (roundrobin), options (httpclose, forwardfor), server definitions (SRV-WEB1 and SRV-WEB2), and statistics configuration. The text is displayed in a monospaced font with some lines highlighted in green.

```
GNU nano 7.2 /etc/haproxy/haproxy.cfg *
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
# Configuration du balancement
listen clusterWebM2L
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 par 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 ce fichier.

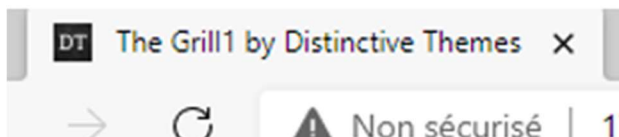
8- Validation n°2 :

service haproxy restart

service haproxy status -> Active running, tout est ok

9- Pour vérifier le balancement :

- Aller sur machine client
- Sur la barre de recherche, entrer **192.168.0.20**,
- Vous devez voir The Grill 1,



- Actualiser. Vous devez voir The Grill 2,



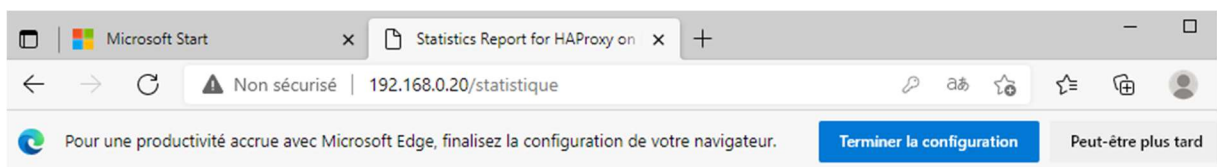
- Le balancement est bien fonctionnel.

10- Pour vérifier les stats :

- Aller sur machine client
- Sur la barre de recherche : **192.168.0.20/statistique**,
login : admin
mdp : password

Vous devez voir une page de statistique avec les deux serveurs Web up en vert.

Comme ci-dessous :



HAProxy

Statistics Report for pid 1153 on HAPROXY

> General process information

pid = 1153 (process #1, nbproc = 1, nbthread = 1)
uptime = 0d 0h06m00s
system limits: memmax = unlimited; ulimit-n = 524287
maxsock = 524287; maxconn = 262124; maxpipes = 0
current conns = 2; current pipes = 0/0; conn rate = 1/sec; bit rate = 0.000 kbps
Running tasks: 0/15; idle = 100 %

active UP
active UP, going down
active DOWN, going up
active or backup DOWN
active or backup DOWN for maintenance (MAINT)
active or backup SOFT STOPPED for maintenance
Note: "NOLB"/"DRAIN" = UP with load-balancing disabled.

Display option:

- Scope :
- Hide 'DOWN' servers
- Disable refresh
- Refresh now
- CSV export
- JSON export (schema)

External resources:

- Primary site
- Updates (v2.6)
- Online manual

clusterWebM2L		Queue		Session rate		Sessions		Bytes		Denied		Errors		Warnings		Status		Server	
		Cur	Max	Limit	Cur	Max	Limit	Total	LbTot	Last	In	Out	Req	Resp	Req	Conn	Resp	Retr	Redis
Frontend		1	25	-	2	6	262 124	39			13 968	1 758 319	0	0	1			OPEN	
SRV-WEB1	0	0	-	0	13	0	3	-	17	17	2m16s	6 529	785 812	0	0	0	0	0	6m9s UP
SRV-WEB2	0	0	-	0	12	0	3	-	17	17	2m16s	6 484	971 599	0	0	0	0	0	6m9s UP
Backend	0	0		0	25		6	26 213	34	34	0s	13 968	1 758 319	0	0	0	0	0	6m9s UP
																		2/2	2 0

11- Pour tester, vous pouvez :

- Entrer la commande **service apache2 stop** sur l'un des serveurs,
- Attendre 30 secondes
- Vérifier que le serveur en question passe en état down, en rouge, dans la page des statistiques.

4. Explications :

- listen cluster_web addressIPHAproxy: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 œuvre et rarement employé.