

# Production 3 : HAProxy : High Availability + Load Balancing

---

## *Qu'est-ce que HeartBeat ?*

---

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.

---

## *Machines nécessaires :*

---

- Une VM HAProxy de répartition de charge
  - Hostname : HAProxy
  - Network adapter : carte 1 en segment LAN « LAN » 172.20.0.20/24
  - Network adapter : carte 2 en segment LAN « WAN » 192.168.0.20/24
- Serveur 1 debian : srvweb1
  - Cartes réseaux segment LAN et statique (172.20.0.11/24)
- Serveur 2 Debian : srvweb2
  - Cartes réseaux segment LAN et statique (172.20.0.12/24)
- Client Windows
  - Une carte réseau en segment LAN d'adresse IP statique (192.168.0.100/24)

---

## *Préparations des machines : installation d'un site web*

---

**VOUS POUVEZ PREPARER UN PREMIER SERVEUR ET EN FAIRE UNE COPIE OU FAIRE LES DEUX SEPAREMENT.**

---

### *Installation de paquets nécessaires*

---

Sur les deux serveurs webs, récupérer les paquets d'apache à l'aide de la commande apt install

```
apt install apache2 php wget unzip
```

Sur le serveur HAProxy, il faut récupérer HAProxy

```
apt install haproxy
```

Les deux serveurs doivent contenir, en plus du système de base, une activité : on choisira ici la fonctionnalité de serveur web. Nous allons donc chercher des pages web html à l'aide de la commande **wget**.

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

Décompresser l'archive et changer le document Root d'Apache 2 en ajoutant le dossier.

```
unzip master.zip
```

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

```
GNU nano 5.4 /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

# Available loglevels: trace8, ..., trace1, debug, info, notice, warn,
# error, crit, alert, emerg.
# It is also possible to configure the loglevel for particular
# modules, e.g.
#LogLevel info ssl:warn

ErrorLog ${APACHE_LOG_DIR}/error.log
CustomLog ${APACHE_LOG_DIR}/access.log combined

# For most configuration files from conf-available/, which are
# enabled or disabled at a global level, it is possible to
# include a line for only one particular virtual host. For example the
# following line enables the CGI configuration for this host only
# after it has been globally disabled with "a2disconf".
#Include conf-available/serve-cgi-bin.conf
</VirtualHost>

# vim: syntax=apache ts=4 sw=4 sts=4 sr noet

Nom du fichier à écrire: /etc/apache2/sites-available/000-default.conf
G Aide M-D Format DOS M-A Ajout (à la fin) M-B Copie de sécu.
C Annuler M-M Format Mac M-P Ajout (au début) T Parcourir
```

