

Nom	Prénom	Distribution	Version	Date
Divaret	Nathan	Debian 8.5	1.0	18/10/2016

HaProxy

Table des matières

HaProxy	1
Prérequis	2
Définition.....	3
Installation du service « Haproxy »	3
Configuration et test du service « HAproxy »	4
Mise en place des statistiques	7
Test permettant la vérification de la solution.....	8

Nom	Prénom	Distribution	Version	Date
Divaret	Nathan	Debian 8.5	1.0	18/10/2016

Prérequis

Pour réaliser cette procédure, nous avons besoin des éléments suivants :

- Trois machines virtuelles sous Debian 8.5.
- Trois IP virtuels de la 192.168.1.120 à la 192.168.1.122.
- Une machine Haproxy.
- Deux machines sous apache 2 qui feront office de web 1 et 2.

Tableau récapitulatif pour la machine virtuel avec HAProxy.

OS	Distribution	Version	C/S	Nom serveur HAProxy	IP principale serveur HAProxy
Debian	Linux	8.5	HAProxy	HaP	192.168.1.120/24

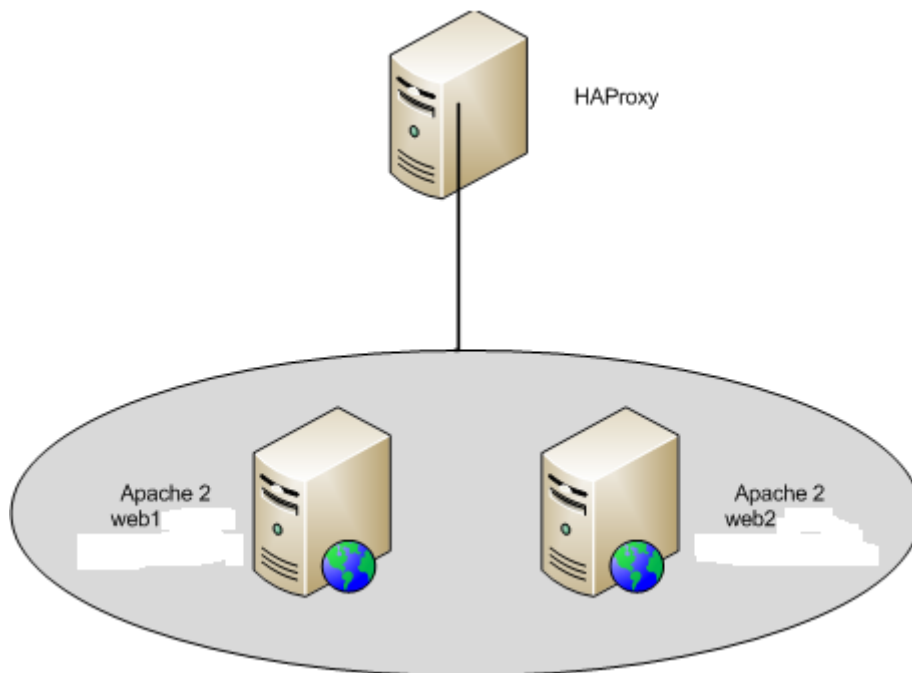
Tableau récapitulatif pour les deux machines virtuelles sous apache 2.

Nom serveur Web1	IP serveur Web1	Nom serveur Web2	IP serveur Web2
Web1	192.168.1.121/24	Web2	192.168.1.122/24

Carte réseau sur les serveurs :

Serveur HAProxy	Serveur Web1	Serveur Web2
1 en accès par pont /	1 carte en réseau	1 carte en réseau

Nom	Prénom	Distribution	Version	Date
Divaret	Nathan	Debian 8.5	1.0	18/10/2016



Définition

Le service **HAProxy** permet de faire de la répartition de charge entre les différents serveurs. La répartition de charges représente un ensemble de techniques qui distribuent une charge de travail sur plusieurs serveurs. Celle-ci assure une haute disponibilité entre eux et donc diminue l'indisponibilité d'un ou plusieurs services.

