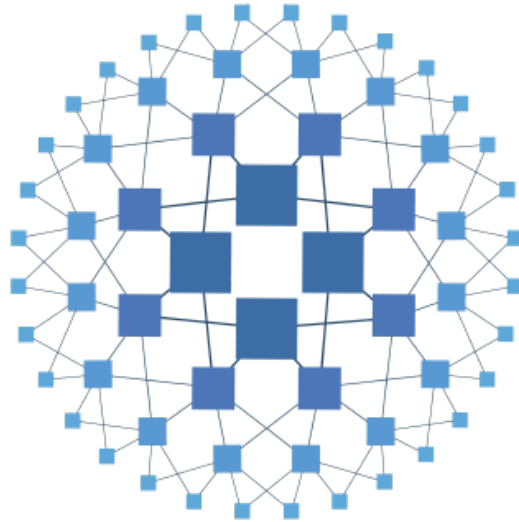


MISE EN PLACE D'UN HAPROXY

Document d'exploitation



HAPROXY

Clément MAROY

Table des matières

1. Introduction.....	3
2. Déploiement du cluster web.....	4
3. Installation HAProxy.....	5
4. Configuration HAProxy.....	8
5. Tests fonctionnels.....	10

1. Introduction

HAProxy est un logiciel open source qui permet de faire du proxying et de l'équilibrage de charge. Il est souvent utilisé pour améliorer la disponibilité, les performances et la résilience des applications web en répartissant le trafic sur plusieurs serveurs en arrière-plan.

HAProxy agit comme un proxy pour les connexions TCP et HTTP, en permettant de définir des règles pour diriger les connexions entrantes vers les différents serveurs d'application en fonction de critères tels que la charge, la disponibilité ou le type de requête.

En utilisant HAProxy, il est possible de créer des architectures haute disponibilité et scalables pour les applications web en répartissant efficacement la charge sur les différents serveurs, en évitant les points de défaillance uniques et en permettant une gestion flexible des serveurs en arrière-plan.

HAProxy est largement utilisé dans les environnements de production et est reconnu pour ses performances élevées, sa fiabilité et sa flexibilité.

2. Déploiement du cluster web

Nous allons utiliser deux conteneurs que nous allons configurer en tant que plateformes LAMP (Linux Apache Mysql PHP) afin de tester l'efficacité de notre load balancing HAProxy.

Nous mettons tout d'abord à jour nos paquets avec le gestionnaire de paquets APT :

```
root@web1:~# apt update && apt upgrade -y
```



Nous installons ensuite les différents composants de notre plateforme LAMP :

```
root@web1:~# apt install apache2 php libapache2-mod-php mariadb-server php-mysql
```



Nous modifions le contenu de `/var/www/html/index.html` avec un message personnalisé indiquant sur lequel des deux serveurs nous sommes.

```
<html><body><h1>It works!</h1>  
<p> You are on server 1</p>  
</body></html>
```

"index.html" 3L, 74B

1,1 All

Nous répliquons la même manipulation sur le deuxième conteneur

3. Installation HAProxy

Nous allons maintenant configurer un serveur Debian en serveur HAPROXY.

Nous mettons à jour nos paquets :

```
root@node1:~# apt update && apt upgrade -y
```

Nous installons Apache2 et PHP :

```
root@node1:~# apt install apache2 php
```

Nous installons le paquet HAProxy :

```
root@node1:~# apt install haproxy -y
```

Nous activons le service HAProxy avec systemctl enable de façon à ce qu'il se lance à chaque démarrage de la machine.

```
root@node1:~# systemctl enable haproxy
Synchronizing state of haproxy.service with SysV service script with /lib/systemd/systemd-sysv-install.
Executing: /lib/systemd/systemd-sysv-install enable haproxy
root@node1:~#
```

4. Configuration HAProxy

Nous ouvrons le fichier de configuration HAProxy avec vim :

```
root@node1:/etc/haproxy# vim /etc/Haproxy/Haproxy.cfg
```



Nous modifions ce fichier de configuration de façon à activer le load balancing entre nos deux serveurs web.

