

HA Proxy

Principes

Pour assurer la continuité de service d'un serveur Web, il faut disposer d'au moins deux serveurs. Ceux-ci peuvent fonctionner en mode **actif/passif** avec l'outil [Heartbeat](#) où un seul serveur est actif à un instant donné, et l'autre prend le relais en cas de panne du premier.

Pour assurer une performance supplémentaire, on peut aussi choisir un fonctionnement en mode **actif/actif** avec l'outil HA Proxy. Dans cette configuration, les deux (ou plus) serveurs fonctionnent en simultané : on doit donc mettre en place un **répartiteur de charge** qui va renvoyer les demandes vers les différents serveurs selon différents critères : chacun son tour (**round robin**, en fonction de la puissance des machines, etc.

Mise en place

Pour installer HA Proxy :

```
apt update
apt upgrade
apt install haproxy
```

Pour vérifier la version installée : `haproxy -v`

Le démon ne démarre pas car le fichier de configuration par défaut ne contient pas toutes les directives nécessaires. Il est nécessaire de procéder à une configuration minimale.

Configuration de base

HA Proxy fonctionne sur deux parties :

- la partie au contact des utilisateurs : **Front end**
- la partie au contact des serveurs Web : **Back end**

La configuration complète du fichier est décrite ici

<https://cbonte.github.io/haproxy-dconv/2.6/configuration.html>.

Le fichier de configuration est **`/etc/haproxy/haproxy.cfg`** . Dans ses options de base, on trouvera :

```
frontend <nomF_E>
  bind <IP_publiqueHaProxy>:<port>
  default_backend <nomB_E>

backend <nomB_E>
  balance <modeRepartition>
```

```
server <nomServ1> <ip_serv1>:<port_serv1>  
server <nomServ2> <ip_serv2>:<port_serv1>
```

- bind : définit la façon dont le serveur est joignable par les utilisateurs
- default_backend : renvoi vers la section suivante
- balance : mode de répartition entre les serveurs Web
- server : décrit chaque serveur Web

Exemple

```
frontend proxypublic  
  bind 10.22.30.210:80  
  default_backend fermeweb  
  
backend fermeweb  
  balance roundrobin  
  server web1 192.168.100.212:80  
  server web2 192.168.100.215:80
```

Options avancées

Le fonctionnement de base, sur l'algorithme roundrobin (ou tourniquet : chacun son tour) peut être optimisé de plusieurs manières **cumulables**.

1. en assurant un **poids** différent selon les capacités et la puissance des serveurs avec l'option **weight**
2. en vérifiant la disponibilité d'un serveur Web avant de lui adresser des demandes avec l'option **check**
3. en utilisant un mode de répartition moins classique que le roundrobin

Affectation d'un poids

```
<code lscript> server web1 10.22.100.212:80 weight 100 server web2 10.22.100.212:80 weight 50  
</code> Le premier serveur recevra 2 fois plus de requêtes que le second
```

Vérification de la disponibilité

```
server web1 10.22.100.212:80 check server web2 10.22.100.212:80 check
```

Modes de répartition

les modes possibles :

- roundrobin : tourniquet passant circulairement vers chacune des serveurs
- leastconn: le serveur sélectionné sera celui ayant précédemment reçu le moins de connexions ;
- source : le serveur est sélectionné en fonction de l'IP source du client ;

- uri : le choix du serveur est fonction du début de l'URI demandée ;
- url_param : le choix du serveur est fonction de paramètres présents dans l'URL demandée ;
- hdr : le choix du serveur est fonction d'un champ présent dans l'en-tête HTTP (Host, User-Agent, ...).

From:

<https://wiki.sio.bts/> - **WIKI SIO : DEPUIS 2017**

Permanent link:

<https://wiki.sio.bts/doku.php?id=haproxy&rev=1677788518>

Last update: **2023/03/02 20:21**

