

BTS SIO – SISR

Travaux Pratiques – Haute Disponibilité

Failover avec Heartbeat et Répartition de Charge avec LVS (ipvsadm)

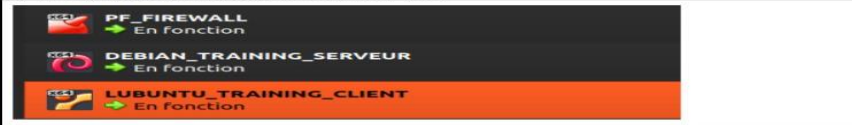
Touil Ayoub

BTS SIO SISR – 2ème année

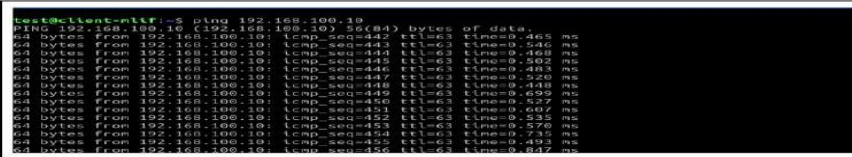
1. Présentation du contexte et de l'architecture

Ce TP aborde deux mécanismes complémentaires de haute disponibilité pour les serveurs web Linux : le failover automatique avec Heartbeat (bascule automatique d'une adresse IP flottante en cas de panne) et la répartition de charge avec LVS/ipvsadm (distribution du trafic entre plusieurs serveurs). Les machines utilisées sont des clones de serveur Debian déployés sous VirtualBox.


On va commencer par démarrer ces machines :



Bien vérifier le fonctionnement minimal de cette maquette : ping entre les machines, utilisation du serveur DNS d'adresse IP 192.168.100.10



Créer un clone de cette machine et ne la démarrer pas pour le moment. Vous devez donc avoir deux serveurs web



Démarrer ces deux machines puis effectuer maintenant la configuration initiale suivante :

Nom VirtualBox : srv-web
• Nom système : srv-web-theodore (Attention, pour la suite de ce TP le nom de la machine est très important) :

- Une interfaces réseau :
 - « eth0 ou enp... » liée à l'interface « sw-dmz », configurée en « static » avec l'adresse 192.168.200.51/24, passerelle par défaut 192.168.200.254 ;
 - Serveur DNS : 192.168.100.10
- Supprimer le fichier « /var/www/html/index.html » ;
- Créer et éditer le fichier « /var/www/html/index.html » en ne mettant qu'une seule ligne :
<h1>Srv-Web de votre_nom</h1>
afin de faire apparaître le nom du serveur « srv-web » en gros caractères.

Figure 1 – Environnement de départ : PF_FIREWALL, DEBIAN_TRAINING_SERVEUR et LUBUNTU_TRAINING_CLIENT démarrés

1.1 Configuration initiale des serveurs web

Deux serveurs Debian sont déployés : srv-web (192.168.200.51/24) et srv-web-bis (192.168.200.52/24). Les deux machines sont raccordées au réseau interne sw-dmz, avec 192.168.200.254 comme passerelle et 192.168.100.10 comme serveur DNS.

Le nom système de chaque serveur est défini avec hostnamectl, la configuration réseau est appliquée dans /etc/network/interfaces, et une page web personnalisée est déployée dans /var/www/html/index.html pour identifier chaque nœud.