---

### Vérification de communication

---

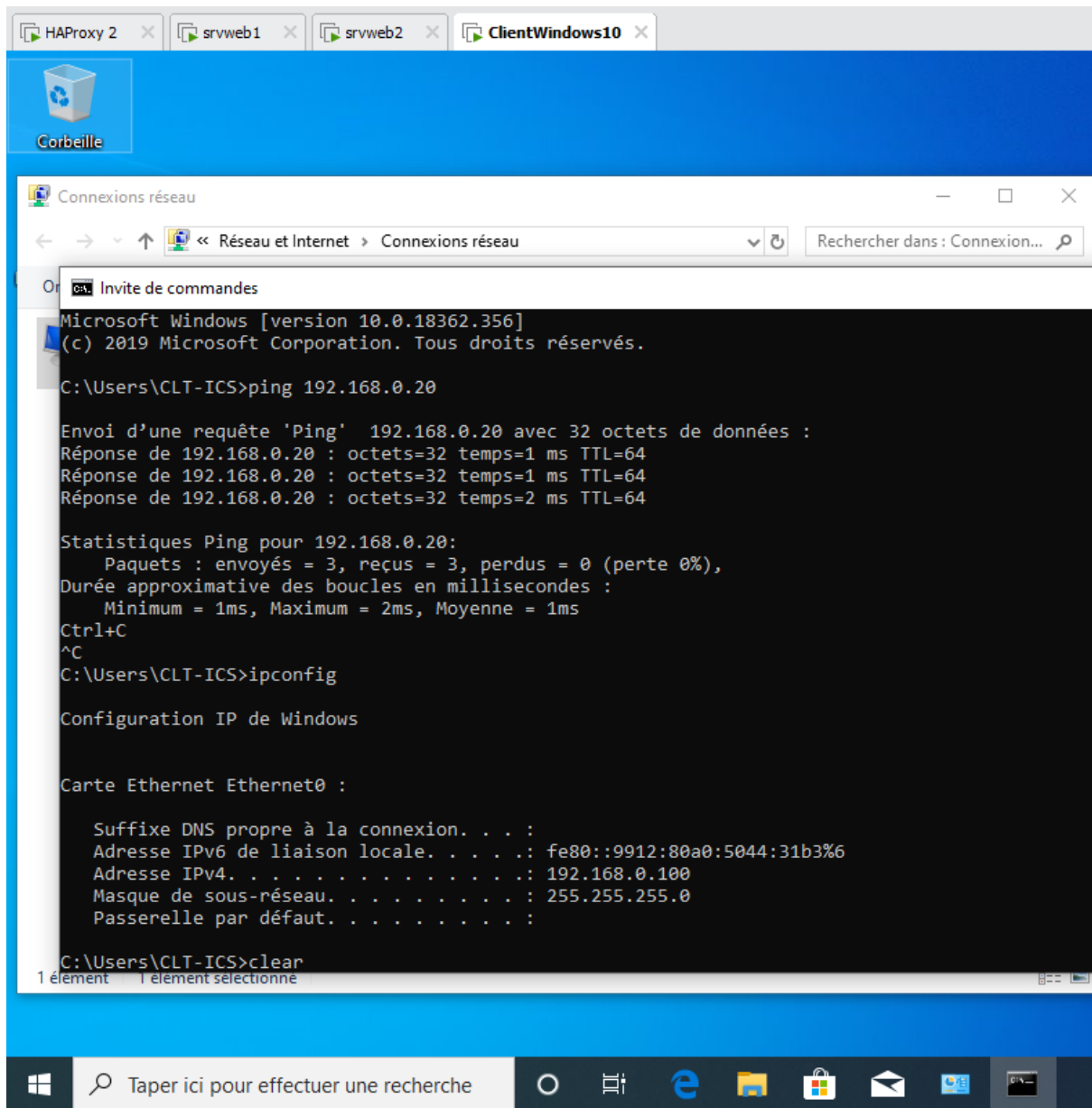
On effectue des tests ping depuis l'HAProxy vers les deux serveurs webs.

```
HAProxy 2 x srvweb1 x srvweb2 x ClientWindows10 x
debian login: root
Password:
Linux debian 5.10.0-17-amd64 #1 SMP Debian 5.10.136-1 (2022-08-13) x86_64

The programs included with the Debian GNU/Linux system are free software; the
exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
Last login: Tue Nov 22 11:19:56 CET 2022 on tty1
root@debian:~# ping 172.20.0.11
PING 172.20.0.11 (172.20.0.11) 56(84) bytes of data.
64 bytes from 172.20.0.11: icmp_seq=1 ttl=64 time=0.753 ms
64 bytes from 172.20.0.11: icmp_seq=2 ttl=64 time=0.336 ms
^C
--- 172.20.0.11 ping statistics ---
2 packets transmitted, 2 received, 0% packet loss, time 1001ms
rtt min/avg/max/mdev = 0.336/0.544/0.753/0.208 ms
root@debian:~# ping 172.20.0.11
PING 172.20.0.11 (172.20.0.11) 56(84) bytes of data.
64 bytes from 172.20.0.11: icmp_seq=1 ttl=64 time=0.435 ms
64 bytes from 172.20.0.11: icmp_seq=2 ttl=64 time=0.883 ms
^C
--- 172.20.0.11 ping statistics ---
2 packets transmitted, 2 received, 0% packet loss, time 1012ms
rtt min/avg/max/mdev = 0.435/0.659/0.883/0.224 ms
root@debian:~# ping 172.20.0.12
PING 172.20.0.12 (172.20.0.12) 56(84) bytes of data.
64 bytes from 172.20.0.12: icmp_seq=1 ttl=64 time=0.452 ms
64 bytes from 172.20.0.12: icmp_seq=2 ttl=64 time=0.836 ms
^C
--- 172.20.0.12 ping statistics ---
2 packets transmitted, 2 received, 0% packet loss, time 1007ms
rtt min/avg/max/mdev = 0.452/0.644/0.836/0.192 ms
root@debian:~# _
```

Depuis la machine cliente vers l'HAProxy



---

### 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 le lignes suivantes à la fin du fichier :

```
# Configuration du balancement
listen clusterWebStadiumCompany
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
```

```
GNU nano 5.4 /etc/haproxy/haproxy.cfg *
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 clusterWebStadiumCompany
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 srvweb1 172.20.0.11:80 check
server srvweb2 172.20.0.12:80 check

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

^G Aide      ^O Écrire    ^W Chercher  ^K Couper    ^T Exécuter  ^C EmplacementM-U Annuler
^X Quitter   ^R Lire fich.^N Remplacer  ^U Coller    ^J Justifier ^_ Aller ligneM-E Refaire
```

## Explications

***listen cluster\_web adresseIPHaProxY:80***: Cette directive nous permet de spécifier sur quelle adresse IP et port 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.

