Sadek

Adel

Sio2

Installation Haproxy/keepalived

Ha proxy permet d'assurer la répartition des charges entre deux services / machines

Je vais créer une machine virtuelle qui sera mis dans un cluster avec mon serveur reverse proxy

Les deux machines seront dans le VLAN40 (DMZ) et seront sous debian 11
IP de la première machine : 172.18.0.8
IP de la seconde machine : 172.18.0.9
IP virtuelle qui sera utiliser : 172.18.0.10

Dans le fichier de configuration de HAproxy :

La directive frontend <Nom> sert à définir l'ip virtuelle du cluster

La directive backend <Nom> sert à définir quel seront les machines intégrer au cluster

Je commence par installer le paquet haproxy

Le fichier de configuration : /etc/haproxy/haproxy.cfg

Je vais dans le fichier de configuration et je rajoute ceci à la fin

Comme sa sera un cluster entre 2 reverse proxy il sera en mode http

Je vais d'abord essayer haproxy entre 2 serveurs web

```
#Je définis l'ip du HAproxy et son port d'ecoute
frontend LOADBALANCER-01
        bind 172.18.0.8:85
#En fonctionnera en mode tcp pas en mode http
#       mode http
#Je definis cette conf sera valide pour quel groupe de machine
        default_backend SERVER-01


#Ensuite je définis les machines qui seront dans le groupe SERVER-01

backend SERVER-01
        balance roundrobin
        server num1 172.18.0.8:80 check
        server num2 172.18.0.9:80 check
```

J'ai commenté le mode http il ne sert pas à grand-chose

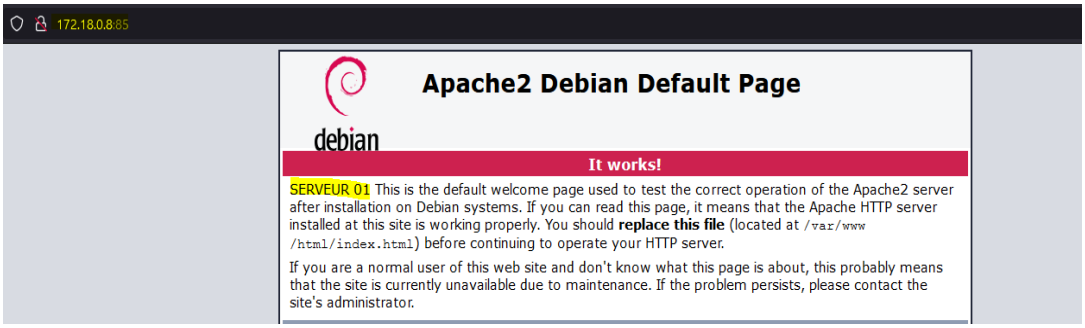
Je définis un nom pour chaque machine (num1 et num2) un peu comme les nœuds sur heartbeat et je définis leur ip et le port d'écoute du service que je veux intégrer au cluster et je rajoute la directive check pour savoir si le service est toujours en vie

Le cluster fonctionnera en mode roundrobin il n'y aura pas de priorité dans les réponses aux requêtes sa sera chacun son tour.

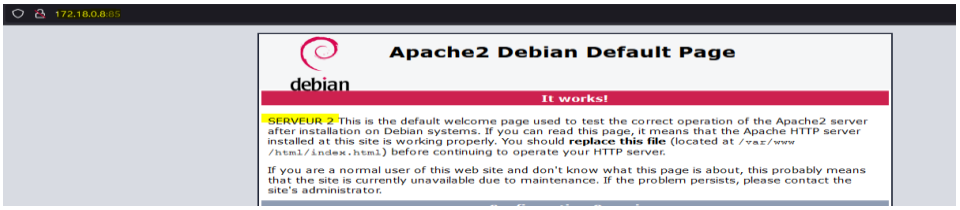
Je fais un test dans ma barre de recherche je tape ceci

