Morgan

Bissey

Procédure :

HAproxy

SOMMAIRE	
1.	INTRODUCTION
2.	Installation D'Haproxy
3.	Configuration d'HAproxy
4.	Problème rencontrer

1.Introduction

HAProxy est un logiciel open-source de répartition de charge et de proxy qui est utilisé pour améliorer la disponibilité et les performances des applications web. Il agit comme un équilibreur de charge en distribuant le trafic entrant entre plusieurs serveurs backend, ce qui permet de répartir la charge et d'éviter les points de défaillance uniques. HAProxy prend en charge différents algorithmes de répartition de charge, tels que round-robin, leastconn, et bien d'autres, et peut être configuré pour effectuer des contrôles de santé des serveurs backend afin de garantir la disponibilité des services. En outre, il offre des fonctionnalités avancées telles que la mise en cache, la compression de contenu et la possibilité de configurer des règles de routage et de redirection du trafic en fonction de divers critères. En bref, HAProxy est un outil puissant utilisé pour optimiser les performances, la fiabilité et la sécurité des applications web en gérant efficacement le trafic réseau.

2.Installation d'HAproxy

Pour commencer HAproxy est un outils de linux pour installer HAproxy nous devons tapez la commande :

```
sudo apt install haproxy
```

Une fois HAproxy installer nous devons taper la commande suivante :

```
sudo apt update && sudo apt upgrade -y
```

Pour mettre à jour tous les packet / application

3. Configuration d'HAproxy.

Il faut allez dans le repertoire ou se situe haproxy avec la commande cd :

```
oot@test-virtual-machine:/home/test# cd /etc/haproxy/
```

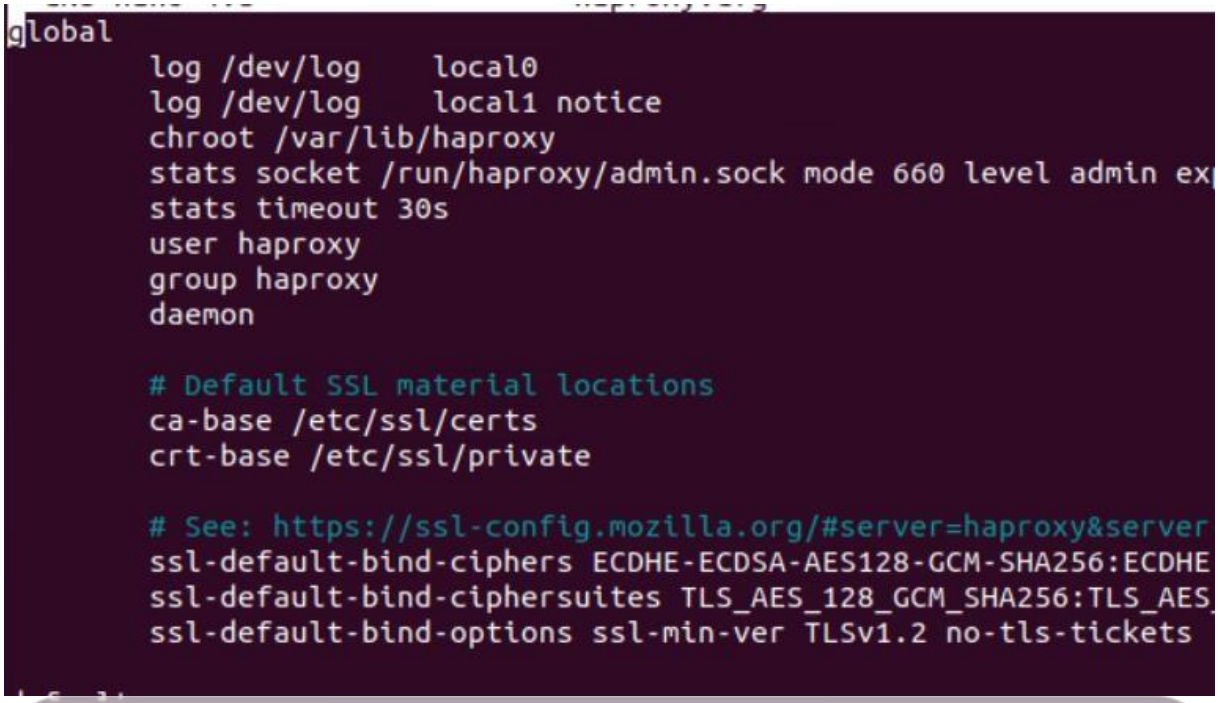
Une fois dans le repertoire nous pouvons voir avec la commande ls les fichiers dans le dossier

```
root@test-virtual-machine:/etc/haproxy# ls
errors  haproxy.cfg  haproxy.cfg.save
root@test-virtual-machine:/etc/haproxy#
```