## Explications bonus

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.

On redémarre le service pour validation

***service haproxy restart***

***service haproxy status -> Active running***

```
root@debian:~# service haproxy restart
root@debian:~# service haproxy status
• haproxy.service - HAProxy Load Balancer
  Loaded: loaded (/lib/systemd/system/haproxy.service; enabled; vendor preset: enabled)
  Active: active (running) since Tue 2022-11-22 11:38:56 CET; 5s ago
    Docs: man:haproxy(1)
           file:/usr/share/doc/haproxy/configuration.txt.gz
  Process: 577 ExecStartPre=/usr/sbin/haproxy -f $CONFIG -c -q $EXTRA_OPTS (code=exited, status=0)
 Main PID: 579 (haproxy)
    Tasks: 2 (limit: 2300)
   Memory: 34.1M
      CPU: 28ms
   CGroup: /system.slice/haproxy.service
           └─579 /usr/sbin/haproxy -Ws -f /etc/haproxy/haproxy.cfg -p /run/haproxy.pid -S /run/h
             581 /usr/sbin/haproxy -Ws -f /etc/haproxy/haproxy.cfg -p /run/haproxy.pid -S /run/h

nov. 22 11:38:56 debian systemd[1]: Starting HAProxy Load Balancer...
nov. 22 11:38:56 debian haproxy[579]: Proxy clusterWebStadiumCompany started.
nov. 22 11:38:56 debian haproxy[579]: Proxy clusterWebStadiumCompany started.
nov. 22 11:38:56 debian haproxy[579]: [NOTICE] 325/113856 (579) : New worker #1 (581) forked
nov. 22 11:38:56 debian systemd[1]: Started HAProxy Load Balancer.
lines 1-19/19 (END)
```