<ipHAproxy> :<port>

Les 20 premières requêtes tombent sur le premier serveur web



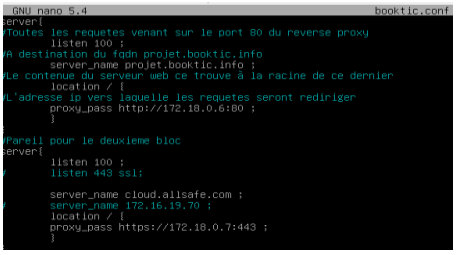
Et après les autres requêtes atterrissent sur le deuxième serveur web le cluster fonctionne correctement



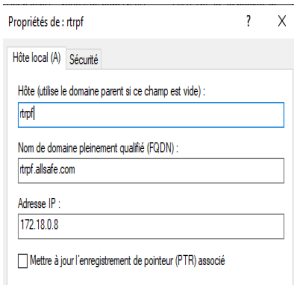
Je passe maintenant au reverse proxy nginx qui écoutera maintenant sur le port 100 et haproxy restera sur le port 80

J'installe nginx sur la seconde machine et je copie via scp ma conf reverse proxy de la première machine qui ressemble à sa

Mon fichier de conf nginx ressemble à sa



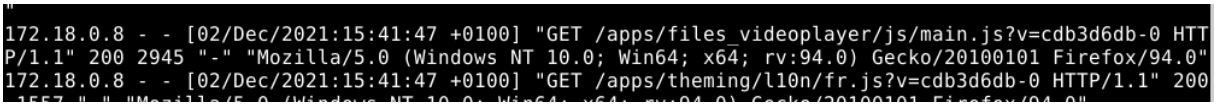
Sur mon serveur DNS je modifie l'enregistrement du serveur reverse proxy pour mettre l'ip du serveur principal qui héberge le service HAproxy qui permet le clustering



Ensuite je me connecte sur ce fqdn cloud.allsafe.com

Je réitère mes requêtes jusqu'à ce que ma requête soit renvoyer vers le second serveur 172.18.0.9

Et je vais regarder les log nginx pour voir si l'ip du HAproxy envoie tes requêtes http vers le serveur reverse proxy et ce reverse proxy renvoie vers le serveur web qui héberge nextcloud



On voit bien dans les logs du serveur nginx l'ip 172.18.0.8 qui est celle du HAproxy

Nous sommes face à un souci si le serveur hébergeant HAproxy tombe tout derrière tombe, il faut une solution de redondance tel VRRP qui nous permettra d'avoir une ip virtuelle celui qui la gèrera sera le serveur MASTER si ce dernier tombe le SLAVE prend le relais.

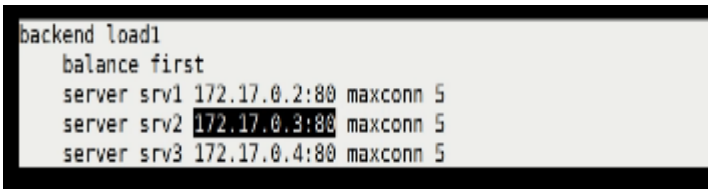
Pour ceci nous allons utiliser le paquet keppalived

(pour détail installation nginx voir la doc dessus)

D’autre algorithme pour répartir les charges

Sur HAproxy il y’a l’algorithme leastconn il suffit de remplacer « roundrobin » par « leastconn » cet algorithme prend en compte le nombre de connexions si un serveur gère déjà un nombre élevé de connexions HAproxy ne lui renverra pas la requête

L’algorithme first connexion pour définir un nombre maximal de requêtes que le serveur va gérer par exemple maximum 5 voici un bon exemple pour illustrer cet algorithme



Avec le paramètre « maxconn » je peux définir le nombre maximal de requêtes

MISE EN PLACE KEEPALIVED

Voici le fichier de configuration de keepalived sur le serveur MASTER

Il n'y a pas de fichier de conf dans « /etc/keepalived » il faut que je crée le fichier keepalived.conf dans ce repertoire

```
#Je définis le nom de l'interface virtuelle
vrrp_instance VI_1 {
        #Je définis si ce serveur est master ou slave
        state MASTER
        #L'interface virtuelle ce basera sur cette interface phisique
        interface eth0
        #L'identifiant groupe
        virtual_router_id 1
        #Sa priorité c'est comme HSRP celui qui a la priorité la plus haute sera définis comme master
        priority 100
        #L'adresse ip virtuelle son masque et l'adresse de diffusion du réseau
        virtual_ipaddress {
        172.18.0.10/28 brd 172.18.0.31 dev eth0
        }
}
```

