

Configuration HAProxy

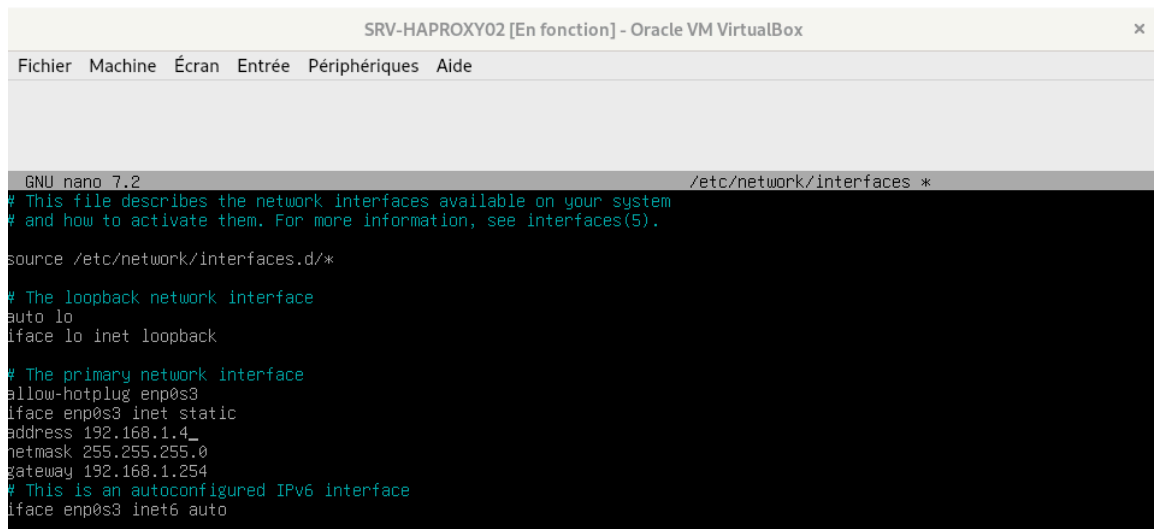
Février 2026

Architecture du Cluster

Nœud	Rôle	Adresse IP
SRV-HAPROXY01	Load Balancer #1	192.168.1.3
SRV-HAPROXY02	Load Balancer #2	192.168.1.4
SRV-WEB1	Serveur web backend	192.168.1.1
SRV-WEB2	Serveur web backend	192.168.1.2

Étape 1 — Configuration Réseau

Sur chaque serveur HAProxy, configurer une adresse IP statique. Le fichier `/etc/network/interfaces` définit la configuration réseau :



```
SRV-HAPROXY02 [En fonction] - Oracle VM VirtualBox
Fichier Machine Écran Entrée Périphériques Aide

GNU nano 7.2 /etc/network/interfaces *
# This file describes the network interfaces available on your system
# and how to activate them. For more information, see interfaces(5).

source /etc/network/interfaces.d/*

# The loopback network interface
auto lo
iface lo inet loopback

# The primary network interface
allow-hotplug enp0s3
iface enp0s3 inet static
address 192.168.1.4_
netmask 255.255.255.0
gateway 192.168.1.254
# This is an autoconfigured IPv6 interface
iface enp0s3 inet6 auto
```

Étape 2 — Accès au Fichier HAProxy

Ouvrir le fichier de configuration principal :

```
root@srv-haproxy1:~# nano /etc/haproxy/haproxy.cfg
```

Étape 3 — Configuration Global et Defaults

Les paramètres globaux et par défaut sont identiques sur les deux nœuds. Voici la configuration requise :

```
global
  log /dev/log local0

defaults
  mode http
  timeout connect 5000ms
  timeout client 50000ms
  timeout server 50000ms
```

Étape 4 — Frontend HTTP

Le frontend écoute sur le port 80 (HTTP standard) et redirige tout le trafic vers le backend http_back :

```
frontend http_front
  bind *:80
  default_backend http_back
```

Étape 5 — Backend et Répartition de Charge

Le backend utilise l'algorithme roundrobin pour distribuer les requêtes entre les deux serveurs web :

```
backend http_back
  balance roundrobin
  server SRV-WEB1 192.168.1.1:80 check
  server SRV-WEB2 192.168.1.2:80 check
```

- balance roundrobin : répartit les requêtes de façon alternée
- check : effectue une vérification de santé régulière sur chaque serveur

Config complète :

```
global
  log /dev/log local0

defaults
  mode http
  timeout connect 5000ms
  timeout client 50000ms
  timeout server 50000ms

frontend http_front
  bind *:80
  default_backend http_back

backend http_back
  balance roundrobin
  server SRV-WEB1 192.168.1.1:80 check
  server SRV-WEB2 192.168.1.2:80 check
```

Étape 6 — Interface de Statistiques

Ajouter une section dédiée pour accéder aux statistiques HAProxy en temps réel sur le port 8404 :

```
frontend stats
    mode http
    bind *:8404
    stats enable
    stats uri /stats
    stats refresh 10s
    stats admin if LOCALHOST
```

