

Contextualisation :

La société **AlphaTech Services**, spécialisée dans la gestion d'infrastructures informatiques pour des clients PME, héberge plusieurs applications web internes et externes. Le service informatique souhaite améliorer la **disponibilité** et la **résilience** de son site web d'entreprise, accessible depuis Internet, tout en assurant une administration simple.

Afin de garantir la continuité de service en cas de panne d'un serveur web et d'équilibrer la charge des requêtes, l'équipe d'exploitation met en œuvre une solution basée sur **HAProxy**, un outil open source de répartition de charge (load balancer) et de haute disponibilité.

Ce TP a pour objectif de déployer cette architecture dans un environnement virtualisé de test reproduisant les conditions d'un réseau d'entreprise.



Sommaire

Belerge Jérémy

BTS SIO 2

Schéma réseau

HAProxy c'est quoi ?

Installation & préparation

Paramétrage réseau

Configuration du HAProxy

Test

La supervision du cluster HAProxy c'est quoi ?

HAProxy Stats

Exemple concret

Conclusion

Schéma réseaux





Consignes:

-HAProxy c'est quoi ?

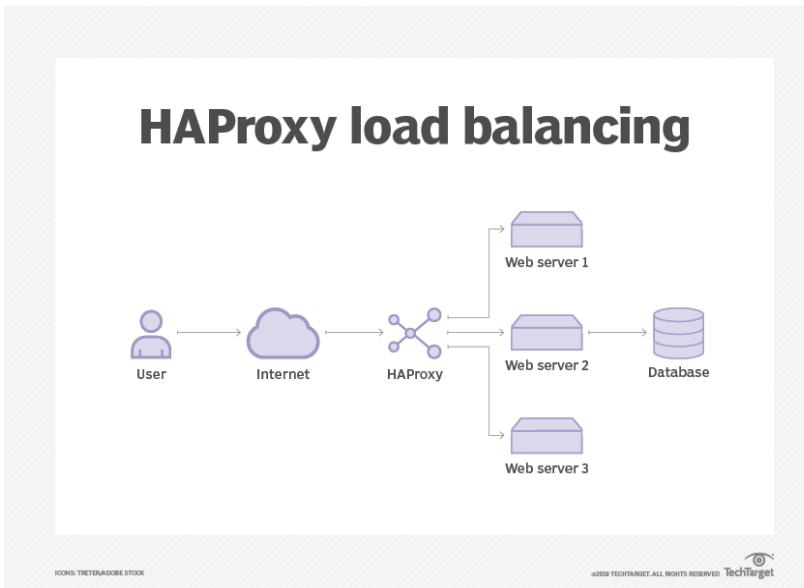
HAProxy c'est quoi ?

HAProxy est un logiciel open source conçu pour la répartition de charge (load balancing) et le rôle de reverse proxy, principalement pour des applications TCP et HTTP.

Fonction principale :

- Il distribue le trafic réseau entre plusieurs serveurs, ce qui permet d'augmenter la disponibilité, la performance et la tolérance aux pannes des applications.
- Il agit comme intermédiaire entre les clients et les serveurs backend, protégeant et optimisant également l'accès à ces derniers.

Schéma :



Source : techtarget.com



Consignes:

-Installation &
préparation

Installation & préparation de HAProxy:

On commence d'abord par installer haproxy :

```
root@debian-template:~# apt install haproxy -y
```

Preparation de nos serveurs Web:

On change notre hostname :

Nano /etc/hostname
en fonction du serveur web"

"Il nous suffit d'y mettre Web1 ou Web2

Puis, on se rend dans le dossier /var/www/html afin de supprimer la page index.html de base d'Apache.

```
root@Web1:~# cd /var/www/html
root@Web1:/var/www/html# ls
index.html  info.php
root@Web1:/var/www/html# rm index.html
```



Consignes:

-Installation & préparation

Installation de HAProxy:

On installe un site fournit dans notre consigne :

```
root@Web1:/var/www/html# wget https://github.com/technext/thegrill/archive/master.zip
```

Puis on installe unzip afin de décompresser le fichier en .zip

```
root@Web1:/var/www/html# apt install unzip
```

```
root@Web1:/var/www/html# unzip master.zip
```

Puis, on modifie le code de façon à avoir "Web 1" ou "Web 2" à titre indicatif, afin de savoir sur quel serveur nous nous trouvons.