Je sauvegarde et redémarre le service je tape la commande « ip a »

L'ip virtuelle s'affiche

```
2: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc mq state UP group default qlen 1000
    link/ether 00:15:5d:13:25:05 brd ff:ff:ff:ff:ff:ff
    inet 172.18.0.8/27 brd 172.18.0.31 scope global eth0
       valid_lft forever preferred_lft forever
    inet 172.18.0.10/28 brd 172.18.0.31 scope global eth0
       valid_lft forever preferred_lft forever
    inet6 fe80::215:5dff:fe13:2505/64 scope link
       valid_lft forever preferred_lft forever
root@ReverseeProxy:/etc/nginx/sites-enabled#
```

Je test un ping vers cette ip

```
root@ReverseeProxy:/etc/nginx/sites-enabled# ping 172.18.0.10
PING 172.18.0.10 (172.18.0.10) 56(84) bytes of data.
64 bytes from 172.18.0.10: icmp_seq=1 ttl=64 time=0.047 ms
64 bytes from 172.18.0.10: icmp_seq=2 ttl=64 time=0.038 ms
64 bytes from 172.18.0.10: icmp_seq=3 ttl=64 time=0.048 ms
64 bytes from 172.18.0.10: icmp_seq=4 ttl=64 time=0.026 ms
64 bytes from 172.18.0.10: icmp_seq=5 ttl=64 time=0.062 ms
```

Le ping passe je transfère maintenant le fichier de conf vers le serveur SLAVE et je modifierai simplement MASTER par BACKUP

```
#Je définis le nom de l'interface virtuelle
vrrp_instance VI_1 {
        #Je définis si ce serveur est master ou sla
        state BACKUP
        #L'interface virtuelle ce basera sur cette
        interface eth0
        #L'identifiant groupe
        virtual_router_id 1
        #Sa priorité c'est comme HSRP celui qui a l
        priority 100
        #L'adresse ip virtuelle son masque et l'adr
        virtual_ipaddress {
        172.18.0.10/28 brd 172.18.0.31 dev eth0
        }
}
```

Je désactive la connexion sur le serveur master avec service networking stop

Je tape la commande « ip a » sur le serveur slave et je regarde s'il a pris l'ip virtuelle

```
2: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 15
    link/ether 00:15:5d:13:25:0f brd ff:ff:ff:ff:
    inet 172.18.0.9/28 brd 172.18.0.15 scope globa
       valid_lft forever preferred_lft forever
    inet 172.18.0.10/28 brd 172.18.0.31 scope glo
       valid_lft forever preferred_lft forever
```

Il l'a bien pris

Ensuite je prends un client Windows appart du contexte pour voir si il ping l'IP virtuelle quand le master tombe pour voir si le slave reprend le relais

```
PS C:\Users\Administrateur> ping 172.18.0.10

Envoi d'une requête 'Ping'  172.18.0.10 avec 32 octets de données :
Réponse de 172.18.0.10 : octets=32 temps=3 ms TTL=62
Réponse de 172.18.0.10 : octets=32 temps=2 ms TTL=62
Réponse de 172.18.0.10 : octets=32 temps=2 ms TTL=62
Réponse de 172.18.0.10 : octets=32 temps=2 ms TTL=62
```

Sa fonctionne parfaitement

Je prend la config du HAproxy et la transfère via scp sur le serveur slave et dans l'enregistrement DNS qui point vers le reverse proxy je mets une étoile « * » pour qu'il écoute sur toute ses ip comme ça dès qu'il aura l'ip virtuelle il écoutera dessus dans la directive frontend comme ici