Pour le nom on le modifie direct dans virtualbox
Pour le nom système, on va faire cette commande :
sudo hostnamectl set-hostname srv-web-theodore
Pour le réseau, on va faire cette commande :
<pre> sudo nano /etc/hosts et on va s'assurer d'avoir ça comme ip: 127.0.0.1 localhost 127.0.1.1 debian12 127.0.1.1 srv-web-theodore 192.168.200.51 srv-web-theodore # The following lines are desirable for IPv6 capable hosts ::1 localhost ip6-localhost ip6-loopback ff02::1 ip6-allnodes ff02::2 ip6-allrouters </pre>
ensuite :
<pre> sudo nano /etc/network/interfaces # This file describes the network # and how to activate them. For 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 192.168.200.51/24 gateway 192.168.200.254 dns-nameservers 192.168.100.10 </pre>
Supprimer le fichier « /var/www/html/index.html »
rm /var/www/html/index.html
Créer et éditer le fichier « /var/www/html/index.html » en ne mettant qu'une seule ligne :
<h1>Srv-Web de votre_nom</h1>
nano /var/www/html/index.html
Refaire la même étape pour le srv-web-bis en pensant à modifier le nom et l'ip au lieu de 51 mettre 52
Vérifier que les deux serveurs communiquent entre eux :
ping 192.168.200.51
- Vérifier qu'ils ont accès à internet :
ping google.fr
- Vérifier qu'ils sont parfaitement à l'heure, l'heure doit être identique.
date

Figure 2 – Configuration réseau de srv-web (/etc/hosts et /etc/network/interfaces) et page web personnalisée

Des vérifications préalables sont effectuées : communication entre les deux serveurs (ping croisé), accès Internet, synchronisation NTP (date identique sur les deux machines), et mise à jour des paquets (apt update).

2. Partie 1 — Failover avec Heartbeat

2.1 Principe du failover avec Heartbeat

Heartbeat est un daemon de surveillance de cluster. Il permet de surveiller en permanence l'état d'un serveur pair via des échanges de « battements de cœur » (heartbeats). En cas de défaillance du nœud maître, le nœud secondaire détecte l'absence de battements et prend automatiquement en charge l'adresse IP flottante ainsi que le service associé (ici Apache2). Le basculement est transparent pour les clients : l'adresse IP virtuelle reste accessible sans interruption perceptible.

2.2 Configuration de Heartbeat sur srv-web (maître)

Le paquet heartbeat est installé, puis trois fichiers de configuration sont créés dans /etc/ha.d/ :
ha.cf — définit le cluster : interface de communication (enp0s3), délai de mort (deadtime 5s), intervalle de battement (keepalive 1s) et liste des nœuds.

haresources — déclare les ressources gérées par le maître : l'adresse IP flottante et le service Apache2.

authkeys — contient la clé d'authentification MD5 partagée entre les nœuds (droits restreints à 600).

```
- Mettre à vos jours vos deux serveurs avec apt update.
apt update
ATTENTION
Il faut vérifier sur le firewall si les règles sont en ouvertes
STOP 1

Sur le serveur maître « srv-web »
Installer le paquet « heartbeat » :
apt install heartbeat
Créer et éditer le fichier « /etc/ha.d/ha.cf » et remplissez-le de la façon suivante

#configuration du cluster
#interface émettant et recevant les battements de coeur
bcast enp0s3
#temps, en secondes, avant que le serveur soit déclaré mort
deadtime 5
#intervalle, en secondes, entre deux battement de coeur
keepalive 1
#liste des noeuds (serveurs) formant le cluster
node srv-web-theodore srv-web-bis-theodore

Créer et éditer le fichier « /etc/ha.d/haresources » et remplissez-le de la façon suivante :

#Ressources fournies par le serveur actif
srv-web-theodore IPaddr::192.168.200.5 apache2_

srv-web-theodore IPaddr::192.168.200.5/24/enp0s3 apache2 mettre cette ligne sinon ça
marche pas

Créer le fichier « /etc/ha.d/authkeys » et remplissez-le de la façon suivante :

#numéro de méthode d'authentification (1 = 1ère méthode)
auth 1
#type de hachage et mot de passe à utiliser pour la méthode 1
#Le mot de passe utilisé est libre.
1 md5 motdepasse

Remarquer que dans ce fichier le mot de passe est indiqué en clair. Il faut donc restreindre
les droits sur ce fichier au seul utilisateur « root ». Il faut donc lancer la commande :
chmod 600 /etc/ha.d/authkeys
Sur le serveur « srv-web-bis »
- Installer le paquet « heartbeat » :
apt install heartbeat
```

Figure 3 – Fichiers ha.cf, haresources (adresse flottante 192.168.200.5/24) et authkeys