---

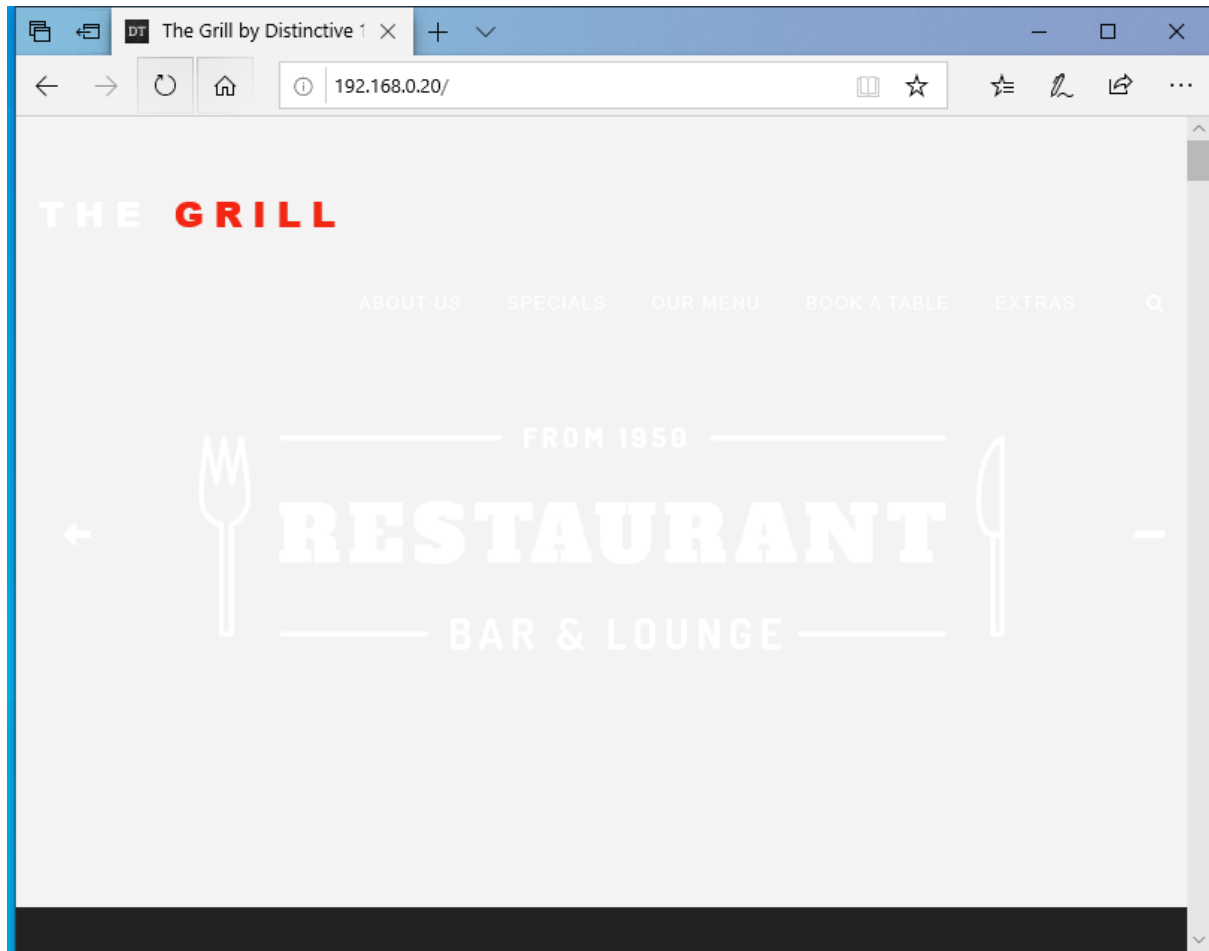
*Validation de l'HAProxy*

---

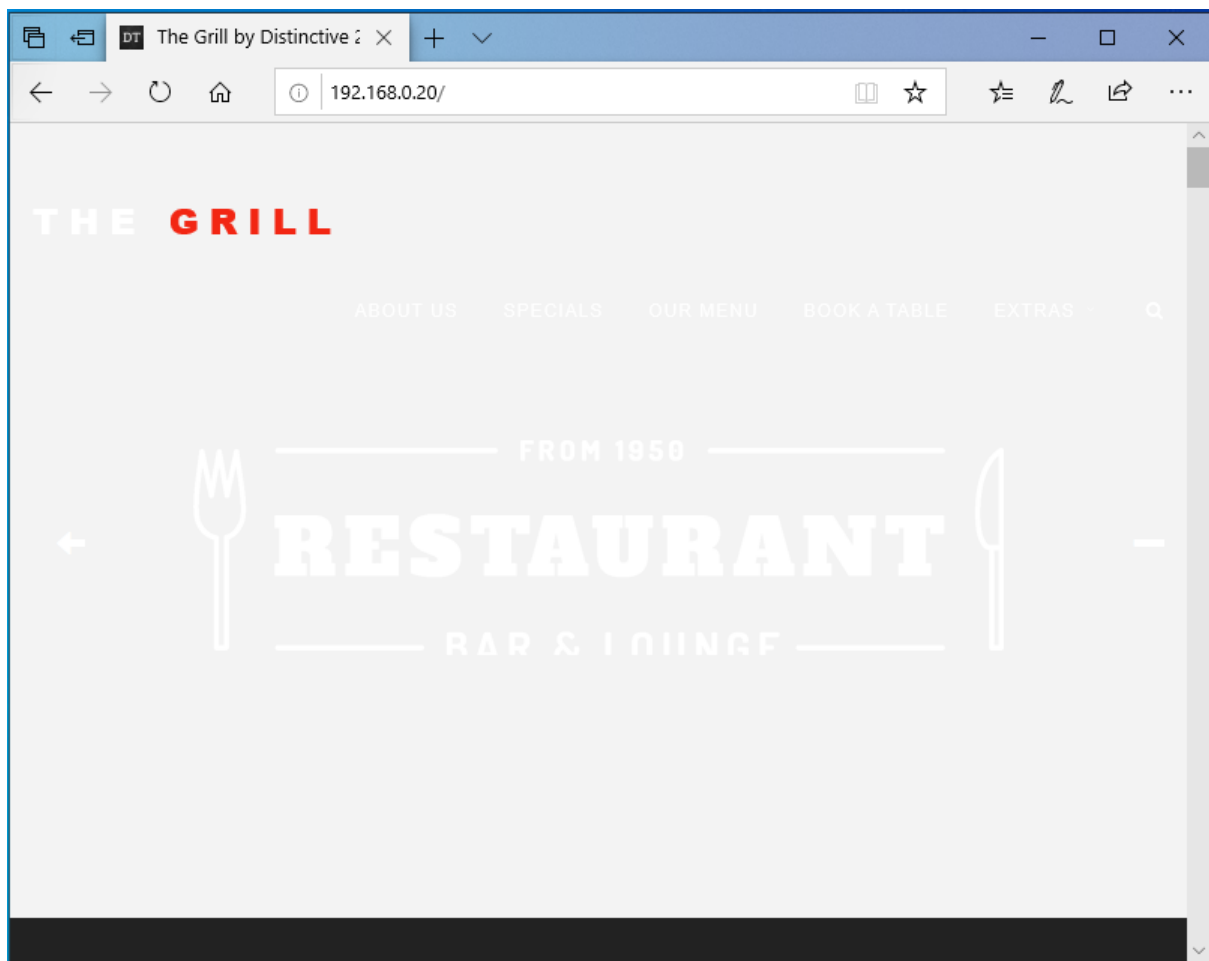


Pour vérifier le balancement : RDV machine client-> IE ou Edge -> 192.168.0.20, vous devez voir The Grill 1, actualisez, vous devez voir The Grill 2, le balancement est bien fonctionnel.

1<sup>ère</sup> essai



2<sup>ème</sup> essai (après 2 actualisations)



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 up en vert.

srvweb2

ClientWindows10

DT

Statistics Report for HAI

+

▼

←

→

↺

🏠

192.168.0.20/statistique

📖

☆

⌵

🔍

🔗

⋮

# HAProxy

## Statistics Report for pid 581 on debian

### > General process information

pid = 581 (process #1, nbproc = 1, nbthread = 1)

uptime = 0d 0h04m19s

system limits: memmax = unlimited; ulimit-n = 524287

maxsock = 524287; maxconn = 262124; maxpipes = 0

current conns = 1; current pipes = 0/0; conn rate = 1/sec; bit rate = 0.000 kbps

Running tasks: 1/11; 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](#)
- [JSON export \(schema\)](#)

clusterWebStadiumCompany

	Queue			Session rate			Sessions					Bytes		Denied		Errors		Warnings		Status		LastC	
	Cur	Max	Limit	Cur	Max	Limit	Cur	Max	Limit	Total	LbTot	Last	In	Out	Req	Resp	Req	Conn	Resp	Retr	Redis	Status	LastC
Frontend				1	30	-	1	6	262 124	91			36 597	1 746 248	0	0	0					OPEN	
srvweb1	0	0	-	0	15		0	3	-	44	44	2m6s	18 079	831 881	0	0	0	0	0	0	0	4m19s UP	L4OK in
srvweb2	0	0	-	0	15		0	3	-	44	44	2m6s	17 767	897 771	0	0	0	0	0	0	0	4m19s UP	L4OK in