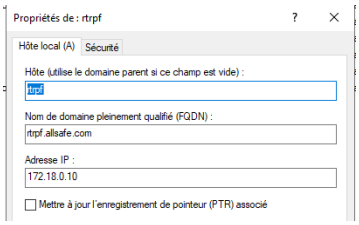
```
#Je définis l'ip du HAproxy et son port d'ecoute
frontend LOADBALANCER-01
        bind *:80
#En fonctionnera en mode tcp pas en mode http
#Je définis cette conf sera valide pour quel groupe de machine
        default_backend SERVER-01
```

Je transfère cette config sur le serveur slave

#Sur nginx bien changer le port d'ecoute dans /etc/ngins/site-enabled/default

Je redémarre le service nginx et haproxy sur le serveur slave

Je modifie l'enregistrement DNS pour y mettre l'IP virtuelle

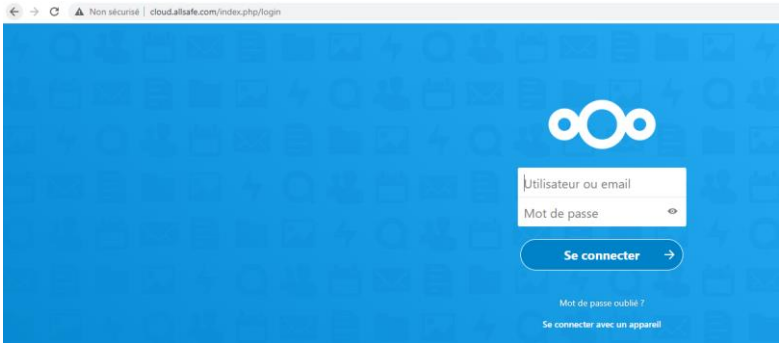


Maintenant je fais un test

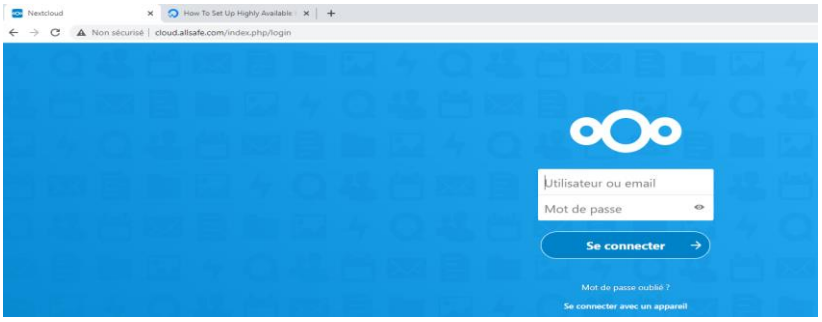
Je fais sur cloud.allsafe.com

Normalement la première requête va vers le serveur haproxy 172.18.0.8 qui gérera

Sa fonctionne



Et lorsque je coupe le service et que je réactualise la page sa fonctionne aussi



Sa fonctionne

Le seul souci c'est que quelquefois lorsque le serveur master tombe haproxy ne répond qu'après un redémarrage du service je fais donc un script cron qui vérifiera sur la machine slave si elle à l'IP virtuel et qui redémarrera le service lorsque cette condition est remplie

Voici le contenue du script

```
#!/bin/sh
a=$(ip a | grep "inet 172.18.0.10/28 brd 172.18.0.31 scope global secondary eth0")
if [ "$a" = "    inet 172.18.0.10/28 brd 172.18.0.31 scope global secondary eth0" ]
then
service haproxy restart


else
fi
```

Je fais en sorte que le script s'exécute toute les une minute

```
# m h  dom mon dow   command
 1 * * * * sh /etc/haproxy/haproxy.sh
```

Ensuite après un test sa fonctionne parfaitement après avoir éteint la connexion sur le serveur master j'ai attendu 10,15 secondes et sa a fonctionner sa a recharger nextcloud

Cluster sur d'autres service tel ssh/sftp

Pour ceci il va falloir modifier le mode http en mode tcp dans les paramètres par défaut

Ici

```
defaults
        log     global
        mode    tcp
        option  httplog
        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
```

