

Installer un serveur de Load Balancing - HAProxy

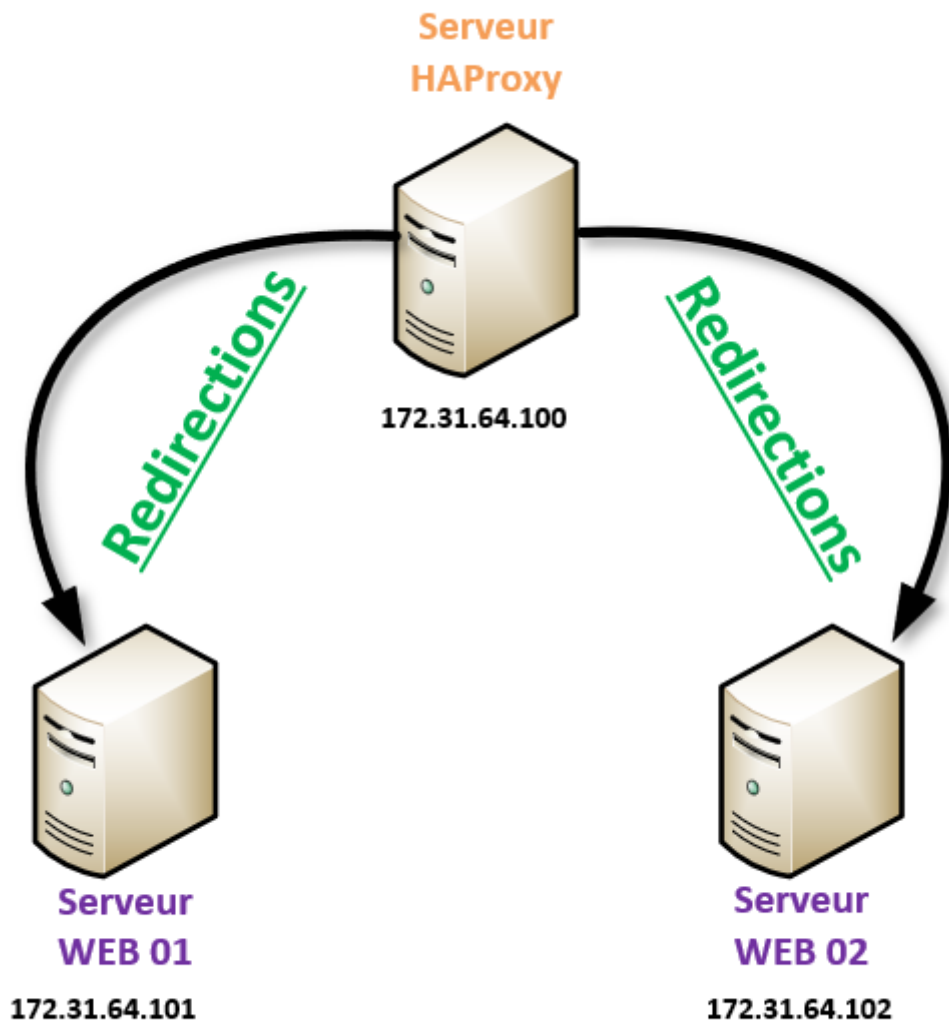


Introduction :

Contexte : Le load balancing permet la tolérance de panne mais aussi la continuité de service lors une forte demande utilisateur.

Concrètement, un serveur de load balancing ou en français, serveur de répartition de charge, redirige l'utilisateur vers plusieurs serveurs de même fonction. Un utilisateur pourra sans même s'en rendre compte consulter deux serveurs web entre deux pages d'un site. Ce mécanisme permet donc de traiter un nombre d'utilisateur bien plus important qu'un serveur seul permettrait.

Schéma :



Étape 1 - Prérequis

Conformément au schéma ci-dessus, pour réaliser l'installation d'HAProxy, vous aurez besoin d'au minimum, deux serveurs web. Pour comprendre correctement le load balancing, j'ai modifié la page d'accueil de chacun de ces sites web afin qu'ils affichent leurs noms. Notons qu'en production, ce serait le même site web qui serait stocké dans les serveurs web.

