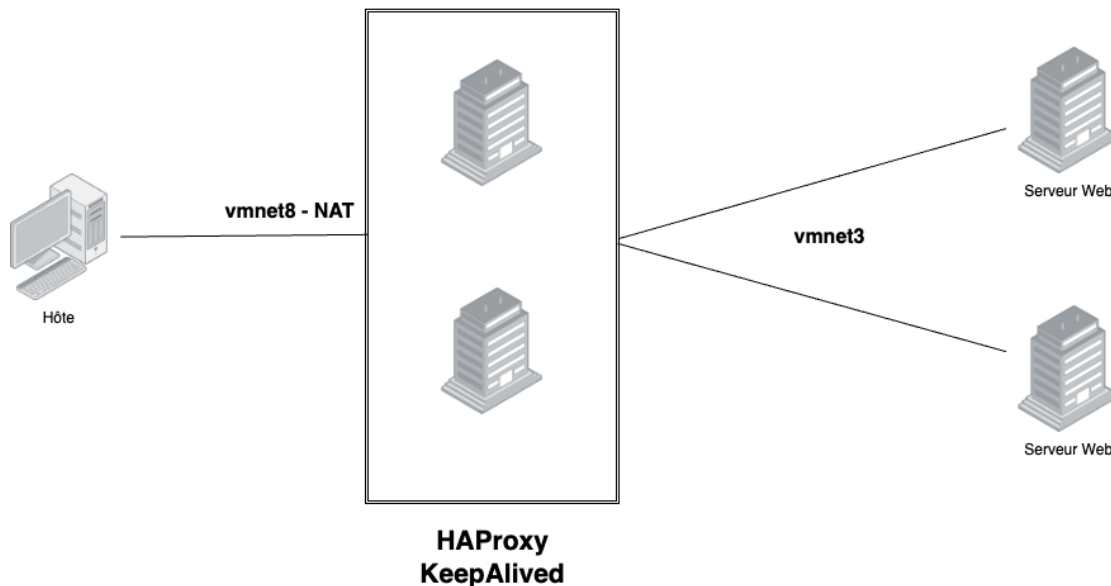


Mise en place de HA Proxy

Présentation

HA Proxy est un load-balancing (<http://www.haproxy.org/>).

Nous allons le mettre en place pour répartir la charge sur 2 serveurs web



Les termes

Proxy

Un proxy est un intermédiaire (mandataire) qui se place entre la demande du client et la station finale.

Load-balancing

Répartition de charge suivant des critères déterminés.

Front-end

Le front-end est composé de serveurs qui reçoivent les connexions en premier, ici ce sera le load-balancer.

Back-end

Ce sont les serveurs qui vont recevoir le trafic en second, ici les serveurs web.

Mise en place

Installation des serveurs web

Installez 2 serveurs web : Apache ou nginx.

Prenons pour exemple les serveurs web d'adresses IP 192.168.1.1 et 192.168.1.2

Installation et configuration du load balancer

Sous Debian

Commande : `apt install haproxy`

Configuration

Le fichier `/etc/haproxy/haproxy.cfg` contient la configuration de HA Proxy. Il est divisé en 4 parties.

Section global

Définit la sécurité et les performances de HAProxy

Section Defaults

Paramètres appliqués aux sections frontend et backend.

Le mode est http pour la transmission des requêtes HTTP.

Section frontend

Définit les paramètres de HAProxy en tant que Reverse Proxy : adresses ip, ports, ...

Section backend

Définit les serveurs de destination (ici les serveurs web) : type de répartition de charge, statistiques, ...

Exemple de configuration

Extrait du fichier haproxy.cfg :

```
# Frontend section
frontend webserver_front
    # Ecoute sur le port 80, toutes les adresses ip
    bind *:80
    # Backend lié
    default_backend webservers_backend
    # Enable send X-Forwarded-For header : envoie de l'ip client et non
    # du proxy
    option forwardfor

# Backend section
backend webservers_backend
    # Test pour roundrobin
    balance roundrobin
    # Voir les autres modes pour balance

    # Vérification de l'état http : A FAIRE

    # Liste des serveurs web
    server backend13 192.168.1.1:80 check
    server backend14 192.168.1.2:80 check
```

Tests de la configuration

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

Les ACL

Les ACLs dans HAProxy permettent de réaliser des tests conditionnels lors du traitement des requêtes.

Exemple de configuration :

```
acl host_theme hdr(host) -i theme.monsite.com
acl url_theme path_beg -i /theme
use_backend theme-backend if host_theme url_theme
```

Dans cet exemple, deux ACLs sont définies pour identifier les requêtes destinées à un thème spécifique et si les deux conditions sont remplies, la requête est routée vers theme-backend.

Répartition de charge basée sur l'URL

La répartition va s'appuyer sur l'utilisation des acl.

Exemple de configuration :

```
frontend http-in
    bind *:80
    acl is_api path_beg -i /api
    use_backend api-backend if is_api
    default_backend web-backend
```

Dans cet exemple, toutes les requêtes commençant par api seront redirigées vers le backend api-backend ; toutes les autres requêtes iront vers web-backend.

Health Checks

Pour garantir que le trafic est seulement envoyé aux serveurs qui fonctionnent correctement, HAProxy peut être configuré pour effectuer des contrôles d'accès spécifiques.

Exemple de configuration :

```
backend app-backend
    option httpchk GET /health
    http-check expect status 200
    server app1 192.168.1.10:80 check
    server app2 192.168.1.11:80 check
```

Dans cet exemple, le serveur est déclaré viable si l'url /health renvoie une réponse http OK.

Limitation du taux de requêtes

Le rate limiting est une technique efficace pour se protéger contre les attaques de type DDoS (Distributed Denial of Service).

Exemple de configuration :

```
frontend http-in
    bind *:80
    stick-table type ip size 100k expire 30s store http_req_rate(10s)
    http-request track-sc0 src
    http-request deny deny_status 429 if { sc_http_req_rate(0) gt 20 }
```

Dans cet exemple, HAProxy utilise une table de suivi pour limiter les clients à 10 requêtes HTTP toutes les 10 secondes. Si le taux est dépassé, la connexion est rejetée.

Voir [Traffic policing | HAProxy config tutorials](#)

Activation du monitoring

HAProxy fournit une interface de statistiques accessible via une page web, qui présente une vue en temps réel des serveurs, des services et des performances.

Exemple de configuration :

```
listen stats
    bind *:8080
    stats enable
    stats uri /stats
    stats hide-version
    stats auth admin:password
```

Dans cet exemple, l'accès aux es statistiques est configuré sur le port 8080, accessible via l'URI /stats. Il est protégé par une authentification basique où admin est le nom d'utilisateur et password le mot de passe.

Liens

<https://alexandre-dosreis.me/blog/infrastructure/haproxy-sous-debian-10-butler/>

<https://notamax.be/debian-10-creer-un-proxy-load-balancer-haproxy-http/>

<https://le-guide-du-secops.fr/2020/08/06/mise-en-place-dun-haproxy-repartiteur-de-charge/>

<https://upcloud.com/resources/tutorials/haproxy-load-balancer-debian>

<https://cbonte.github.io/haproxy-dconv/1.7/configuration.html>

<https://www.haproxy.com/blog/the-four-essential-sections-of-an-haproxy-configuration/>

<https://www.haproxy.com/blog/dns-service-discovery-haproxy/>