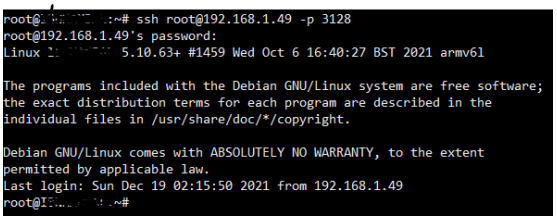
Je fais en sorte que mon serveur écoute sur le port 3128 et renvoie vers le port 22 de la même machine (je suis sur un autre réseau pour ce test)

```
frontend LOADBALANCER-01
        bind 192.168.1.49:3128
        default_backend SERVER-01

backend SERVER-01
        balance roundrobin
        server num1 192.168.1.49:22
```

Je vais un test

Sa fonctionne parfaitement



HAproxy sur une machine séparer de nginx avec ssl et redirection

Pour éviter un trop grand nombre de redirection il faut que la mise en place du ssl ce fasse que sur le serveur HAproxy et que ni sur le serveur nginx ou apache SSL soit mis en place

D'abord sur le serveur HAproxy créer un certificat + une clé privée

Ensuite faire écouter notre serveur HAproxy sur le port 443 en mode ssl avec le chemin du certificat

Toujours faire écouter HAproxy sur le port 80

Ensuite il faut que je force ma redirection de http vers https avec cette directive

http-request redirect scheme https unless { ssl_fc }

La redirection se fait donc au niveau du HAproxy et non plus du serveur NGINX

```
#Je définis l'ip du HAproxy et son port d'ecoute
frontend LOADBALANCER-01
        bind *:80
        bind *:443 ssl crt /etc/ssl/haproxy.pem
#       bind *:443
        mode http
        #Forcer redirection http vers https
        http-request redirect scheme https unless { ssl_fc }
```

Pour éviter une erreur renommer la clé privée comme ceci haproxy.pem.key car HAproxy va chercher automatiquement la clé privée

Ensuite maintenant que mon serveur HAproxy est sur une machine séparer de mon nginx je fais écouter mon nginx sur le port 80 et je redirige les requêtes sur mon serveur nginx vers le port 80 et non plus le 100

NGINX :

```
server{
        listen 80 ;
#A destination du fqdn glpi.booktic.info
        server_name glpi.booktic.info ;
#Le contenue du serveur web ce trouve à la racine de ce dernier
        location / {
        proxy_pass http://172.17.1.82:443 ;
        }
}
```

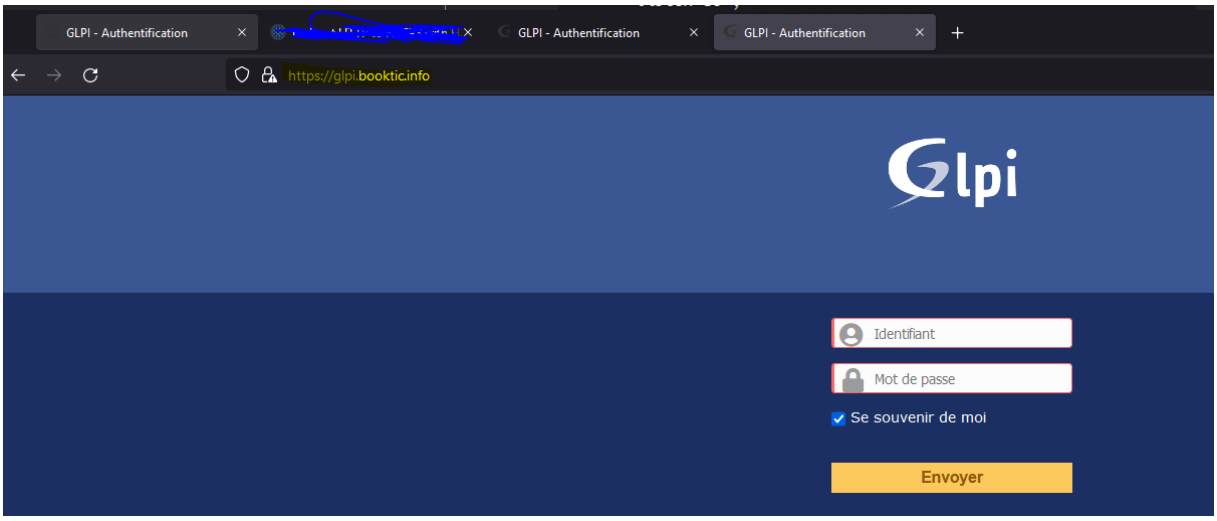
Voila ensuite je redirige en http vers le port 443 sur mon serveur apache2 et sur mon serveur apache je désactive le module SSL