Exemple sur notre serveur Web 1 :





Consignes:

-Paramétrage réseaux

Paramètre réseaux

Afin de faire communiquer notre infrastructure on doit y attribuer nos @ IP :

HAProxy

```
source /etc/network/interfaces.d/*
auto lo
iface lo inet loopback

auto ens18
iface ens18 inet static
    address 192.168.137.20
    netmask 255.255.255.0
    gateway 192.168.137.200

auto ens19
iface ens19 inet dhcp
```

Serveur Web 1

```
source /etc/network/interfaces.d/*
# The loopback network interface
auto lo
iface lo inet loopback

# The primary network interface
allow-hotplug ens18
iface ens18 inet static
    address 192.168.137.10
    netmask 255.255.255.0
    gateway 192.168.137.200
```

Serveur Web 2

```
# The loopback network interface
auto lo
iface lo inet loopback

# The primary network interface
allow-hotplug ens18
iface ens18 inet static
    address 192.168.137.11
    netmask 255.255.255.0
    gateway 192.168.137.200
```

Client windows

Général

Les paramètres IP peuvent être déterminés automatiquement si votre réseau le permet. Sinon, vous devez demander les paramètres IP appropriés à votre administrateur réseau.

Obtenir une adresse IP automatiquement

Utiliser l'adresse IP suivante :

Adresse IP :	192 . 168 . 137 . 13
Masque de sous-réseau :	255 . 255 . 255 . 0
Passerelle par défaut :	192 . 168 . 137 . 200



Consignes:

-Configuration du HA Proxy

Configuration de /etc/haproxy/haproxy.cfg

```
global
  log /dev/log local0 # configure la journalisation des événements vers le système de logs local.
  Daemon          # lance HAProxy en mode daemon (c'est à dire en arrière plan)
  maxconn 2048     # limite le nombre maximum de connexions simultanées que HAProxy accepte

defaults
  mode http          # indique que HAProxy traite les connexions en mode HTTP
  log global          # utilise la configuration globale de journalisation
  timeout connect 5s
  timeout client 50s
  timeout server 50s } # fixent des délais d'attente pour les connexions, la communication client et serveur

listen clusterWebAlphaTech    #: Le proxy en mode écoute qui combine frontend et backend pour le service web
  bind 192.168.137.20:80      # Le HAProxy 'écoute' sur l'adresse IP 192.168.137.20 au port 80
  mode http
  balance roundrobin          # répartit les requêtes entre serveurs la première requête ira au serveur 1, la deuxième au serveur 2 et etc...
  option httpclose            # ferme la connexion HTTP après chaque requête
  option forwardfor           # transmet simplement l'IP du client au serveur web via un en-tête HTTP.
  server SRV-WEB1 192.168.137.10:80 check   # Déclare le Web 1 afin de savoir ou envoyer les requêtes
  server SRV-WEB2 192.168.137.11:80 check   # Déclare le Web 2 afin de savoir ou envoyer les requêtes

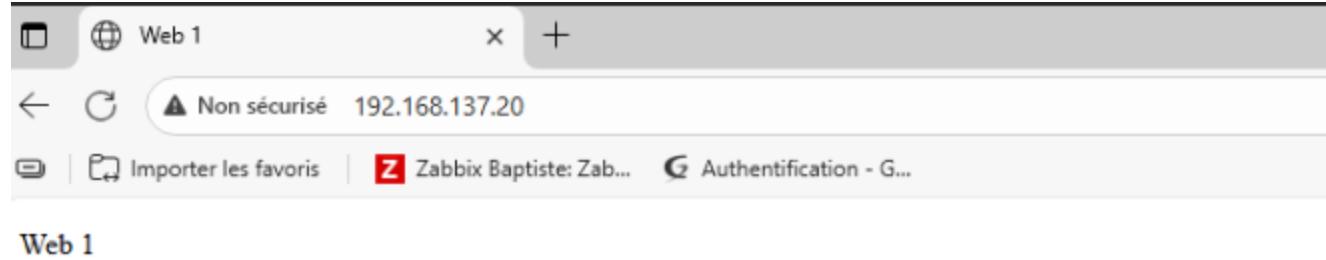
listen stats                 # Interface web pour consulter les statistiques HAProxy.
  bind 0.0.0.0:8080          # écoute sur toutes les interfaces réseau sur le port 8080
  mode http
  stats enable                # active la page de statistiques
  stats uri /statistique     # définit l'URL pour accéder aux stats
  stats refresh 30s           # rafraîchit la page toutes les 30 secondes
  stats show-node             # affiche le nom du nœud HAProxy dans l'interface
  stats auth admin:password  # On met en place un login / password (ici admin et password )
```