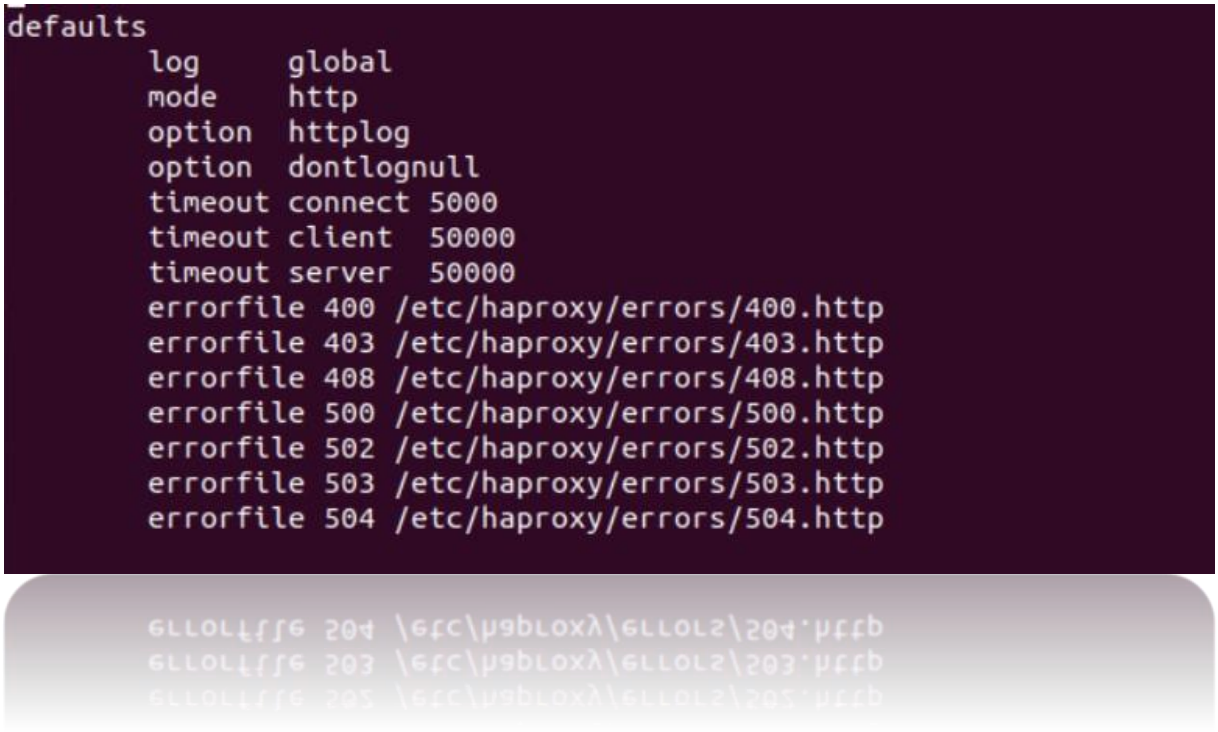
Nous allons modifier notre haproxy.cfg(les .cfg sont des dossier de configuration) et pour modifier un fichier nous pouvons utiliser plusieurs commande mais nous allons utiliser la commande nano.

```
oot@test-virtual-machine:/etc/haproxy# nano haproxy.cfg
```

Après avoir taper la commande il nous fait renter dans le dossier de configuration d'HAproxy



Dans cette image ci-dessus nous pouvons voir nos certificats.



