

HAProxy

HAProxy, ou High Availability Proxy, est un logiciel libre qui permet de répartir la charge du trafic réseau entre plusieurs serveurs. Il est principalement utilisé pour améliorer les performances, la disponibilité et la fiabilité des applications web. En agissant comme un équilibreur de charge (load balancer), HAProxy distribue les demandes des clients aux serveurs en arrière-plan de manière équilibrée, assurant ainsi que chaque serveur reçoit une part équitable de la charge de travail. Cela aide à éviter la surcharge de certains serveurs, améliore le temps de réponse des applications et offre une meilleure tolérance aux pannes.

- [Guide de Configuration HAProxy](#)

Guide de Configuration

HAProxy

Ce guide explique comment configurer HAProxy pour différents types de services : RDP, Minecraft et sites Internet. Chaque section contient une explication étape par étape pour permettre à quiconque de configurer HAProxy avec ses propres liens et adresses.

Configuration Générale

Fichier de Configuration

Commencez par créer ou éditer le fichier de configuration HAProxy situé généralement à `/etc/haproxy/haproxy.cfg`.

commande linux pour éditer la config :

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

Voici la configuration générale :

```
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
```

```
# SSL configurations
```

```
ssl-default-bind-ciphers ECDHE-ECDSA-AES128-GCM-SHA256:ECDHE-RSA-AES128-GCM-SHA256
```

```
ssl-default-bind-ciphersuites TLS_AES_128_GCM_SHA256:TLS_AES_256_GCM_SHA384
```

```
ssl-default-bind-options ssl-min-ver TLSv1.2 no-tls-tickets
```

```
defaults
```

```
log global
```

```
mode tcp
```

```
option tcplog
```

```
option dontlognull
```

```
timeout connect 5000
```

```
timeout client 50000
```

```
timeout server 50000
```

```
errorfile 400 /etc/haproxy/errors/400.http
```

```
errorfile 403 /etc/haproxy/errors/403.http
```

```
errorfile 408 /etc/haproxy/errors/408.http
```

```
errorfile 500 /etc/haproxy/errors/500.http
```

```
errorfile 502 /etc/haproxy/errors/502.http
```

```
errorfile 503 /etc/haproxy/errors/503.http
```

```
errorfile 504 /etc/haproxy/errors/504.http
```

```
listen stats
```

```
bind :9000
```

```
mode http
```

```
stats enable
```

```
stats hide-version
```

```
stats uri /
```

```
stats auth admin:admin
```

Configuration des Services

Frontend et Backend

Pourquoi un Frontend ?

Le **frontend** dans HAProxy est responsable de l'écoute des connexions entrantes des clients. Il définit quels ports et quelles adresses IP sont surveillés par HAProxy pour recevoir les demandes de connexion. Le frontend peut également effectuer certaines actions de filtrage ou de routage initial avant de transférer la demande à l'un des backends appropriés.

Pourquoi un Backend ?

Le **backend** dans HAProxy est responsable de la gestion des connexions sortantes vers les serveurs réels. Chaque backend contient une ou plusieurs directives `server` qui définissent les serveurs vers lesquels les connexions doivent être dirigées. Le backend inclut également des paramètres pour surveiller la santé des serveurs et d'autres configurations spécifiques à la gestion des connexions.

Comment Modifier les Liens et Adresses ?

Pour changer les liens ou les adresses IP, vous devrez modifier les entrées correspondantes dans les sections frontend et backend. Par exemple, si vous avez un nouveau site ou serveur, vous ajouterez ou modifierez les directives de `use_backend` et `server` en conséquence.

1. Configuration RDP

Pour configurer HAProxy pour les services RDP (Remote Desktop Protocol), suivez ces étapes :

Frontend

Définissez un frontend qui écoute sur les ports RDP :

```
frontend rdp_frontend
    bind *:3389
    bind *:3390
    bind *:3391
    mode tcp
    tcp-request inspect-delay 5s
    tcp-request content accept if RDP_COOKIE

    use_backend rdp_backend_qoyri if { dst_port 3389 }
    use_backend rdp_backend_refu if { dst_port 3390 }
    use_backend rdp_backend_bawikx if { dst_port 3391 }
```

Pour changer les ports RDP ou ajouter un nouveau, modifiez ou ajoutez une nouvelle ligne `bind` et une nouvelle directive `use_backend`.

