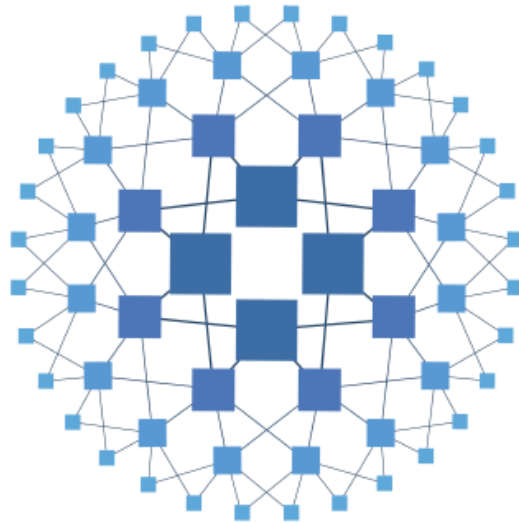


Document d'exploitation



HAPROXY

Nassim AADARI

Table des matières

1. Introduction	3
2. Déploiement de cluster Web.....	4
3. Installation HAProxy	5
4. Configuration HAProxy	8
5. Essais fonctionnels	10

1. Introduction

HAProxy est un logiciel open source qui permet le proxying et 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, vous permettant de définir des règles pour diriger les connexions entrantes vers différents serveurs d'applications en fonction de critères tels que la charge, la disponibilité ou le type de demande.


Grâce à HAProxy, il est possible de créer des architectures évolutives à haute disponibilité pour les applications Web en répartissant efficacement la charge sur les 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 connu pour ses hautes performances, sa fiabilité et sa flexibilité.

2. Déploiement de cluster Web

Nous allons utiliser deux conteneurs que nous configurerons en plateformes LAMP (Linux Apache Mysql PHP) pour tester l'efficacité de notre load balancing HAProxy. Nous mettons d'abord à jour nos packages avec le gestionnaire de packages 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 nous trouvons.

Nous installons Apache2 et PHP :

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

Nous installons le package HAProxy :

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

Nous activons le service HAProxy avec systemctl enable afin qu'il démarre 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 pour permettre l'équilibrage de charge entre nos deux serveurs Web.

Pour cela nous redirigeons toutes les requêtes vers le port 80 de l'adresse avant de l'HAPROXY vers les deux serveurs web situés à l'arrière. Nous utilisons l'équilibrage de charge roundrobin pour répartir les connexions tour à tour sur le serveur 1 puis sur le serveur 2.

Nous redirigeons également toutes les requêtes vers le port 8080 depuis l'adresse frontale du HAPROXY vers le serveur Zabbix

Et enfin toutes les connexions au port 9090 vers la page de statistiques 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. Essais fonctionnels

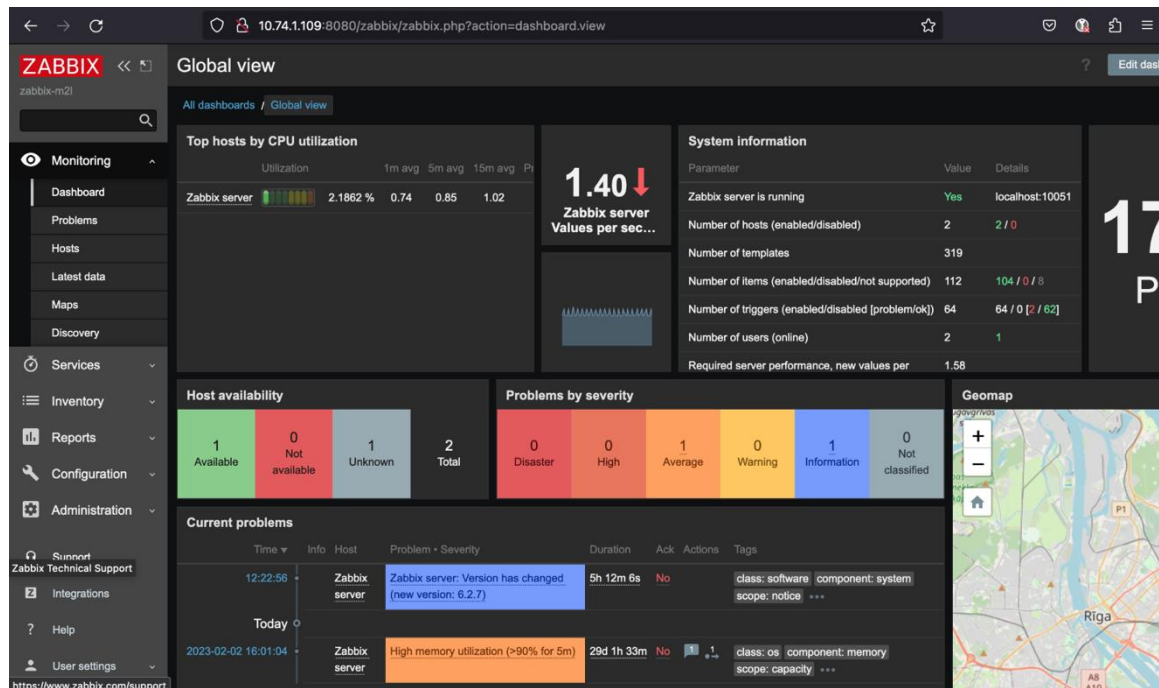
Nous établissons une première connexion. Nous sommes redirigés vers le serveur 1 :



Nous établissons une deuxième connexion. Nous sommes redirigés vers le serveur 2 :



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



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