2.3 Déploiement sur srv-web-bis via SSH/SCP

OpenSSH est installé sur les deux serveurs (PermitRootLogin yes activé pour le TP). Les fichiers de configuration Heartbeat sont copiés depuis le maître vers le secondaire via SCP :

```
scp -p root@192.168.200.51:/etc/ha.d/{ha.cf,haresources,authkeys}
/etc/ha.d/
```

Le daemon heartbeat est ensuite redémarré sur les deux nœuds :

```
systemctl restart heartbeat
```

```
Installer openssh partout :
sudo apt update && sudo apt install -y openssh-server
sur les 2 machines
Puis aller dans :
nano /etc/ssh/sshd_config
enlever le # de PermitRootLogin puis mettre PermitRootLogin yes
enlever le # PasswordAuthentication
systemctl restart ssh
Puis on fait :
scp -p root@192.168.200.51:/etc/ha.d/{ha.cf,haresources,authkeys} /etc/ha.d/
Sur chacun des deux serveurs (srv-web et srv-web-bis) relancer le démon heartbeat :
systemctl restart heartbeat
Vérification du fonctionnement :
• Sur « srv-web » la commande ifconfig doit faire apparaître, en plus de « eth0 ou enp... »,
l'interface virtuelle « eth0:0 ou enp...:0 » avec l'adresse flottante « 192.168.200.5 ». Par
contre « srv-web-bis » ne possède que son adresse normale.
• Depuis le navigateur de la machine cliente, accéder à l'adresse flottante via
http://192.168.200.5.
C'est le serveur « srv-web » qui vous répond. Attention, penser à ouvrir le flux correspondant
sur votre firewall.
• Essayez plusieurs fois, c'est toujours « srv-web » qui assure le service http ;
• Éteignez, « à la barbare » (avec « poweroff ») le serveur « srv-web » ;
• Exécutez à nouveau un accès client via 1http://192.168.200.5 Après, au plus, cinq
secondes pendant lesquelles le service est interrompu, c'est le serveur « srv-web-bis » qui
assure le service http ;
remarque : penser à vider le cache du navigateur si nécessaire.
• Une commande « ip a » sur « srv-web-bis » fait apparaître que c'est lui maintenant qui
possède l'adresse flottante 192.168.200.5 ;
• Redémarrer le serveur « srv-web » ;
• Testez à nouveau un accès depuis la machine cliente du contexte via http://192.168.200.5.
Après quelques secondes, c'est à nouveau le serveur « srv-web » qui reprend le service http ;
• Une commande « ip a » sur « srv-web » fait apparaître que c'est lui maintenant qui a repris
l'adresse flottante ;
• Une commande « ip a » sur « srv-web-bis » montre qu'il ne possède plus l'adresse flottante
STOP 2
Restaurer l'instantané « apache-base » des deux serveur web (srv-web et srv-web-bis) ;
Je l'ai pas fait
• Modifier la connexion réseau de chacun d'eux :
Une seule interface réseau « eth0 » qui sera raccordée à un réseau interne nommé
«cluster», adresse statique 192.168.201.11/24 pour srv-web et 192.168.201.12/24 pour
web-bis ;
La passerelle par défaut devra être 192.168.201.1
```

Figure 4 – Copie des fichiers de configuration via SCP et configuration réseau du load-balancer

2.4 Validation du failover

Après démarrage du cluster Heartbeat, srv-web porte l'interface virtuelle enp0s3:0 avec l'adresse 192.168.200.5. La page web est accessible depuis le client via http://192.168.200.5, et c'est toujours srv-web qui répond.

Test de basculement : srv-web est éteint brutalement (poweroff). Après au plus 5 secondes d'interruption, srv-web-bis détecte la panne, acquiert l'adresse flottante et reprend le service HTTP. La commande ip a sur srv-web-bis confirme la présence de l'adresse 192.168.200.5. Au redémarrage de srv-web, le maître reprend automatiquement l'adresse flottante.

