

Installation et configuration de Loadbalancer sur debian12

Partie1 : Installation et configuration de HAproxy

1. Après installation de debian12, on met à jour le serveur :

- **sudo apt update**
- **sudo apt upgrade -y**

2. Ensuite on installe le paquet fournissant haproxy : **sudo apt install -y haproxy**

3. Ensuite on confirme l'installation avec : **apt-cache policy haproxy**

```
root@debian:~# apt-cache policy haproxy
haproxy:
  Installé : 2.6.12-1+deb12u1
  Candidat : 2.6.12-1+deb12u1
  Table de version :
  *** 2.6.12-1+deb12u1 500
        500 http://security.debian.org/debian-security bookworm-security/main i
d64 Packages
        100 /var/lib/dpkg/status
        2.6.12-1 500
        500 http://deb.debian.org/debian bookworm/main amd64 Packages
root@debian:~#
```

4. Pour configurer HAProxy en tant qu'équilibreur de charge, nous devons lui indiquer le type de connexions qu'il doit écouter et vers où les connexions doivent être relayées. Cela se fait en éditant un fichier de configuration **/etc/haproxy/haproxy.cfg**. On entre la commande :

sudo vim /etc/haproxy/haproxy.cfg

```
global
    log /dev/log      local0
    log /dev/log      local1 notice
    chroot /var/lib/haproxy
    stats socket /run/haproxy/admin.sock mode 660 level admin
    stats timeout 30s
    user haproxy
    group haproxy
    daemon

    # Default SSL material locations
    ca-base /etc/ssl/certs
    crt-base /etc/ssl/private

    # See: https://ssl-config.mozilla.org/#server=haproxy&server-version=2.0
    .3&config=intermediate
    ssl-default-bind-ciphers ECDHE-ECDSA-AES128-GCM-SHA256:ECDHE-RSA-AES128-
GCM-SHA256:ECDHE-ECDSA-AES256-GCM-SHA384:ECDHE-RSA-AES256-GCM-SHA384:ECDHE-ECDSA
-CHACHA20-POLY1305:ECDHE-RSA-CHACHA20-POLY1305:DHE-RSA-AES128-GCM-SHA256:DHE-RSA
-AES256-GCM-SHA384
    ssl-default-bind-ciphersuites TLS_AES_128_GCM_SHA256:TLS_AES_256_GCM_SHA
```

5. Ajouter ces lignes pour configurer l'interface d'écoute et l'interface backend sur lesquelles la charge du trafic sera équilibrée. Les deux dernières lignes représente les données des deux serveurs web(deux conteneur créés au préalable) qui seront dans notre loadbalancer :

frontend http_front

bind *:80

stats uri /haproxy?stats

default_backend http_back

backend http_back

balance roundrobin

```
server <nom_du_serveur1> <adresseIP_serveur1>:80 check
server <nom_du_serveur2> <adresseIP_serveur2>:80 check
```

6. Nous configurons ensuite l'équilibreur de charge pour qu'il fonctionne sur la couche 7. Ceci peut être utile quand des parties de notre application Web sont situées sur différents hôtes. On peut le faire en conditionnant le transfert de connexion par exemple par l'URL.

backend blog_back

```
server <server name> <private IP>:80 check
server <server name> <private IP>:80 check
```

NB: Le frontend est une règle nommée **url_blog** qui s'applique à toutes les connexions dont les chemins commencent par **/blog**. **use_backend** définit que les connexions correspondant à la **url_blog** condition doivent être servies par le backend nommé **blog_back**, tandis que toutes les autres requêtes sont traitées par le backend par défaut. Côté backend, la configuration configure deux groupes de serveurs, **http_back** comme avant et le nouveau appelé **blog_back** ces serveurs spécifiquement connectés à **example.com/blog**.

7. Après avoir effectué les configurations, enregistrez le fichier et redémarrez HAProxy avec la commande suivante : **sudo systemctl restart haproxy**
NB : avant de redémarrer HAProxy, nous devons nous assurer que la syntaxe de notre fichier de configuration est correcte en exécutant la commande suivante :
sudo haproxy -c -f /etc/haproxy/haproxy.cfg

8. On vérifie ensuite que HAProxy fonctionne correctement : **sudo systemctl status haproxy**

Partie 2 : Configuration de SSL

1. Tous d'abord il faut une CSR, et on commence par installer **OpenSSL**, un outil open-source pour la gestion des certificats SSL. On utilise la commande suivante :
sudo apt-get install openssl
2. Générer une clé privée en utilisant la commande suivante :
sudo openssl genrsa -out nomdedomaine.key 2048
sudo chmod 600 nomdedomaine.key

NB: Attention, il faut garder la clé et la protéger pour éviter que quelqu'un puisse avoir accès et la supprimer. Si elle se perd ou elle est compromise il faudra probablement refaire le certificat SSL.

3. Ensuite on utilise la clé précédemment générée pour créer la CSR avec la commande suivante : **sudo openssl req -new -key nomdedomaine.key -out nomdedomaine.csr**
Ensuite le terminal va demander des informations à rentrer pour le certificat :
 - **Common Name (CN)** : Le nom commun représente le nom de domaine (FQDN) pour lequel le certificat est demandé. Par exemple, "www.example.com" ou "example.com".

- **Organization (O)** : Le nom officiel de votre organisation, tel qu'enregistré légalement. Par exemple, "Example Corp."
- **Organizational Unit (OU)** : Il s'agit de la division ou du département au sein de votre organisation qui gère le certificat SSL/TLS. Par exemple, "IT Department" ou "Web Security."
- **Locality (L)** : Le nom de la ville où votre organisation est située. Par exemple, "Marseille."
- **State or Province (ST)** : Le nom complet de l'État ou de la province dans lequel se trouve votre organisation. Par exemple, "California" ou "Île-de-France."
- **Country (C)** : Le code à deux lettres représentant le pays dans lequel votre organisation est basée. Par exemple, "US" pour les États-Unis ou "FR" pour la France.
- **Email Address (EA)** : Une adresse e-mail de contact pour les questions liées au certificat.
- **Public Key** : La clé publique qui sera incluse dans le certificat SSL/TLS. Généralement, cette clé est dérivée de la clé privée que vous avez créée avant de générer la CSR.

```
Country Name (2 letter code) [AU]:FR
State or Province Name (full name) [Some-State]:France
Locality Name (eg, city) []:Montévrain
Organization Name (eg, company) [Internet Widgits Pty Ltd]:Sukuku
Organizational Unit Name (eg, section) []:Administration Systems Networks
Common Name (e.g. server FQDN or YOUR name) []:
Email Address []:k.koffi@ecole-ipssi.net
```