Installation du service « Haproxy »

Tout d'abord, nous allons renommer toutes les machines qui sont répertorié dans les différents tableaux ci-dessus.

Pour cela nous utiliseront la commande **apt** avec la mise à jour sur les trois serveurs

```
#apt update
```

Avant tout chose, nous devons ajouter une ligne dans **sources.list**

Editons le fichier /etc/apt/sources.list et ajoutons cette ligne :

```
deb http://ftp.fr.debian.org/debian/ jessie-backports main
```

Maintenant, nous allons pouvoir installer Haproxy avec les dépôts **backports**

Pour ce faire :

Nom	Prénom	Distribution	Version	Date
Divaret	Nathan	Debian 8.5	1.0	18/10/2016

```
root@HaP:~# apt install haproxy -t jessie-backports_
#apt-get install haproxy -t jessie-backports
```

Afin de vérifier la version d'Haproxy, nous tapons la commande :

Haproxy -v

```
root@HaP:~# haproxy -v
HA-Proxy version 1.6.9-2~bpo8+1 2016/10/06
Copyright 2000-2016 Willy Tarreau <willy@haproxy.org>
```

Maintenant nous devons configurer l'interface **eth1** du serveur Haproxy

```
auto eth1
iface eth1 inet static
-   address 10.22.100.210
    netmask 255.255.255.0
    network 192.168.1.0
    broadcast 192.168.1.255
```

Voici ce que cela donne avec la commande « **ifconfig** »

```
eth1      Link encap:Ethernet  HWaddr 08:00:27:b1:d6:c8
          inet adr:10.22.100.210  Bcast:192.168.1.255  Masque:255.255.255.0
          adr inet6: fe80::a00:27ff:feb1:d6c8/64 Scope:Lien
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:0 errors:0 dropped:0 overruns:0 frame:0
          TX packets:72 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 lg file transmission:1000
          RX bytes:0 (0.0 B)  TX bytes:4578 (4.4 KiB)
```

Maintenant, nous devons éditer le fichier **/etc/hosts** et nous ajoutons les deux serveurs Web sur lesquels nous avons installé apache2 avec la commande « **apt install apache2** »

```
127.0.0.1    localhost
127.0.1.1    HaP
192.168.1.121 Web1
192.168.1.122 Web2_
```

Configuration et test du service « Haproxy »

Pour ce faire, nous devons modifier le fichier « **/etc/haproxy/haproxy.cfg** » afin de configurer Haproxy.

```
root@HaP:~# nano /etc/haproxy/haproxy.cfg _
```

Nom	Prénom	Distribution	Version	Date
Divaret	Nathan	Debian 8.5	1.0	18/10/2016

Nous allons procéder à la configuration de Haproxy avec les sections **frontend** et **backend** qui se trouve à la fin du fichier de configuration.

Voici le contenu à modifier ou saisir pour les adresses IP des serveurs Web 1 et 2.

```
frontend public
listen haproxy
    bind *:80
    default_backend fermeweb

backend fermeweb
    balance roundrobin
    option httpclose
    option httpchk HEAD / HTTP/1.0
    server web1 192.168.1.121:80 check
    server web2 192.168.1.122:80 check
    stats uri /stats
    stats auth HaP:sio2a
    stats refresh 30s
```

Permet d'ajouter le port d'écoute, le 80.

Ce sont les deux Adresses IP des deux serveurs Web.

C'est le nom d'utilisateur et le mot de passe pour se connecter et afficher la page des statistiques.

