

Qu'est-ce que HaProxy?

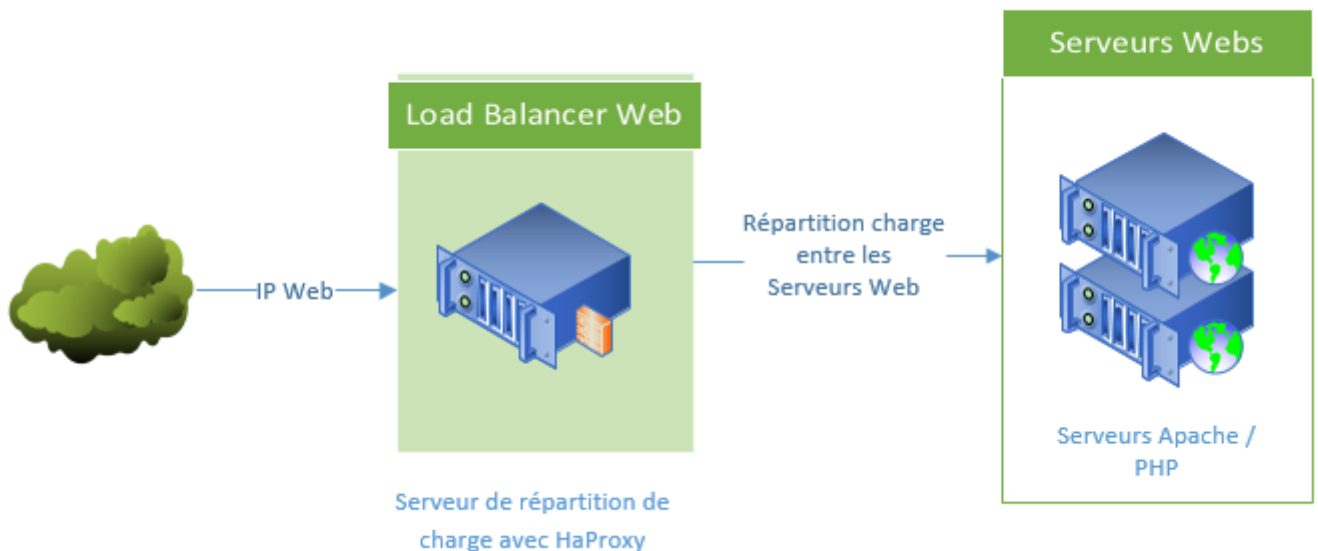
Développée par le français **Willy Tarreau** en 2002, **HAProxy** est une solution libre, fiable et très performante de répartition de charge de niveau 4 (TCP) et 7 (HTTP).

Elle est particulièrement adaptée aux sites web fortement chargés qui nécessitent de la disponibilité.

Dans ce tutoriel, nous allons mettre en place trois serveurs

- Un serveur de répartition de charge avec **HaProxy**
- Deux serveurs web avec Apache et PHP

Voici la topologie que nous allons mettre en place:



Topologie

Dans **Virtualbox**, le réseau en **192.168.2.XXX** est mon réseau local (mode pont) et pour le réseau en **10.1.0.XXX** la configuration de la carte est en mode **Réseau interne**.

Les serveurs sont configurés de la façon suivante:

Serveur de répartition de charge

hostname: lb

Deux carte réseaux

eth0: 192.168.2.100

eth1: 10.1.0.1

Serveurs Web

Serveur web1

hostname: web1

eth0: 10.1.0.21

Serveur web2
hostname: web2
eth0: 10.1.0.22

Installation du Load Balancer.

Une fois les cartes réseau de notre serveur de répartition de charge configurés correctement, nous allons pouvoir installer et configurer HaProxy.

Config IP du serveur

```
# The primary network interface
auto eth0
allow-hotplug eth0
iface eth0 inet static
address 192.168.2.100
netmask 255.255.255.0
gateway 192.168.2.1

auto eth1
iface eth1 inet static
address 10.1.0.1
netmask 255.255.255.0
```

Malheureusement, HaProxy n'est pas dans les dépôts principaux de Debian, nous allons devoir utiliser les dépôts Backports pour l'installer.

```
echo "deb http://ftp.debian.org/debian/ wheezy-backports main" >>
/etc/apt/sources.list
apt-get update
apt-get install haproxy
```

Une fois l'installation terminée, nous devons éditer la valeur **ENABLED=0** à **ENABLED=1** dans les fichiers **/etc/default/haproxy** et **/etc/init.d/haproxy**.