Consignes:

-Test

Depuis notre client Windows, on saisit alors l'adresse IP de notre HAProxy ; on arrive ainsi à accéder au premier site web, qui est hébergé sur notre serveur Web 1



Si on refait une requête HTTP, c'est-à-dire en actualisant la page web, on constate que cette requête nous amène vers notre deuxième site web, hébergé sur notre serveur Web 2 :





Consignes:

- La supervision du cluster HAProxy c'est quoi ?

La supervision d'un cluster HAProxy, c'est le fait de surveiller en temps réel l'état, la performance et la disponibilité de plusieurs instances HAProxy travaillant ensemble pour répartir le trafic réseau entre différents serveurs web ou applicatifs

Assurer la haute disponibilité : si une instance HAProxy tombe, une autre prend automatiquement le relais pour éviter toute coupure de service.

Contrôler la santé des serveurs backend : la supervision permet de détecter rapidement si un serveur web (backend) n'est plus joignable, et d'automatiquement le retirer du pool.

Surveiller les métriques : nombre de connexions, temps de réponse, charge par serveur, état des nœuds du cluster.

Automatiser les alertes et les interventions en cas d'incident technique.

Comment faire ?

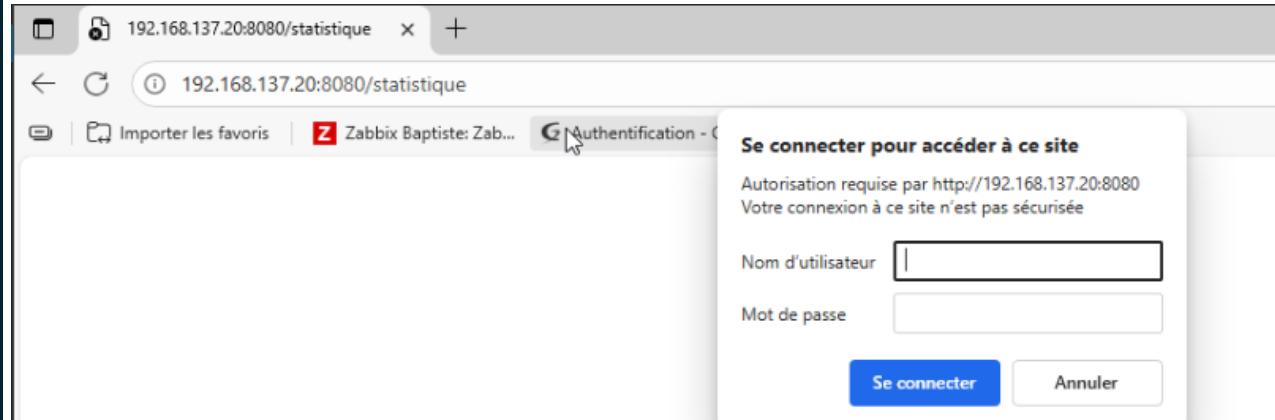
On peut faire cette supervision grâce à l'Interface web de HAProxy Stats qui permet de visualiser en temps réel les statistiques et l'état des serveurs.



Consignes:

- HAProxy Stats

Afin d'accéder à cette interface web, il suffit simplement de saisir l'adresse IP suivie de :8080/statistique, comme ci-dessous:



Où trouver ou modifier l'utilisateur et le mot de passe afin d'accéder à l'interface web ?

Pour cela, il faut consulter ou modifier les lignes que nous avons mises juste avant, lors de la configuration du HAProxy (commandes complètes ici) :

```
stats auth admin:password
```

Pour cela, il suffit simplement de remplacer "admin" par votre nom d'utilisateur, et "Password" pour définir votre mot de passe.