Dans l'image si dessus nous pouvons voir fichier d'error.Imaginons que nous tapons notre adresse ip de l'un de nos serveur si il ne réponds il nous afficher les erreur que vous voyez si dessus comme error 400 /403/408/500/502/503/504.

Une fois nos messages d'erreur configurer nouus allons configurer nos server d'application web.

Pour ça nous devons déclarer nos server avec backend nom_du_server/service le mode utiliser en l'occurrence http, roundrobin pour l'équilibrage de charge puis server suivi du nom puis après l'ip du server concerner. Il ne faut pas oublier le check a la fin de chaque ip.

```
backend glpi_backend
        mode http
        balance roundrobin
        server glpi_server 192.168.10.9:80 check

backend nagios_backend
        mode http
        balance roundrobin
        server nagios_server 192.168.10.50:80 check

backend HAproxy_backend
        mode http
        option forwardfor
        balance roundrobin
        server HAproxy_server 192.168.10.70:9999 check

backend xivo_backend
        mode http
        balance roundrobin
        server xivo_server 192.168.12.3:80 check
```

Une fois tous nos server déclarer tous nos server , nous allons configurer notre cluster web

Avec les options ci-dessous

```
listen cluster_web
        bind 192.168.10.70:1000
        mode http
        option httpclose
        option forwardfor
        balance roundrobin
        server web1 192.168.10.9:80 check
        server web2 192.168.10.50:80 check
        server web3 192.168.12.8:80 check
```

Une fois notre cluster web configurer, Il reste un dernière parti qui est pour accédez à notre pages HAproxy

```
listen stats
 bind *:9999
 stats enable
 stats hide-version
 stats refresh 30s
 stats auth admin:admin
 stats uri /stats
```