Pour cela nous redirigeons toutes les requêtes au port 80 de l'adresse front du HAPROXY aux deux serveurs web situés en back. Nous utilisons la répartition de charge roundrobin pour que les connexions soient distribuées à tour de rôle au serveur 1 puis au serveur 2.

Nous redirigeons également toutes les requêtes au port 8080 de l'adresse front du HAPROXY au serveur Zabbix

Et finalement toutes les connections au port 9090 à la page stats de HA Proxy.


```
frontend web
  bind *:80
  default_backend backend_web

frontend zabbix
  bind *:8080
  default_backend backend_zabbix

frontend stats
  mode http
  bind *:9090
  stats enable
  stats uri /stats
  stats refresh 10s

backend backend_web
  balance roundrobin
  server web1 172.20.0.10:80 check
  server web2 172.20.0.14:80 check

backend backend_zabbix
```

53,9

97%

5. Tests fonctionnels

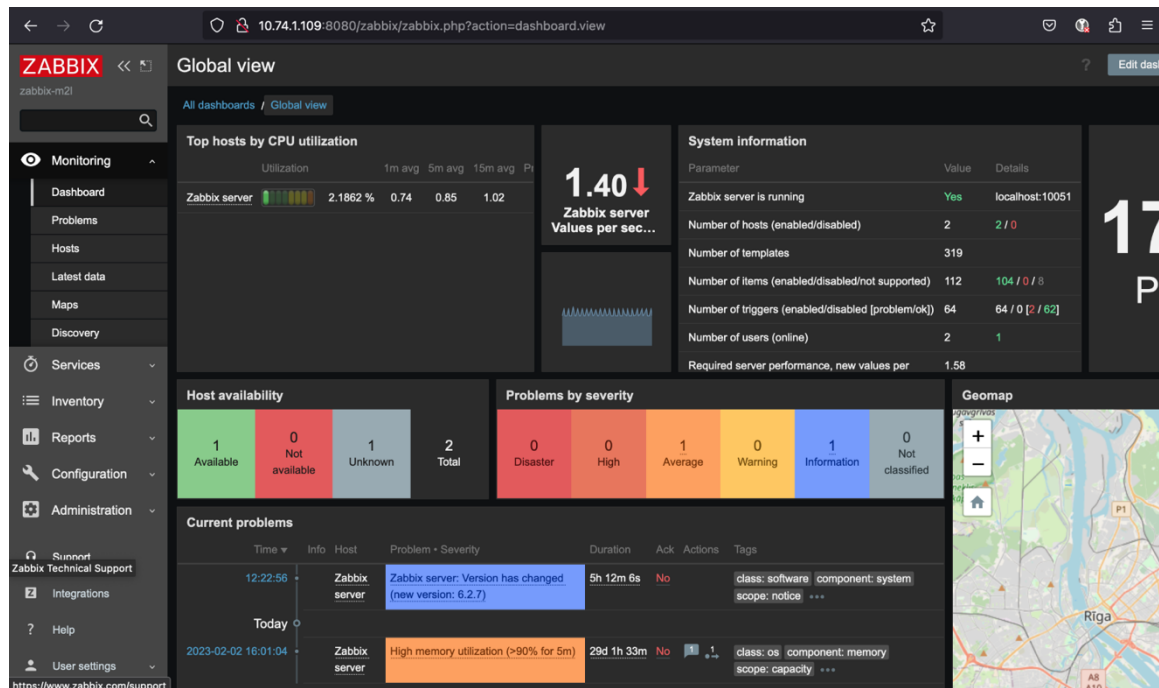
Nous effectuons une première connexion. Nous sommes redirigés vers le serveur 1 :



Nous effectuons une seconde connexion. Nous sommes bien redirigés vers le serveur 2 :



En utilisant le port 8080, nous arrivons bien sur notre serveur Zabbix :



En utilisant le port 9090 nous arrivons bien sur la page de statistiques HAPROXY :