Vous pouvez le faire avec les commandes suivantes:

```
sed -i "s/ENABLED=0/ENABLED=1/g" /etc/default/haproxy
sed -i "s/ENABLED=0/ENABLED=1/g" /etc/init.d/haproxy
```

Configuration de HaProxy

La configuration de **HaProxy** est assez simple et se fait dans un seul fichier de configuration, ce fichier se trouve dans **/etc/haproxy/haproxy.cfg**.

```
global
    log /dev/log local0
    log /dev/log local1 notice
    chroot /var/lib/haproxy
    user haproxy
    group haproxy
    daemon
```

```

defaults
    log      global
    mode     http
    option   httplog
    option   dontlognull
    timeout  5000
    clitimeout 50000
    srvtimer 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 cluster_web 192.168.2.100:80
    #Web
    mode http

    #Mode de balancement Round Robin
    balance roundrobin

    #Options
    option httpclose
    option forwardfor

    #Les serveurs Web
    server web1 10.1.0.21:80 check
    server web2 10.1.0.22:80 check

    #Stats
    stats enable
    stats hide-version
    stats refresh 30s
    stats show-node
    stats auth admin:admin
    stats uri /stats

```

Je vais analyser ce fichier de configuration:

listen cluster_web 192.168.2.100:80: 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.

server web1 10.1.0.21:80 check: permet déclarer les différents serveurs web qui vont être utilisés pour la répartition de charge

Enfin **stats**, permet de configurer la page de statistiques de **HaProxy**, dans cette configuration, la page de statistiques sera disponible via l'adresse suivante **http://192.168.2.100/stats** avec les identifiants **admin/admin**.

Maintenant vous pouvez enregistrer le fichier et démarrez **HaProxy** avec la commande suivante:

```
/etc/init.d/haproxy start
```

Vous pouvez maintenant accéder à l'interface de statistiques de HaProxy via l'adresse: **http://192.168.2.100/stats**

Vous pouvez voir sur l'image suivante que les deux serveurs Web sont affichés comme **DOWN**, c'est normal parce que nous ne les avons pas encore configurés.

HaProxy

Statistics Report for pid 2902 on lb

> General process information

pid = 2902 (process #1, nbproc = 1)
 uptime = 0d 0h12m02s
 system limits: memmax = unlimited; ulimit-n = 4013
 maxsock = 4013; maxconn = 2000; maxpipes = 0
 current conns = 1; current pipes = 0/0
 Running tasks: 1/3

cluster_web		Queue			Session rate			Sessions				Bytes		Denied		Errors			Warnings		Status	
		Cur	Max	Limit	Cur	Max	Limit	Cur	Max	Limit	Total	LbTot	In	Out	Req	Resp	Req	Conn	Resp	Retr	Redis	
Frontend					0	2	-	1	2	2 000	2		353	262	0	0	0					OPEN
web1		0	0	-	0	0		0	0	-	0	0	0	0		0		0	0	0	0	12m2s DOWN
web2		0	0	-	0	0		0	0	-	0	0	0	0		0		0	0	0	0	11m59s DOWN
Backend		0	0		0	0		0	0	2 000	0	0	353	262	0	0		0	0	0	0	11m59s DOWN

Ha Stats DOWN

Avant de mettre en place les serveurs Web, nous allons mettre en place l'**IP forwarding** sur le serveur **LB** pour que les serveurs Web aient accès à internet et aussi quelques règles iptables pour mettre en place du NAT afin de pouvoir accéder aux serveurs Web via **SSH**.

Configuration des serveurs web.

Maintenant que le serveur est configuré, nous allons pouvoir commencer à configurer les serveurs Web.

En ce qui concerne les adresses des deux serveurs Web, n'oubliez pas d'utiliser **comme passerelle l'IP du serveur haproxy**.

Voici la configuration IP de web1:

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

# The loopback network interface
auto lo
iface lo inet loopback

# The primary network interface
auto eth0
allow-hotplug eth0
iface eth0 inet static
address 10.1.0.21
netmask 255.255.255.0
gateway 10.1.0.1
```

Si **web1** et **web2** sont bien configurée, vous pouvez maintenant accéder à ces serveurs via SSH

- `ssh root@192.168.2.100 -p 2221` pour **web1**
- `ssh root@192.168.2.100 -p 2222` pour **web2**

Une fois sur ces deux serveurs, vous pouvez installer **apache** et **PHP** avec la commande suivante:

```
apt-get install apache2 php5
```

Une fois l'installation de apache et PHP terminée sur les deux serveurs, accédez à **http://192.168.2.100/stats** et vous verrez maintenant que tout est vert. Cela veut dire que HaProxy fonctionne maintenant.

HAProxy

Statistics Report for pid 2610 on lb

> General process information

pid = 2610 (process #1, nbproc = 1)
uptime = 0d 0h00m16s
system limits: memmax = unlimited; ulimit-n = 4013
maxsock = 4013; maxconn = 2000; maxpipes = 0
current conns = 2; current pipes = 0/0
Running tasks: 1/4

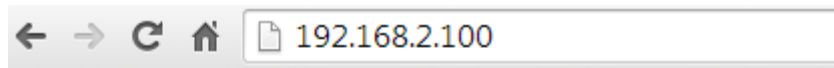
active UP backup UP
active UP, going down backup UP, going down
active DOWN, going up backup DOWN, going up
active or backup DOWN not checked
active or backup DOWN for maintenance (MAINT)

Note: UP with load-balancing disabled is reported as "NOLB".

cluster_web																						
	Queue			Session rate			Sessions					Bytes		Denied		Errors			Warnings		Status	LastC
	Cur	Max	Limit	Cur	Max	Limit	Cur	Max	Limit	Total	LbTot	In	Out	Req	Resp	Req	Conn	Resp	Retr	Redis		
Frontend				4	4	-	2	2	2 000	7		2 275	43 639	0	0	0					OPEN	
web1	0	0	-	0	0		0	0	-	0	0	0	0				0	0	0	0	16s UP	L4OK in
web2	0	0	-	0	0		0	0	-	0	0	0	0				0	0	0	0	16s UP	L4OK in
Backend	0	0		0	0		0	0	2 000	0	0	2 275	43 639	0	0		0	0	0	0	16s UP	

HA Stat UP

Vous pouvez accéder maintenant à <http://192.168.2.100> et vous verrez la page par défaut d'Apache, mais vous ne savez pas sur quel serveur Web vous êtes.



It works!

This is the default web page for this server.

The web server software is running but no content has been added, yet.

Page apache défaut

Test de la répartition de charge

Pour tester que la répartition de charge fonctionne, nous allons créer une page en PHP sur les deux serveurs web, cette page affichera le **hostname** de la machine.

Donc sur **les deux serveurs Web**, créez un fichier **index.php** dans le dossier **/var/www/** et copiez ceci dedans:

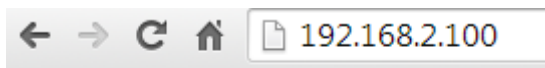
```
<?php
    echo "Un coucou de ".gethostname()."\n"
?>
```

]

Supprimez aussi le fichier index.html présent dans ce dossier.

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

Maintenant accédez à **http://192.168.2.100** et vous verrez ceci:



Un coucou de web1

Coucou de web1

Rafraîchissez la page vous de devez voir ceci:



Un coucou de web2

Coucou de web2

Cela veut dire que la répartition de charge entre les deux serveurs fonctionne :)

Ce qui est intéressant avec **HaProxy**, si apache est stoppé sur l'un des deux serveurs, **HaProxy va arrêter d'utiliser ce serveur.**

Si par exemple vous arrêtez Apache sur **web2**, vous verrez sur la page de stats que **web2** est marqué comme **DOWN**, si vous rafraîchissez la page **http://192.168.2.100** seul **Un coucou de web1** s'affichera, **HaProxy a décidé de lui même de ne plus utiliser Web2.**

HAProxy

Statistics Report for pid 2610 on lb

> General process information

pid = 2610 (process #1, nbproc = 1)
 uptime = 0d 0h17m41s
 system limits: memmax = unlimited; ulimit-n = 4013
 maxsock = 4013; maxconn = 2000; maxpipes = 0
 current conns = 2; current pipes = 0/0
 Running tasks: 1/4

active UP	backup UP
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)	

Display options

- [Hide 'NOLB'](#)
- [Disable 'NOLB'](#)
- [Refresh](#)
- [CSV export](#)

Note: UP with load-balancing disabled is reported as "NOLB".

cluster_web		Queue			Session rate			Sessions				Bytes		Denied		Errors			Warnings		Status	Last		
		Cur	Max	Limit	Cur	Max	Limit	Cur	Max	Limit	Total	LbTot	In	Out	Req	Resp	Req	Conn	Resp	Retr	Redis			
Frontend					1	6	-	2	4	2 000	76		30 655	361 775	0	0	4						OPEN	
web1		0	0	-	0	3		0	1	-	18	18	7 408	5 053	0	0	0	0	0	0	0	0	17m41s UP	L4OK
web2		0	0	-	0	3		0	1	-	21	12	5 047	3 078	0	0	3	0	9	0	0	0	11s DOWN	L4CON
Backend		0	0		0	5		0	3	2 000	30	30	30 655	361 775	0	0	3	0	9	0	0	0	17m41s UP	

Ha Stat un serveur down

Voilà ce qui met fin à ce tuto sur HaProxy.