Backend	0	0		0	30		0	5	26 213	88	88	0s	36 597	1 746 248	0	0	0	0	0	0	0	4m19s UP	

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.

DT

Statistics Report for HAI

+

▼

←

→

↺

🏠

🕒

192.168.0.20/statistique

📖

☆

⌘

🔍

🔗

⋮

# HAProxy

## Statistics Report for pid 581 on debian

> General process information

pid = 581 (process #1, nbproc = 1, nbthread = 1)

uptime = 0d 0h00m11s

system limits: memmax = unlimited; ulimit-n = 524287

maxsock = 524287; maxconn = 262124; maxpipes = 0

current conns = 1; current pipes = 0/0; conn rate = 1/sec; bit rate = 0.000 kbps

Running tasks: 1/11; 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

Display option:

Scope :

Hide "DOWN" servers

Disable refresh

Refresh now

CSV export

JSON export (schema)

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

clusterWebStadiumCompany

	Queue			Session rate			Sessions					Bytes		Denied		Errors			Warnings					
	Cur	Max	Limit	Cur	Max	Limit	Cur	Max	Limit	Total	LbTot	Last	In	Out	Req	Resp	Req	Conn	Resp	Retr	Redis	Status	Las	
Frontend				1	30	-	1	6	262 124	97			38 755	1 812 434	0	0	0						OPEN	
srweb1	0	0	-	0	15		0	3	-	45	45	1m52s	18 324	832 320	0	0	0	0	0	0	0	0	17s DOWN	L4COM
srweb2	0	0	-	0	15		0	3	-	45	45	1m22s	18 012	898 210	0	0	0	0	0	0	0	0	2s DOWN	L4COM
Backend	0	0		0	30		0	5	26 213	90	90	0s	38 755	1 812 434	0	0	0	0	0	0	0	0	2s DOWN	

http://192.168.0.20/statistique#clusterWebStadiumCompany