Ensuite le résultat

J'ai saisi ceci dans ma barre de recherche

http://glpi.booktic.info:80

On voit bien que je suis automatiquement redirigée en https



Haproxy avec openvpn

On a vu qu'il y'a deux modes pour HAproxy tcp et http je vais essayer de mettre le mode par tcp par défaut sur une nouvelle machine HAproxy

Qui sera aussi dans le vlan 40 et son IP sera 172.18.0.5

Le fichier de conf HAproxy ressemble à sa

```
defaults
        log     global
        mode    tcp
        option  httplog
        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

#Je définis l'ip du HAproxy et son port d'ecoute
frontend LOADBALANCER-01
        bind *:1195

#En fonctionnera en mode tcp pas en mode http
#Je definis cette conf sera valide pour quel groupe de machine
        default_backend OPENVPN-01


#Ensuite je définis les machines qui seront dans le groupe SERVER-01

backend OPENVPN-01
        balance roundrobin
        server num1 172.19.0.6:1195 check
        mode    tcp
```

Le serveur écoute sur le port 1195 et renvoie sur mon serveur openvpn qui écoute sur le port 1195

Ensuite il y'a une ligne à supprimer dans mon fichier de conf de serveur c'est celle-ci car elle ne peut être utiliser qu'avec « udp »

```
explicit-exit-notify 1
```

Ensuite il faut que je modifie le protocole pour mettre UDP sur le fichier de conf de mon serveur

```
proto tcp
dev tun
port 1195
ca /secondeConf/easy-rsa/pki/ca.crt
cert /secondeConf/easy-rsa/pki/issued/secondeSRV.crt
key /secondeConf/easy-rsa/pki/private/secondeSRV.key
dh /secondeConf/easy-rsa/pki/dh.pem

server 10.1.0.0 255.255.255.0
push "redirect-gateway def1"
#push "dhcp-option DNS 172.17.1.8"
push "dhcp-option DNS 8.8.8.8"

client-to-client

keepalive 10 120
persist-key
persist-tun

cipher AES-256-CBC
compress lz4-v2
verb 5
```

J’ai modifié le protocole de udp à tcp

Puis sur le fichier de conf mon client il faut que je modifie le protocole de udp à tcp aussi

```
client
dev tun
proto tcp
remote 172.18.0.5 1195

ca ca.crt
cert secondeCLT.crt
key secondeCLT.key
```