3. Partie 2 — Répartition de charge avec LVS/ipvsadm

3.1 Architecture de la grappe

Pour la répartition de charge, l'architecture est modifiée. Les serveurs web sont déplacés dans un réseau dédié « cluster » (192.168.201.0/24) inaccessible directement depuis les clients. Un nouveau nœud load-balancer-touil-ayoub est créé avec deux interfaces réseau : enp0s3 côté clients (192.168.200.5/24, sw-dmz) et enp0s8 côté grappe (192.168.201.1/24, cluster).

Les serveurs web reçoivent les nouvelles configurations IP dans le réseau cluster : srv-web sur 192.168.201.11/24 et srv-web-bis sur 192.168.201.12/24, avec 192.168.201.1 comme passerelle.

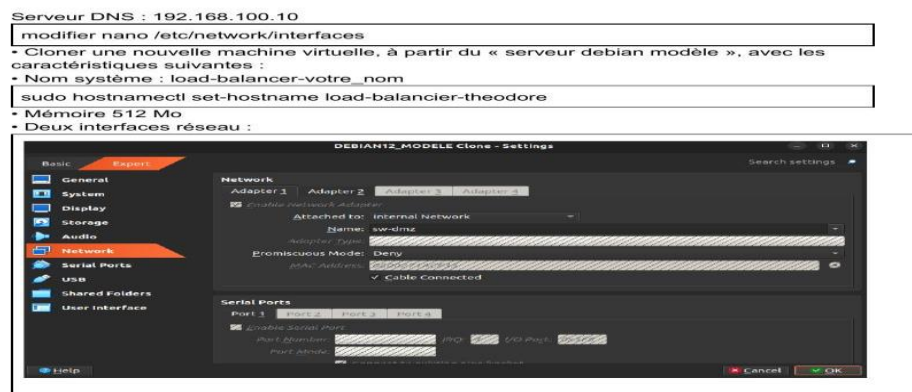


Figure 5 – Configuration du load-balancer (deux interfaces : sw-dmz et cluster)



Penser à mettre sur les srv-web et srv-web-bis le réseau : cluster

- « eth0 ou enp... » liée au réseau interne « sw-dmz », configurée en « static » avec l'adresse 192.168.200.5/24.
- La passerelle par défaut devra être 192.168.200.254
- Nom de domaine DNS : mlif.local
- Serveur DNS : 192.168.100.10

```
domain mlif.local
search mlif.local
nameserver 192.168.100.10
```

C'est l'interface par laquelle les clients accéderont au service :

- « eth1 ou enp... » liée au réseau interne nommé « cluster » avec l'adresse Statique 192.168.201.1/24. C'est l'interface permettant l'accès à la grappe de serveurs.
- attention à bien faire les modifications sur virtualbox avec le réseau interne nommé cluster. ; Vérifier que votre client dans le réseau « Clients » peut accéder au « load-balancer » avec un « ping 192.168.200.5 » ;
- Vérifier que le load-balancer accède bien aux deux serveurs web par un « ping 192.168.201.11 » et un « ping 192.168.201.12 » ;
- Vérifiez que les deux serveurs communiquent entre eux et avec le « load-balancer (192.168.201.1) » ;
- Vérifiez que le « load-balancer a accès à internet » ;

Le routage doit être activé de façon permanente. Pour cela, ouvrir le fichier /etc/sysctl.conf et chercher la ligne suivante :

```
#net.ipv4.ip_forward=1
Enlever le # et sauvegarder le fichier puis saisir la commande suivante : sysctl -p.
```

Figure 6 – Configuration réseau de srv-web (réseau cluster, 192.168.201.11/24)

Vous devriez voir s'afficher la valeur 1

installer le paquet « ipvsadm » qui installera le répartiteur de charge « LVS » :

```
sudo apt install ipvsadm
```

Editer le fichier /etc/ipvsadm.rules et complétez-le avec :

```
#Définition du service
ipvsadm -A -t 192.168.200.5:80 -s rr
#Définition des membres du cluster
ipvsadm -a -t 192.168.200.5:80 -r 192.168.201.11:80 -m
ipvsadm -a -t 192.168.200.5:80 -r 192.168.201.12:80 -m
```

