2025/12/17 10:03 1/4 HA Proxy

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.

Principe

- Plusieurs serveurs possèdent un service et des données identiques (par exemple une application Web) pour constituer un **cluster**
- un **répartiteur** (**HA Proxy**) est connu des utilisateurs (par son IP). Il redirige les demandes vers l'un des serveurs du **cluster**, les prochains échanges se passent ensuite directement entre le client et le serveur désigné
- le **répartiteur distribue la charge** entre les différents serveurs (par exemple chacun son tour, ou plus souvent vers un serveur plus performant)
- le répartiteur peut détecter la panne d'un des serveurs pour éviter de lui envoyer des demandes qui n'aboutiront pas (il assure alors une tolérance de panne)

Mise en place

Pour installer HA Proxy:

apt update
apt install haproxy

Pour vérifier la version installée :

haproxy -v

Juste après l'installation, 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

Last update: 2025/02/12 11:45

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 ajoutera:

```
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_serv2>
```

- bind : définit la façon dont le serveur est joignable par les utilisateurs
- **defautit 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
```

Vérification de la configuration

```
haproxy -c -f /etc/haproxy/haproxy.cfg
```

- -c pour vérifier (check) le fichier
- -f pour spécifier le fichier de configuration

https://wiki.sio.bts/ Printed on 2025/12/17 10:03

2025/12/17 10:03 3/4 HA Proxy

Options avancées

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

- 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

```
server web1 10.22.100.212:80 weight 100 server web2 10.22.100.212:80 weight 50
```

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
```

Chaque serveur sera vérifié avant de lui adresser une requête.

Limitation du nombre d'utilisateurs instantanée

```
server web1 10.22.100.212:80 maxconn 20 server web2 10.22.100.212:80 maxconn 20
```

Chaque serveur sera limité à 20 utilisateurs simultanément

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, ...).

Last update: 2025/02/12 11:45

Sécurisation SSL

Le fichier de configuration est ${\it haproxy.cfg}$. Dans ses options on devra mettre les ports en 443 et ajouter mode tcp :

```
frontend <nomF_E>
  mode tcp
  bind <IP_publiqueHaProxy>:443
  default_backend <nomB_E>

backend <nomB_E>
  balance <modeRepartition>
  mode tcp
  server <nomServ1> <ip_serv1>:443
  server <nomServ2> <ip_serv2>:443
```

Exemple

```
frontend proxypublic
mode tcp
bind 10.22.30.210:443
default_backend fermeweb

backend fermeweb
mode tcp
balance roundrobin
server web1 192.168.100.212:443
server web2 192.168.100.215:443
```

From:

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

Permanent link:

https://wiki.sio.bts/doku.php?id=haproxy&rev=1739360750

Last update: 2025/02/12 11:45



https://wiki.sio.bts/ Printed on 2025/12/17 10:03