Résultat

Ensuite je teste la connexion en regardant les logs de mon serveur openvpn

Sa marche parfaitement

```
Apr 13 16:01:55 OPENVPN ovpn-second[2991978]: TCP connection established with [AF_INET]172.18.0.5:35378
Apr 13 16:01:55 OPENVPN ovpn-second[2991978]: TCPv4_SERVER link local: (not bound)
Apr 13 16:01:55 OPENVPN ovpn-second[2991978]: TCPv4_SERVER link remote: [AF_INET]172.18.0.5:35378
Apr 13 16:01:55 OPENVPN ovpn-second[2991978]: 172.18.0.5:35378 TLS: Initial packet from [AF_INET]172.18.0.5:35378, sid=4ae559d6 ad39db9c
Apr 13 16:01:55 OPENVPN ovpn-second[2991978]: 172.18.0.5:35378 VERIFY OK: depth=1, CN=FR
Apr 13 16:01:55 OPENVPN ovpn-second[2991978]: 172.18.0.5:35378 VERIFY OK: depth=0, CN=secondeCLT
Apr 13 16:01:55 OPENVPN ovpn-second[2991978]: 172.18.0.5:35378 peer info: IV_VER=3.git::d3f8b18b
Apr 13 16:01:55 OPENVPN ovpn-second[2991978]: 172.18.0.5:35378 peer info: IV_PLAT=win
Apr 13 16:01:55 OPENVPN ovpn-second[2991978]: 172.18.0.5:35378 peer info: IV_NCP=2
Apr 13 16:01:55 OPENVPN ovpn-second[2991978]: 172.18.0.5:35378 peer info: IV_TCPNL=1
Apr 13 16:01:55 OPENVPN ovpn-second[2991978]: 172.18.0.5:35378 peer info: IV_PROTO=30
Apr 13 16:01:55 OPENVPN ovpn-second[2991978]: 172.18.0.5:35378 peer info: IV_CIPHERS=AES-256-GCM:AES-128-GCM:CHACHA20-POLY1305:BF-CBC
Apr 13 16:01:55 OPENVPN ovpn-second[2991978]: 172.18.0.5:35378 peer info: IV_AUTO_SESS=1
Apr 13 16:01:55 OPENVPN ovpn-second[2991978]: 172.18.0.5:35378 peer info: IV_GUI_VER=OCWindows_3.3.3-2562
Apr 13 16:01:55 OPENVPN ovpn-second[2991978]: 172.18.0.5:35378 peer info: IV_SSO=webauth,openurl,crtext
Apr 13 16:01:55 OPENVPN ovpn-second[2991978]: 172.18.0.5:35378 peer info: IV_BS64DL=1
Apr 13 16:01:55 OPENVPN ovpn-second[2991978]: 172.18.0.5:35378 WARNING: 'link-mtu' is used inconsistently, local='link-mtu 1560', remote='link-mtu 1543'
Apr 13 16:01:55 OPENVPN ovpn-second[2991978]: 172.18.0.5:35378 WARNING: 'comp-lzo' is present in local config but missing in remote config, local='comp-lzo'
Apr 13 16:01:55 OPENVPN ovpn-second[2991978]: 172.18.0.5:35378 WARNING: 'keysize' is used inconsistently, local='keysize 256', remote='keysize 128'
Apr 13 16:01:55 OPENVPN ovpn-second[2991978]: 172.18.0.5:35378 Control Channel: TLSv1.3, cipher TLSv1.3 TLS_AES_256_GCM_SHA384, 2048 bit RSA
Apr 13 16:01:55 OPENVPN ovpn-second[2991978]: 172.18.0.5:35378 [secondeCLT] Peer Connection Initiated with [AF_INET]172.18.0.5:35378
Apr 13 16:01:55 OPENVPN ovpn-second[2991978]: secondeCLT/172.18.0.5:35378 MULTI_sva: pool returned IPv4=10.1.0.6, IPv6=(Not enabled)
Apr 13 16:01:55 OPENVPN ovpn-second[2991978]: secondeCLT/172.18.0.5:35378 MULTI: Learn: 10.1.0.6 -> secondeCLT/172.18.0.5:35378
Apr 13 16:01:55 OPENVPN ovpn-second[2991978]: secondeCLT/172.18.0.5:35378 MULTI: primary virtual IP for secondeCLT/172.18.0.5:35378: 10.1.0.6
Apr 13 16:01:55 OPENVPN ovpn-second[2991978]: secondeCLT/172.18.0.5:35378 Data Channel: using negotiated cipher 'AES-256-GCM'
Apr 13 16:01:55 OPENVPN ovpn-second[2991978]: secondeCLT/172.18.0.5:35378 Data Channel MTU parms [ L:1552 D:1450 EF:52 EB:406 ET:0 EL:3 ]
Apr 13 16:01:55 OPENVPN ovpn-second[2991978]: secondeCLT/172.18.0.5:35378 Outgoing Data Channel: Cipher 'AES-256-GCM' initialized with 256 bit key
Apr 13 16:01:55 OPENVPN ovpn-second[2991978]: secondeCLT/172.18.0.5:35378 Incoming Data Channel: Cipher 'AES-256-GCM' initialized with 256 bit key
Apr 13 16:01:55 OPENVPN ovpn-second[2991978]: secondeCLT/172.18.0.5:35378 SENT CONTROL [secondeCLT]: 'PUSH_REPLY,redirect-gateway def1,dhcp-option DNS 8.8.8.8,route
g 10,ping-restart 120,ifconfig 10.1.0.6 10.1.0.5,peer-id 0,cipher AES-256-GCM' (status=1)
Apr 13 16:01:55 OPENVPN ovpn-second[2991978]: secondeCLT/172.18.0.5:35378 PUSH: Received control message: 'PUSH_REQUEST'
Apr 13 16:01:56 OPENVPN ovpn-second[2991978]: secondeCLT/172.18.0.5:35378 MULTI: bad source address from client [::], packet dropped
Apr 13 16:01:58 OPENVPN systemd[1]: NetworkManager-dispatcher.service: Succeeded.
^C
```