Un second fichier configure le fonctionnement du démon d'ipvs : /etc/default/ipvsadm

```
# ipvsadm
# If you want to start ipvsadm on boot set this to true
AUTO="true"
# daemon method (none|master|backup)
DEMON="master"
# use interface (eth0,eth1,...)
IFACE="erp033"
# syncid to use
# (0 means no filtering of syncids happen, that is the default)
# SYNCID="0"
```

Lancer la commande :

```
dpkg-reconfigure ipvsadm
```

- Depuis votre load balancer, saisir la commande ipvsadm -Ln. Vous devriez voir les deux serveurs web de la grappe.
- Depuis la machine cliente, testez un accès au load-balancer en saisissant dans le navigateur : http://192.168.200.5.
- Essayez à plusieurs reprises en attendant environ 10 secondes entre chaque essai. A chaque nouvelle requête c'est un nœud différent du cluster qui répond (srv-web ou srv-web-bis).
- Arrêter « à la barbare » un des deux serveur web ;
- Testez à nouveau l'accès à http://192.168.200.5. Cette fois c'est toujours le même serveur qui répond vu qu'il est maintenant seul dans la grappe. Remarquez qu'il n'y a pas eu d'interruption de service suite à la panne d'un des nœuds de la grappe ;
- Redémarrer le serveur web arrêté ;
- Essayez de nouveau l'accès à http://192.168.200.5. Le serveur a repris sa place dans la

Figure 7 – Configuration réseau de srv-web-bis (réseau cluster, 192.168.201.12/24)

3.2 Activation du routage IP sur le load-balancer

Le transfert de paquets IPv4 doit être activé de façon permanente sur le load-balancer pour que le trafic puisse être routé entre les deux réseaux :

```
nano /etc/sysctl.conf → décommenter net.ipv4.ip_forward=1
sysctl -p
```

3.3 Configuration de LVS avec ipvsadm

Le paquet ipvsadm (Linux Virtual Server) est installé. Le fichier /etc/ipvsadm.rules définit le service virtuel et les membres de la grappe en mode round-robin :

```
ipvsadm -A -t 192.168.200.5:80 -s rr # Service virtuel
ipvsadm -a -t 192.168.200.5:80 -r 192.168.201.11:80 -m # srv-web
ipvsadm -a -t 192.168.200.5:80 -r 192.168.201.12:80 -m # srv-web-bis
```

Le daemon ipvs est configuré via dpkg-reconfigure ipvsadm (AUTO=true, DAEMON=master, IFACE=enp0s3).

grappe et assure de nouveau une partie de la charge de travail.

STOP 3

Si ça ne marche pas, il faut vérifier les configurations réseau

Config réseau de la load balancier :

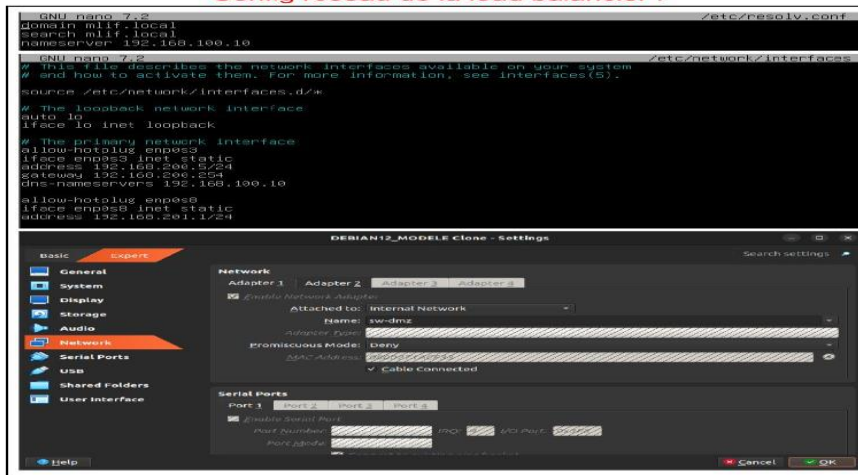


Figure 8 – Fichier /etc/ipvsadm.rules et configuration du daemon (/etc/default/ipvsadm)