Serveur web 01 :

← → ↻ ⚠ Non sécurisé | 172.31.64.101

SITE 1

Serveur web 02 :

← → ↻ ⚠ Non sécurisé | 172.31.64.102

SITE 2

Étape 2 - Mise à jour du système et installation d'HAProxy

Premièrement nous allons mettre à jour notre serveur. Un système à jour est un système fiable !

```
apt-get update ; apt-get upgrade -y
```

En suite nous allons installer le paquet haproxy :

```
apt-get install haproxy -y
```

Étape 3 - Modification de la configuration d'HAProxy

Maintenant, modifions la configuration du paquet :

```
nano /etc/haproxy/haproxy.cfg
```

Nous laisserons la configuration par défaut et ajouterons à la fin du fichier de configuration les lignes suivantes :

```
# FrontEND
frontend frontend-base
    # IP d'écoute du proxy.
    bind 172.31.64.100:80
```

```
default_backend backend-base
option forwardfor
# BackEND
backend backend-base
    # On utilise le mode Roundrobin
    balance roundrobin
    # On liste les serveurs de backend
    server serveur-web-1 172.31.64.101:80 check
    server serveur-web-2 172.31.64.102:80 check
```

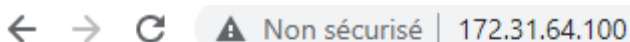
Pour appliquer les modifications, il va falloir redémarrer le service haproxy. Faites la commandes ci dessous :

```
/etc/init.d/haproxy restart
```

Étape 4 - Tests de la configuration d'HAProxy

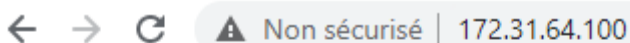
Pour tester si la configuration fonctionne, nous allons utiliser un navigateur et consulter l'adresse IP du serveur HAPROXY. Celui ci écoute sur le port 80 (HTTP) et redirigera toutes nos requêtes vers l'un des deux serveurs du cluster.

Nous nous y rendons une première fois :



SITE 1

Nous nous y rendons une seconde fois :



SITE 2

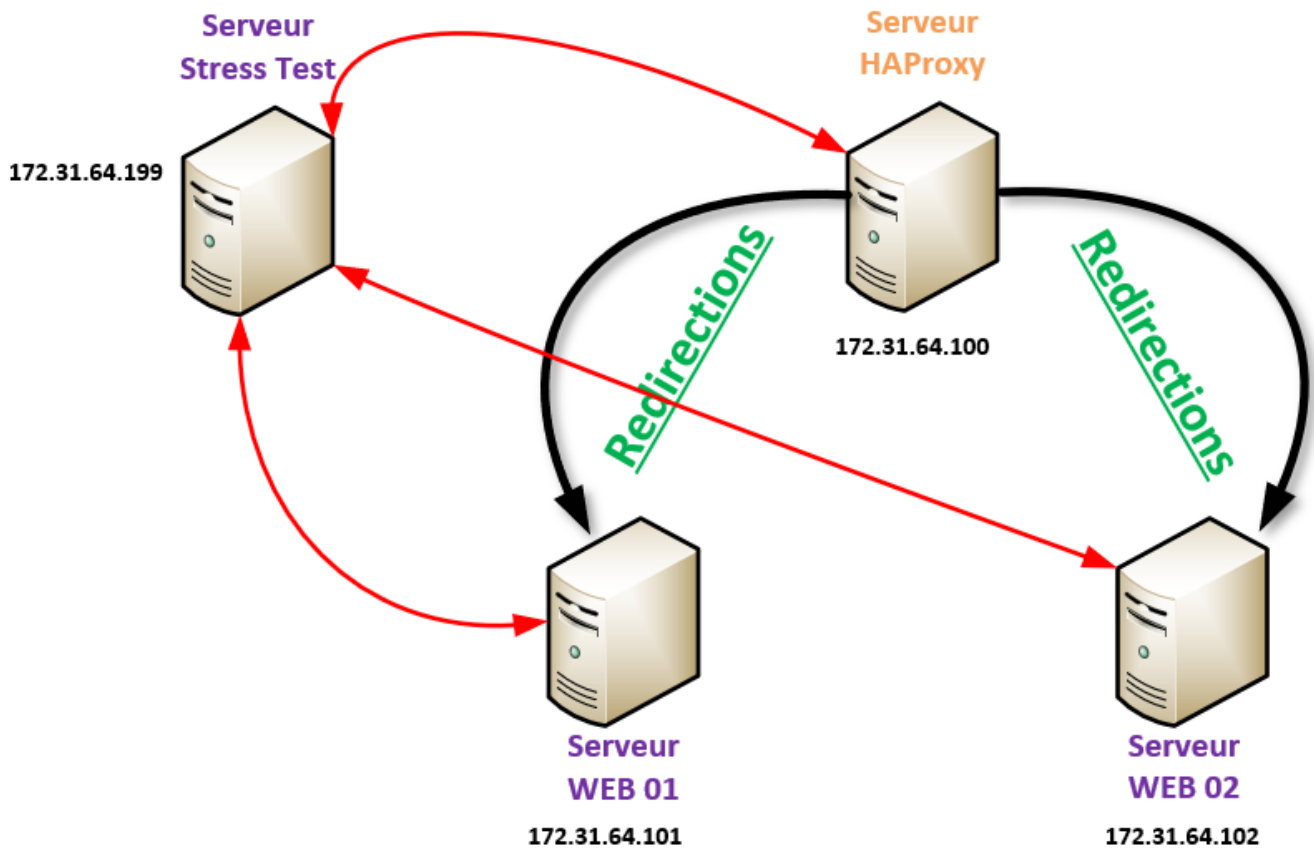
Le load balancing fonctionne belle est bien, nous avons donc correctement installé HAProxy.