Load-Balancing entre deux serveur openvpn

Je vlais cloner ma VM openvpn

Le second serveur openvpn son IP sera 172.19.0.10

Je modifie mon HAproxy comme ceci

```
backend OPENVPN-01
        balance roundrobin
        server num1 172.19.0.6:1195 check
        server num2 172.19.0.10:1195 check
        mode    tcp
```

Ensuite je test j'ai effectué une connexion que j'ai directement coupé ensuite je me suis reconnecté sa fonctionne parfaitement

Premier serveur :

```
Apr 13 17:07:19 OPENVPN ovpn-second[548]: TCP connection established with [AF_INET]172.18.0.5:39192
Apr 13 17:07:19 OPENVPN ovpn-second[548]: TCPv4_SERVER link local: (not bound)
Apr 13 17:07:19 OPENVPN ovpn-second[548]: TCPv4_SERVER link remote: [AF_INET]172.18.0.5:39192
Apr 13 17:07:19 OPENVPN ovpn-second[548]: 172.18.0.5:39192 TLS: Initial packet from [AF_INET]172.18.0.5:39192, sid=b5d7de57 25c6c6e4
Apr 13 17:07:19 OPENVPN ovpn-second[548]: 172.18.0.5:39192 VERIFY OK: depth=1, CN=FR
Apr 13 17:07:19 OPENVPN ovpn-second[548]: 172.18.0.5:39192 VERIFY OK: depth=0, CN=secondeCLT
Apr 13 17:07:19 OPENVPN ovpn-second[548]: 172.18.0.5:39192 peer info: IV_VER=3.git::d3f8b18b
Apr 13 17:07:19 OPENVPN ovpn-second[548]: 172.18.0.5:39192 peer info: IV_PLAT=win
```

Second serveur :

```
Apr 13 17:07:23 OPENVPN ovpn-second[555]: TCP connection established with [AF_INET]172.18.0.5:51470
Apr 13 17:07:23 OPENVPN ovpn-second[555]: TCPv4_SERVER link local: (not bound)
Apr 13 17:07:23 OPENVPN ovpn-second[555]: TCPv4_SERVER link remote: [AF_INET]172.18.0.5:51470
Apr 13 17:07:23 OPENVPN ovpn-second[555]: 172.18.0.5:51470 TLS: Initial packet from [AF_INET]172.18.0.5:51470, sid=88cd31b8 5c73e3dc
Apr 13 17:07:23 OPENVPN ovpn-second[555]: 172.18.0.5:51470 VERIFY OK: depth=1, CN=FR
Apr 13 17:07:23 OPENVPN ovpn-second[555]: 172.18.0.5:51470 VERIFY OK: depth=0, CN=secondeCLT
Apr 13 17:07:23 OPENVPN ovpn-second[555]: 172.18.0.5:51470 peer info: IV_VER=3.git::d3f8b18b
Apr 13 17:07:23 OPENVPN ovpn-second[555]: 172.18.0.5:51470 peer info: IV_PLAT=win
Apr 13 17:07:23 OPENVPN ovpn-second[555]: 172.18.0.5:51470 peer info: IV_NCP=2
```

On voit bien quelques secondes de différences entre les deux connexions sur les deux serveurs.