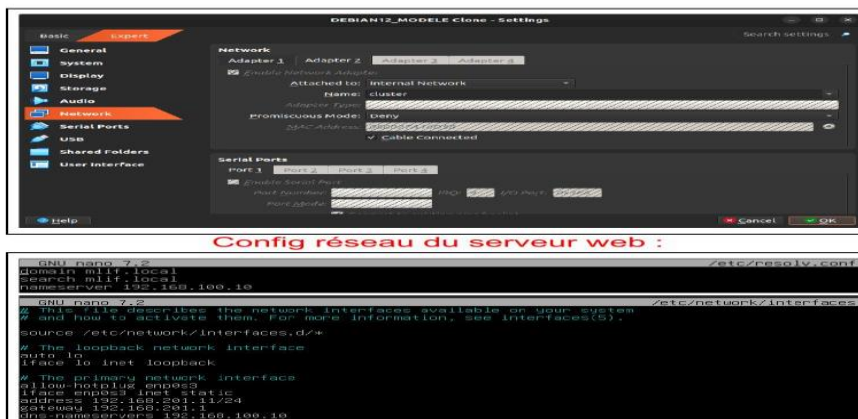


Figure 9 – Vérification des membres de la grappe avec `ipvsadm -Ln`

3.4 Validation de la répartition de charge

Depuis la machine cliente, des requêtes répétées sur `http://192.168.200.5` (avec ~10 secondes d'intervalle) montrent alternativement la page de `srv-web` et de `srv-web-bis` : la distribution round-robin fonctionne.

Test de tolérance aux pannes : en arrêtant brutalement un des serveurs web, le service reste accessible sans interruption — le load-balancer redirige automatiquement tout le trafic vers le nœud survivant. Au redémarrage du serveur arrêté, il réintègre la grappe et reprend sa part de charge.

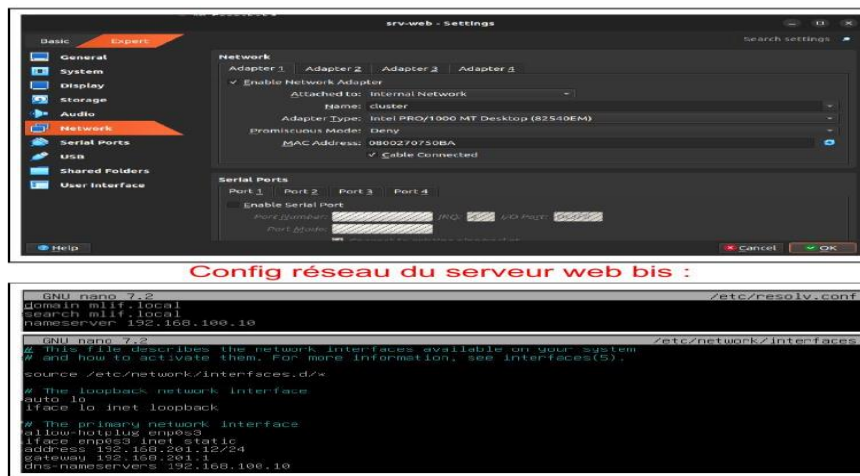


Figure 10 – Configurations réseau récapitulatives : load-balancer, `srv-web` et `srv-web-bis`



Figure 11 – Adapter 2 (réseau cluster) de srv-web-bis confirmé

4. Conclusion

Ce TP a mis en œuvre deux approches de haute disponibilité complémentaires. Heartbeat offre un failover automatique basé sur la surveillance d'un pair : en cas de panne du maître, le secondaire prend le relais en quelques secondes via une adresse IP flottante. LVS/ipvsadm implémente une répartition de charge transparente entre plusieurs nœuds : le trafic client est distribué de façon équilibrée sans point unique de défaillance. Ces deux techniques sont fondamentales pour la conception d'infrastructures web résilientes en production.