Bind*= port utiliser

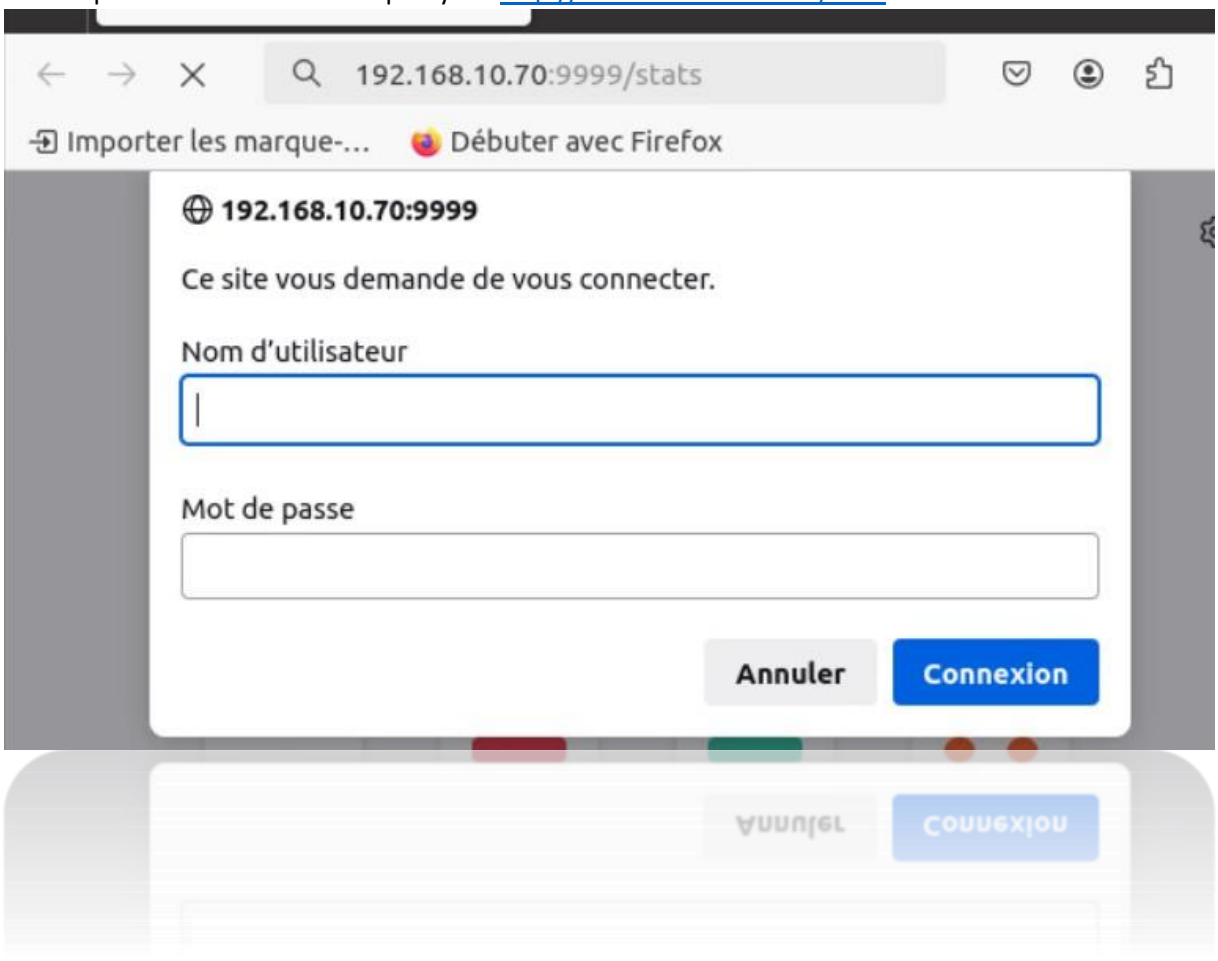
Stats enable = stats activer

Stats refresh 30s = toute les 30 sec il se mets a jour

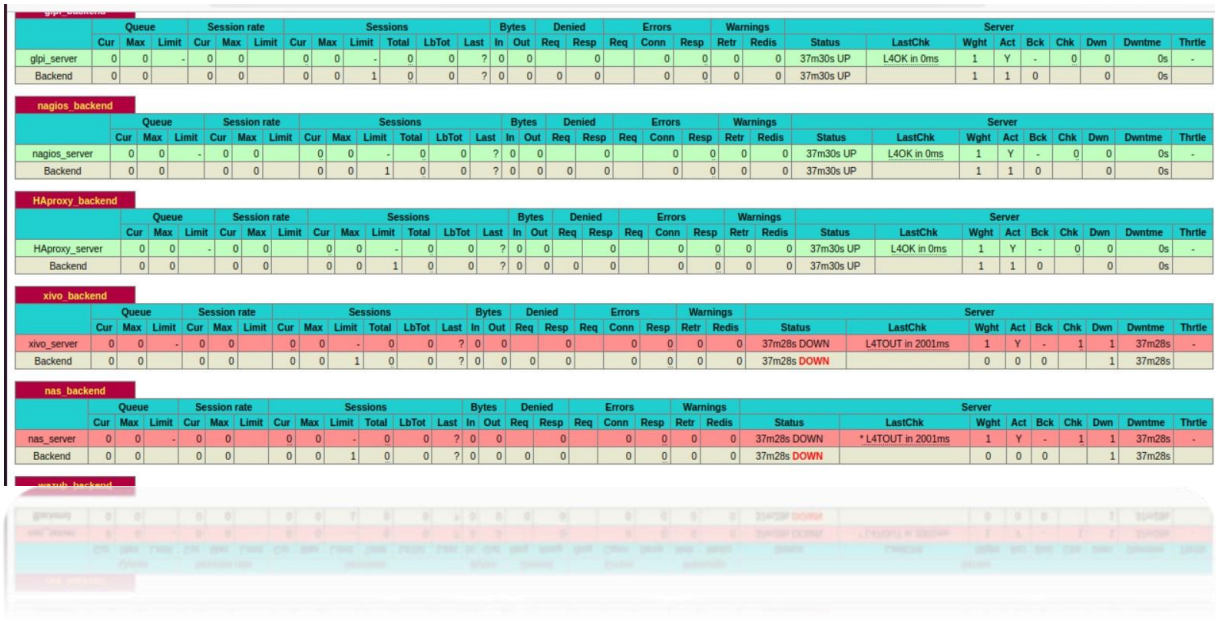
Stats auth = user / password

Stats uri /stats = liens

Le liens pour accédez a notre HAproxy est http://192.168.10.70:9999/stats



Il nous demande le user et mot de passe en l'occurrence admin : admin



gipi_backend

	Queue			Session rate			Sessions						Bytes		Denied		Errors			Warnings		Server							
	Cur	Max	Limit	Cur	Max	Limit	Cur	Max	Limit	Total	LbTot	Last	In	Out	Req	Resp	Req	Conn	Resp	Retr	Redis	Status	LastChk	Wght	Act	Bck	Chk	Dwn	Dwntme	Thrtle
gipi_server	0	0	-	0	0		0	0	-	0	0	?	0	0		0		0	0	0	0	37m30s UP	L4OK in 0ms	1	Y	-	0	0	0s	-
Backend	0	0		0	0		0	0	1	0	0	?	0	0	0	0		0	0	0	0	37m30s UP		1	1	0		0	0s	

nagios_backend

	Queue			Session rate			Sessions						Bytes		Denied		Errors			Warnings		Server							
	Cur	Max	Limit	Cur	Max	Limit	Cur	Max	Limit	Total	LbTot	Last	In	Out	Req	Resp	Req	Conn	Resp	Retr	Redis	Status	LastChk	Wght	Act	Bck	Chk	Dwn	Dwntme	Thrtle
nagios_server	0	0	-	0	0		0	0	-	0	0	?	0	0		0		0	0	0	0	37m30s UP	L4OK in 0ms	1	Y	-	0	0	0s	-
Backend	0	0		0	0		0	0	1	0	0	?	0	0	0	0		0	0	0	0	37m30s UP		1	1	0		0	0s	

HAproxy_backend

	Queue			Session rate			Sessions						Bytes		Denied		Errors			Warnings		Server							