Étape 5 - Stress Test.

Un stress test est le fait de surcharger un serveur pour en connaître ses limites. Nous allons donc utiliser une commande venant d'apache2. Cette commande permet de simuler un nombre très important de connexions sur un serveur Web.

Afin d'effectuer une comparaison fiable, nous allons utiliser des serveurs web ayant la même puissance à chaque fois.

1 - Schéma :



2 - Installation :

Pour installer apache2, il faudra exécuter les commandes suivantes sur votre machine de stress test :

```
apt-get install apache2 -y
```

Aucune configuration pour Apache2 n'est nécessaire.

3 - Stress d'un serveur web (seul) :

Nous allons donc procéder à la surcharge d'un de nos serveur web et c'est pourquoi, dans la commande suivante nous mettrons l'adresse IP du serveur web 01.

```
ab -n 100000000 -c 9999 172.31.64.101/
```

- L'argument -n permet de spécifier le nombre total de requêtes à envoyer.
- L'argument -c définit le nombre de requêtes à faire simultanément.
- Pour ce qui est de l'adresse IP, n'oubliez pas le / à la fin sinon cela ne fonctionnera pas.

Résultat :

```
root@STRESS-TEST:~# ab -n 100000000 -c 9999 172.31.64.101/
This is ApacheBench, Version 2.3 <$Revision: 1843412 $>
Copyright 1996 Adam Twiss, Zeus Technology Ltd, http://www.zeustech.net/
Licensed to The Apache Software Foundation, http://www.apache.org/

Benchmarking 172.31.64.101 (be patient)
apr_socket_recv: Connection timed out (110)
Total of 372763 requests completed
```

Cela signifie qu'au bout de **372 763 requêtes**, le serveur web 01 ne peut plus tenir la charge.

4 - Stress test avec load balancing (2 machines) :

C'est le cluster entier que nous allons surcharger, pour cela, nous avons simplement à renseigner l'adresse IP du serveur HAProxy.

```
ab -n 100000000 -c 9999 172.31.64.100/
```

Résultat :

```
root@STRESS-TEST:~# ab -n 100000000 -c 9999 172.31.64.100/
This is ApacheBench, Version 2.3 <$Revision: 1843412 $>
Copyright 1996 Adam Twiss, Zeus Technology Ltd, http://www.zeustech.net/
Licensed to The Apache Software Foundation, http://www.apache.org/

Benchmarking 172.31.64.100 (be patient)
apr_socket_recv: Connection timed out (110)
Total of 529522 requests completed
```

Cela signifie qu'au bout de **529 522 requêtes**, le cluster ne peut plus tenir la charge.

5 - Conclusion :

Nous avons bel et bien installé une solution de load balancing permettant de réaliser une continuité de service.

Revision #19

Created 11 December 2020 20:01:23 by Nehemie

Updated 26 December 2022 13:41:14 by Nehemie