Pour vérifier si notre fichier de configuration est correctement saisi nous pouvons utiliser la commande suivante :

```
root@HaP:~# haproxy -c -f /etc/haproxy/haproxy.cfg
```

Maintenant, que tout ceci est correctement effectuer, nous devons redémarrer le service pour que notre modification sur le fichier soit prise en compte.

```
#systemctl restart haproxy.service.
```

```
root@HaP:~# systemctl restart haproxy.service
root@HaP:~# _
```

Le service est correctement configurer et fonctionnelle.

```
root@HaP:~# systemctl status haproxy.service
• haproxy.service - HAProxy Load Balancer
  Loaded: loaded (/lib/systemd/system/haproxy.service; enabled)
  Active: active (running) since mar. 2016-10-18 10:16:38 CEST; 29s ago
  Docs: man:haproxy(1)
        file:/usr/share/doc/haproxy/configuration.txt.gz
  Process: 942 ExecStartPre=/usr/sbin/haproxy -f $CONFIG -c -q $EXTRA_OPTS (code=
exited, status=0/SUCCESS)
  Main PID: 943 (haproxy-systemd)
  CGroup: /system.slice/haproxy.service
          └─943 /usr/sbin/haproxy-systemd-wrapper -f /etc/haproxy/haproxy.cf...
            └─946 /usr/sbin/haproxy -f /etc/haproxy/haproxy.cfg -p /run/haprox...
              └─947 /usr/sbin/haproxy -f /etc/haproxy/haproxy.cfg -p /run/haprox...

oct. 18 10:16:38 HaP systemd[1]: Started HAProxy Load Balancer.
oct. 18 10:16:38 HaP haproxy-systemd-wrapper[943]: haproxy-systemd-wrapper: e...
oct. 18 10:16:38 HaP haproxy-systemd-wrapper[943]: [WARNING] 291/101638 (946)...
oct. 18 10:16:38 HaP haproxy[946]: Proxy public started.
oct. 18 10:16:38 HaP haproxy[946]: Proxy public started.
oct. 18 10:16:38 HaP haproxy[946]: Proxy haproxy started.
oct. 18 10:16:38 HaP haproxy[946]: Proxy haproxy started.
oct. 18 10:16:38 HaP haproxy[946]: Proxy fermeweb started.
oct. 18 10:16:38 HaP haproxy[946]: Proxy fermeweb started.
Hint: Some lines were ellipsized, use -l to show in full.
```

Le proxy s'est correctement démarré comme nous pouvons le constater avec les lignes en blanc.

Nom	Prénom	Distribution	Version	Date
Divaret	Nathan	Debian 8.5	1.0	18/10/2016

Nom	Prénom	Distribution	Version	Date
Divaret	Nathan	Debian 8.5	1.0	18/10/2016

Mise en place des statistiques

Maintenant que tout fonctionne, nous allons voir et nous connecter pour vérifier les statistiques des serveurs apaches2 (web1 et web2) qui sont actuellement en fonctions.

Il faut pour cela, se connecter via un navigateur avec l'adresse du proxy suivie du nom du fichier test qui est directement installé lors de l'installation de Haproxy mais il va nous demander une authentification.

192.168.1.120/statsHaprxxy

Authentification requise

http://192.168.1.120 nécessite un nom d'utilisateur et un mot de passe.

Votre connexion à ce site n'est pas privée.

Nom d'utilisateur :

Mot de passe :

Les identifiants ont été défini dans le fichier de configuration ci-dessus. Les identifiants sont très simpliste mais vous devez en mettre des plus « forts ».

Maintenant, nous avons accès aux statistiques des deux serveurs Web apache qui seront rafraichi toutes les 30 secondes ou par vous-même avec « F5 ».

HAProxy version 1.6.9, released 2016/08/30

Statistics Report for pid 947

> General process information

pid = 947 (process #1, nbproc = 1)
 uptime = 0d 0h09m14s
 system limits: memmax = unlimited; ulimit-n = 4013
 maxsock = 4013; maxconn = 2000; maxpipes = 0
 current conns = 2; current pipes = 0/0; conn rate = 1/sec
 Running tasks: 1/8; idle = 100 %

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)
 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](#)

External resources:

- [Primary site](#)
- [Updates \(v1.5\)](#)
- [Online manual](#)

public																														
	Queue			Session rate			Sessions					Bytes		Denied	Errors			Warnings	Server											
	Cur	Max	Limit	Cur	Max	Limit	Cur	Max	Limit	Total	LbTot	Last	In	Out	Req	Resp	Req	Conn	Resp	Retr	Redis	Status	LastChk	Wght	Act	Bck	Chk	Dwn	Dwntme	Thrtle
Frontend	0	0	0	0	0	0	0	0	0	2 000	0	0	0	0	0	0	0	0	0	0	0	OPEN								
Backend	0	0	0	0	0	0	0	0	0	200	0	0	0	0	0	0	0	0	0	0	0	0m14s UP								

haproxy																														
	Queue			Session rate			Sessions					Bytes		Denied	Errors			Warnings	Server											
	Cur	Max	Limit	Cur	Max	Limit	Cur	Max	Limit	Total	LbTot	Last	In	Out	Req	Resp	Req	Conn	Resp	Retr	Redis	Status	LastChk	Wght	Act	Bck	Chk	Dwn	Dwntme	Thrtle
Frontend	0	0	0	1	2	0	2	2	2	2 000	19	0	0	0	7 743	268 907	0	0	1	0	0	OPEN								
Backend	0	0	0	0	0	0	0	0	0	200	0	0	0	0	0	213	0	0	0	0	0	0m14s UP								

fermeweb																															
	Queue			Session rate			Sessions					Bytes		Denied	Errors			Warnings	Server												
	Cur	Max	Limit	Cur	Max	Limit	Cur	Max	Limit	Total	LbTot	Last	In	Out	Req	Resp	Req	Conn	Resp	Retr	Redis	Status	LastChk	Wght	Act	Bck	Chk	Dwn	Dwntme	Thrtle	
web1	0	0	0	0	0	0	0	0	0	1	1	1	58s	380	488	0	0	0	0	0	0m14s UP	L7OK/200 in 2ms	1	Y	-	0	0	0s	-		
web2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0m14s UP	L7OK/200 in 2ms	1	Y	-	0	0	0s	-
Backend	0	0	0	1	2	0	1	1	1	200	17	1	0s	7 743	268 894	0	0	0	0	0	0m14s UP		2	2	0	0	0	0s			

Nom	Prénom	Distribution	Version	Date
Divaret	Nathan	Debian 8.5	1.0	18/10/2016

Maintenant, le même tableau mais pas au même moment, vous verrez qu'il y a eu des changements ce sont l'activé de ces deux serveurs, les flèches indiquent les changements entre ces deux prise.

HAProxy version 1.6.9, released 2016/08/30

Statistics Report for pid 947

> General process information

pid = 947 (process #1, nbproc = 1)
 uptime = 0d 0h 10m 11s
 system limits: memmax = unlimited; ulimit-n = 4013
 maxsock = 4013; maxconn = 2000; maxpipes = 0
 current conns = 2; current pipes = 0/0; conn rate = 2/sec
 Running tasks: 1/8; 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

backup UP
 backup UP, going down
 backup DOWN, going up
 not checked

Note: "NOLB"/"DRAIN" = UP with load-balancing disabled.

Display option:
 Scope:
 Hide DOWN servers
 Disable refresh
 Refresh now
 CSV export

External resources:
 Primary site
 Updates (v1.6)
 Online manual

public																															
	Queue			Session rate			Sessions				Bytes		Denied		Errors		Warnings		Server												
	Cur	Max	Limit	Cur	Max	Limit	Cur	Max	Limit	Total	LbTot	Last	In	Out	Req	Resp	Req	Conn	Resp	Retr	Redis	Status	LastChk	Wght	Act	Bck	Chk	Dwn	Dwntme	Thrtl	
Frontend	0	0	0	0	0	-	0	0	2 000	0	0	0	0	0	0	0	0	0	0	0	0	OPEN									