Paramètre	Signification
bind *:8404	Écoute sur le port 8404
stats uri /stats	Accès via <code>http://IP:8404/stats</code>
stats refresh 10s	Rafraîchissement auto toutes les 10 secondes
stats admin if LOCALHOST	Administration restreinte au serveur local

Étape 7 — Application de la Configuration

Après chaque modification du fichier `haproxy.cfg`, appliquer les changements sur chaque nœud :

Redémarrer le service :

```
systemctl restart haproxy
```

Vérifier le statut :

```
systemctl status haproxy
```

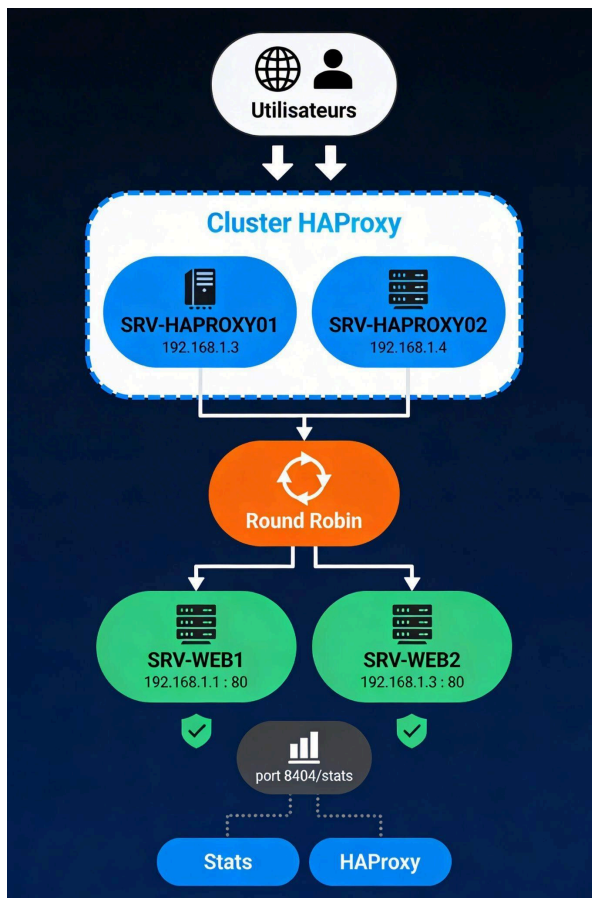
Accès aux Interfaces

Service	URL d'accès
Site web (répartition de charge)	http://@ipdulycee/
Statistiques HAProxy	http://@ipdulycee:8404/stats

Récapitulatif

Le cluster HAProxy est maintenant configuré pour :

- Écouter les requêtes HTTP sur le port 80
- Répartir la charge entre SRV-WEB1 et SRV-WEB2 en roundrobin
- Vérifier la santé de chaque serveur backend régulièrement
- Fournir une interface de statistiques sur le port 8404
- Fonctionner en cluster haute disponibilité (SRV-HAPROXY01 + SRV-HAPROXY02)



Configuration Haute Disponibilité (Keepalived / HAProxy)

```
GNU nano 7.2 keepalived.conf
vrrp_script reload_haproxy {
    script "killall -0 haproxy"
    interval 1
}

vrrp_instance VI_1 {
    virtual_router_id 100
    state MASTER
    priority 100
    # Check inter-load balancer toutes les 1 secondes
    advert_int 1
    # Synchro de l'état des connexions entre les LB sur l'interface enp0s3
    lvs_sync_daemon_interface enp0s3
    interface enp0s3
    # Authentification mutuelle entre les LB, identique sur les deux membres
    authentication {
        auth_type PASS
        auth_pass secret
    }
    # Interface réseau commune aux deux LB
    virtual_ipaddress {
        192.168.1.5/32 brd 192.168.1.255 scope global
    }

    track_script {
        reload_haproxy
    }
}
```

Ce fichier `keepalived.conf` configure le nœud **principal (MASTER)** du cluster pour assurer la haute disponibilité du service HAProxy.

Paramètres clés :

- **Rôle du nœud** : Déclaré en tant que MASTER (Priorité 100, Routeur ID 100).
- **IP Virtuelle (VIP)** : L'adresse de service commune est `192.168.1.5`, portée par l'interface `enp0s3`.
- **Surveillance (Healthcheck)** : Keepalived vérifie toutes les secondes si le processus HAProxy est actif (`killall -0 haproxy`).
- **Basculement automatique** : Si HAProxy s'arrête, la VIP bascule instantanément vers le serveur de secours (BACKUP).
- **Synchronisation réseau** : L'interface `enp0s3` gère la communication sécurisée (mot de passe `secret`) et synchronise l'état des sessions TCP entre les serveurs.

Sur le SRV-HAPROXY2, le state est en SLAVE et la priorité à 90.