Consignes:

- Exemple concret

Une fois connecté, on retrouve le tableau ci-dessous ; on y voit nos serveurs ainsi qu'un code couleur, qui permet de savoir si notre serveur est "up" ou non. Actuellement, ils sont tous opérationnels et arrivent à communiquer avec notre HAProxy.

clusterWebAlphaTech																		
	Queue			Session rate			Sessions						Bytes			Denied		
	Cur	Max	Limit	Cur	Max	Limit	Cur	Max	Limit	Total	LbTot	Last	In	Out	Req	Resp		
Frontend				0	5	-	0	2	2 048	27			9 915	7 756	0	0		
SRV-WEB1	0	0	-	0	2		0	1	-	10	10	1m49s	5 092	4 096		0		
SRV-WEB2	0	0	-	0	4		0	1	-	9	9	2m6s	4 823	3 660		0		
Backend	0	0		0	5		0	1	205	19	19	1m49s	9 915	7 756	0	0		

stats																		
	Queue			Session rate			Sessions						Bytes			Denied		R
	Cur	Max	Limit	Cur	Max	Limit	Cur	Max	Limit	Total	LbTot	Last	In	Out	Req	Resp		
Frontend				1	2	-	1	2	2 048	3			459	469	0	0	0	
Backend	0	0		0	0		0	0	205	0	0	0s	459	469	0	0	0	

Or, si un serveur passe en "Down", c'est-à-dire que notre HAProxy n'arrive plus à communiquer avec ce serveur, alors il passera en rouge comme ci-dessous. Cela nous permet de repérer quel serveur a un souci afin de le régler au plus vite.

Quand un serveur est en "Down", le HAProxy ne lui envoie plus de requêtes, ce qui empêche les clients d'être redirigés vers ce serveur défaillant. Cela évite ainsi que les utilisateurs subissent des erreurs ou des lenteurs dues à ce serveur hors service, assurant une continuité du service via les autres serveurs disponibles.

clusterWebAlphaTech																					
	Queue			Session rate			Sessions						Bytes			Denied		Status			
	Cur	Max	Limit	Cur	Max	Limit	Cur	Max	Limit	Total	LbTot	Last	In	Out	Req	Resp	Req	Conn	Resp	Retr	Redis
Frontend				0	5	-	0	2	2 048	28			9 915	7 756	0	0	9	OPEN			
SRV-WEB1	0	0	-	0	2		0	1	-	10	10	2m50s	5 092	4 096	0	0	0	0	0	0	1h33m UP
SRV-WEB2	0	0	-	0	4		0	1	-	9	9	3m7s	4 823	3 660	0	0	0	0	0	0	23s DOWN
Backend	0	0		0	5		0	1	205	19	19	2m50s	9 915	7 756	0	0	0	0	0	0	1h33m UP

stats																					
	Queue			Session rate			Sessions						Bytes			Denied		Status			
	Cur	Max	Limit	Cur	Max	Limit	Cur	Max	Limit	Total	LbTot	Last	In	Out	Req	Resp	Req	Conn	Resp	Retr	Redis
Frontend				0	2	-	1	2	2 048	4			1 544	46 742	0	0	2				
Backend	0	0		0	0		0	0	205	0	0	0s	1 544	46 742	0	0	0	0	0	0	1h



Consignes:

- Conclusion

LE HAProxy est un outil qui permet de répartir le trafic internet entre plusieurs serveurs. Il aide à éviter que les serveurs soient surchargés, ce qui rend les sites et applications plus rapides et toujours disponibles. En cas de panne d'un serveur, HAProxy redirige automatiquement les demandes vers un serveur en bonne santé, assurant ainsi la continuité du service. C'est une solution simple, fiable, et efficace pour améliorer la performance et la stabilité des services en ligne.

Avoir un HAProxy au sein d'une entreprise est intéressant car il permet de répartir intelligemment le trafic entre plusieurs serveurs, évitant ainsi les surcharges. Il assure la haute disponibilité en redirigeant automatiquement les requêtes vers des serveurs en bon état de fonctionnement, ce qui évite les interruptions de service.