haproxy																															
	Queue			Session rate			Sessions				Bytes		Denied		Errors		Warnings		Server												
	Cur	Max	Limit	Cur	Max	Limit	Cur	Max	Limit	Total	LbTot	Last	In	Out	Req	Resp	Req	Conn	Resp	Retr	Redis	Status	LastChk	Wght	Act	Bck	Chk	Dwn	Dwntme	Thrtl	
Frontend	0	0	0	0	0	-	2	2	2 000	22	0	0	8 757	307 476	0	0	0	0	0	0	0	OPEN									
Backend	0	0	0	0	0	0	0	0	200	0	0	?	0	401	0	0	0	0	0	0	0	10m11s UP		0	0	0	0	0	0	0	0

fermeweb																														
	Queue			Session rate			Sessions				Bytes		Denied		Errors		Warnings		Server											
	Cur	Max	Limit	Cur	Max	Limit	Cur	Max	Limit	Total	LbTot	Last	In	Out	Req	Resp	Req	Conn	Resp	Retr	Redis	Status	LastChk	Wght	Act	Bck	Chk	Dwn	Dwntme	Thrtl
web1	0	0	-	0	1	0	0	1	-	1	1	1m55s	380	488	0	0	0	0	0	0	0	10m11s UP	L7OK/200 in 2ms	1	Y	-	0	0	0	0s
web2	0	0	-	0	0	0	0	0	-	0	0	?	0	0	0	0	0	0	0	0	0	10m11s UP	L7OK/200 in 2ms	1	Y	-	0	0	0	0s
Backend	0	0	0	1	2	0	1	1	200	19	1	0s	8 757	307 075	0	0	0	0	0	0	0	10m11s UP		2	2	0	0	0	0	0s

En effet, ce sont les « Bytes » qui changent car des données transitent et le serveur effectue des requêtes.

Test permettant la vérification de la solution

Nous allons maintenant, réaliser quelques tests, pour vérifier le bon fonctionnement de cette solution.

Les deux serveurs Web1 et Web2 sont actifs et soudainement le serveur Web tombe dû à une surcharge ou une attaque extérieure. Nous allons voir comment les statistiques vont soudainement changer et que vont-elle afficher.

Les deux serveurs sont opérationnels.

fermeweb																														
	Queue			Session rate			Sessions				Bytes		Denied		Errors		Warnings		Server											
	Cur	Max	Limit	Cur	Max	Limit	Cur	Max	Limit	Total	LbTot	Last	In	Out	Req	Resp	Req	Conn	Resp	Retr	Redis	Status	LastChk	Wght	Act	Bck	Chk	Dwn	Dwntme	Thrtl
web1	0	0	-	0	1	0	0	1	-	1	1	9m20s	380	488	0	0	0	0	0	0	0	47s UP	L7OK/200 in 1ms	1	Y	-	0	2	3m28s	-
web2	0	0	-	0	0	0	0	0	-	0	0	?	0	0	0	0	0	0	0	0	0	49s UP	L7OK/200 in 1ms	1	Y	-	3	1	1m47s	-
Backend	0	0	0	4	7	0	1	1	200	99	1	0s	49 317	1 851 576	0	0	0	0	0	0	0	49s UP		2	2	0	0	1	1m47s	-

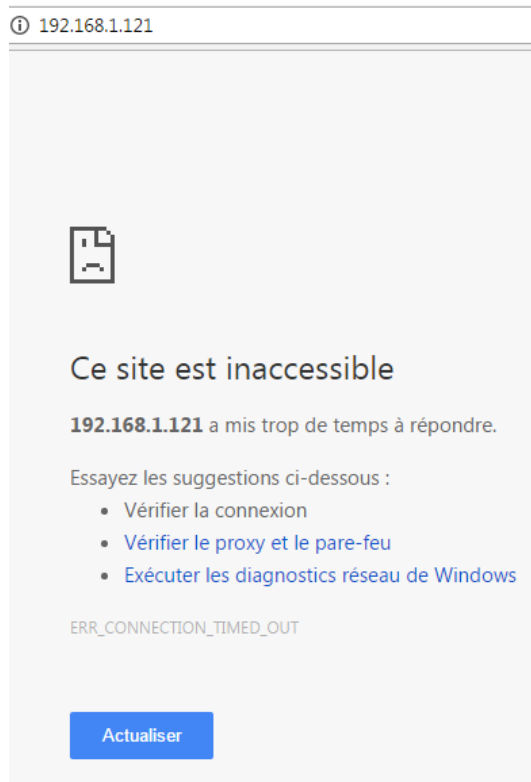
Lorsque le serveur Web1 tombe.