	Cur	Max	Limit	Cur	Max	Limit	Cur	Max	Limit	Total	LbTot	Last	In	Out	Req	Resp	Req	Conn	Resp	Retr	Redis	Status	LastChk	Wght	Act	Bck	Chk	Dwn	Dwntme	Thrtle
HAproxy_server	0	0	-	0	0		0	0	-	0	0	?	0	0		0		0	0	0	0	37m30s UP	L4OK in 0ms	1	Y	-	0	0	0s	-
Backend	0	0		0	0		0	0	1	0	0	?	0	0	0	0		0	0	0	0	37m30s UP		1	1	0		0	0s	

xivo_backend

	Queue			Session rate			Sessions						Bytes		Denied		Errors			Warnings		Server							
	Cur	Max	Limit	Cur	Max	Limit	Cur	Max	Limit	Total	LbTot	Last	In	Out	Req	Resp	Req	Conn	Resp	Retr	Redis	Status	LastChk	Wght	Act	Bck	Chk	Dwn	Dwntme	Thrtle
xivo_server	0	0	-	0	0		0	0	-	0	0	?	0	0		0		0	0	0	0	37m28s DOWN	L4TOUT in 2001ms	1	Y	-	1	1	37m28s	-
Backend	0	0		0	0		0	0	1	0	0	?	0	0	0	0		0	0	0	0	37m28s DOWN		0	0	0		1	37m28s	

nas_backend

	Queue			Session rate			Sessions						Bytes		Denied		Errors			Warnings		Server							
	Cur	Max	Limit	Cur	Max	Limit	Cur	Max	Limit	Total	LbTot	Last	In	Out	Req	Resp	Req	Conn	Resp	Retr	Redis	Status	LastChk	Wght	Act	Bck	Chk	Dwn	Dwntme	Thrtle
nas_server	0	0	-	0	0		0	0	-	0	0	?	0	0		0		0	0	0	0	37m28s DOWN	* L4TOUT in 2001ms	1	Y	-	1	1	37m28s	-
Backend	0	0		0	0		0	0	1	0	0	?	0	0	0	0		0	0	0	0	37m28s DOWN		0	0	0		1	37m28s	

Une fois sur l'interface nous voyons le quel de nos server si le server est vert ca veut dire que le service / application web fonctionne L4OK signifie Layer 4 ok .