Backend

Définissez les backends pour chaque port RDP :

```
backend rdp_backend_qoyri
    mode tcp
    server qoyri_server 192.168.0.5:3389 check

backend rdp_backend_refu
    mode tcp
    server refu_server 192.168.0.52:3389 check

backend rdp_backend_bawikx
    mode tcp
    server bawikx_server 192.168.0.169:3389 check
```

Pour changer les adresses des serveurs RDP, modifiez les lignes `server` en conséquence.

2. Configuration Minecraft

Pour configurer HAProxy pour les serveurs Minecraft, suivez ces étapes :

Frontend

Définissez un frontend qui écoute sur le port Minecraft par défaut (25565) :

```
frontend minecraft
    mode tcp
    bind 0.0.0.0:25565
    tcp-request inspect-delay 5s

    acl mc_feiti_flg payload(5,15) -m sub feiti.qoyri.fr
    tcp-request content accept if mc_feiti_flg
    use_backend mc_feiti_app if mc_feiti_flg

    acl mc_bawikx_flg payload(5,13) -m sub mc.bawikx.fr
```

```
tcp-request content accept if mc_bawikx_flg
use_backend mc_bawikx_app if mc_bawikx_flg

acl mc_refugier_flg payload(5,17) -m sub refugier.qoyri.fr
tcp-request content accept if mc_refugier_flg
use_backend mc_refugier_app if mc_refugier_flg

acl mc_garabou_flg payload(5,17) -m sub garabou.qoyri.fr
tcp-request content accept if mc_garabou_flg
use_backend mc_garabou_app if mc_garabou_flg
```

Pour ajouter ou modifier des serveurs Minecraft, ajustez les lignes `acl`, `tcp-request content accept`, et `use_backend`.

Backend

Définissez les backends pour chaque serveur Minecraft :

```
backend mc_feiti_app
server mc_feiti_srv 192.168.0.52:25566 check

backend mc_bawikx_app
server mc_bawikx_srv 192.168.0.110:25565 check

backend mc_refugier_app
server mc_refugier_srv 192.168.0.52:25565 check

backend mc_garabou_app
server mc_garabou_srv 192.168.0.52:25610 check
```

Pour changer les adresses des serveurs Minecraft, modifiez les lignes `server`.

3. Configuration des Sites Internet

Pour configurer HAProxy pour les sites web via SSL, suivez ces étapes :

Frontend

Définissez un frontend qui écoute sur le port HTTPS (443) :

```
frontend https_frontend
  bind *:443
  mode tcp
  option tcplog
  tcp-request inspect-delay 5s
  tcp-request content accept if { req_ssl_hello_type 1 }
  use_backend ssl_archive_backend if { req_ssl_sni -i archive.qoyri.fr }
  use_backend ssl_cloud_backend if { req_ssl_sni -i cloud.qoyri.fr }
  use_backend ssl_ftp_backend if { req_ssl_sni -i ftp.qoyri.fr }
  use_backend ssl_gprojet_backend if { req_ssl_sni -i gprojet.qoyri.fr }
```

Pour ajouter ou modifier des sites web, ajustez les lignes `use_backend` et `req_ssl_sni`.

Backend

Définissez les backends pour chaque site :

```
backend ssl_archive_backend
  mode tcp
  server server1 192.168.0.52:443 check

backend ssl_cloud_backend
  mode tcp
  server server2 192.168.0.52:443 check

backend ssl_ftp_backend
  mode tcp
  server server3 192.168.0.52:443 check

backend ssl_gprojet_backend
  mode tcp
  server server3 192.168.0.52:443 check
```

Pour changer les adresses des serveurs web, modifiez les lignes `server`.

Notes Importantes

- **Redémarrage HAProxy** : Après avoir modifié la configuration, redémarrez HAProxy pour appliquer les changements.

```
sudo systemctl restart haproxy
```

- **Vérification de la Configuration** : Vous pouvez vérifier la syntaxe de la configuration avant de redémarrer HAProxy avec la commande suivante :

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

- **Sécurité** : Assurez-vous de sécuriser l'interface de statistiques en utilisant des mots de passe forts et en limitant l'accès par adresse IP si nécessaire.

En suivant ces étapes, vous devriez être capable de configurer HAProxy pour différents types de services et gérer efficacement la répartition de la charge et la disponibilité de vos applications.