fermeweb																														
	Queue			Session rate			Sessions				Bytes		Denied		Errors		Warnings		Server											
	Cur	Max	Limit	Cur	Max	Limit	Cur	Max	Limit	Total	LbTot	Last	In	Out	Req	Resp	Req	Conn	Resp	Retr	Redis	Status	LastChk	Wght	Act	Bck	Chk	Dwn	Dwntme	Thrtl
web1	0	0	-	0	1	0	0	1	-	1	1	10m16s	380	488	0	0	0	0	0	0	0	1s DOWN	L4TOUT in 2003ms	1	Y	-	9	3	3m29	-
web2	0	0	-	0	0	0	0	0	-	0	0	?	0	0	0	0	0	0	0	0	0	1m45s UP	L7OK/200 in 1ms	1	Y	-	3	1	1m47	-
Backend	0	0	0	2	7	0	1	1	200	114	1	0s	56 922	2 142 449	0	0	0	0	0	0	0	1m45s UP		1	1	0	0	1	1m47	-

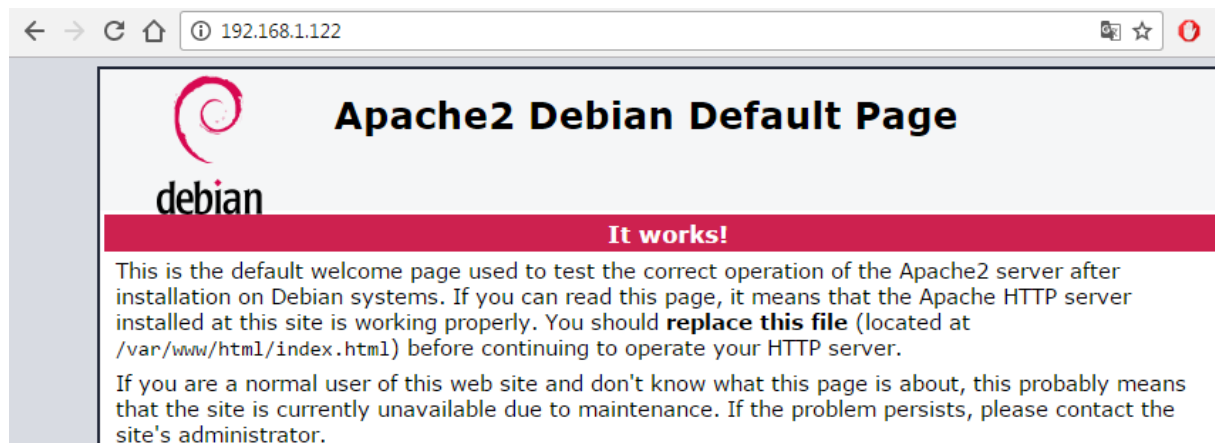
Nous pouvons constater que le Web1 est rouge car il est inactif (Dead), il affiche une latence de 2003ms.

Nom	Prénom	Distribution	Version	Date
Divaret	Nathan	Debian 8.5	1.0	18/10/2016

On peut également constater que si on saisit l'adresse du serveur Web1 est inaccessible du faite qu'il ait « tombé »



Contrairement au serveur Web2 qui fonctionne et nous renvoie la page par défaut.



Pour conclure, nous pouvons dire que le service Haproxy est fonctionnel car celui-ci permet de répartir la charge de travail entre les différents serveurs Web 1 et 2 dans notre cas. Et ainsi faire de la tolérance de pannes entres ces serveurs.