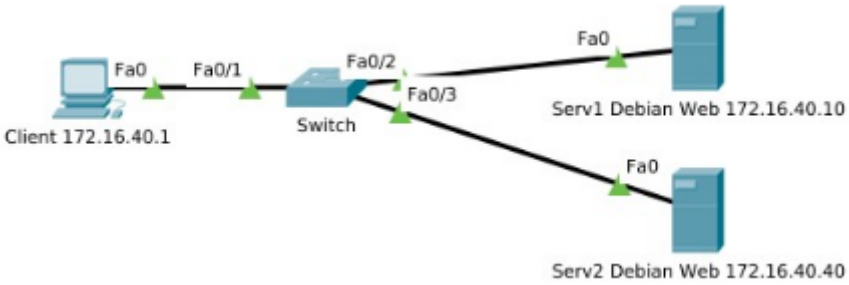
TP - HeartBeat

`Galiegue Enzo - 09/12/2022`

HeartBeat est un système sous linux permettant la mise en cluster de plusieurs serveurs, c'est ce que l'on appelle un outil de haute disponibilité. Il permet à un serveur `passif` d'attendre, et puisse prendre le relais lorsque le serveur `actif` tombe en panne. HeartBeat va donc créer une adresse "virtuelle" du groupe de serveur par laquelle les clients passeront à la place du serveur directement.

Machines	Adresse	Masque
Client	172.16.40.1	/16
SRV1	172.16.40.10	/16
SRV2	172.16.40.40	/16

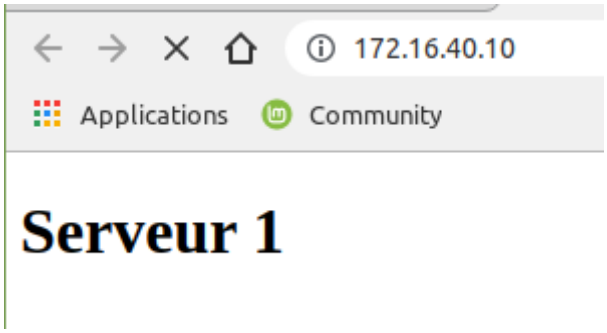
Représentation de la mise en place:



On commence par créer 2 serveurs LAMP sur un Debian 9 (La création, la configuration et le changement d'IP est déjà vu avant ainsi que la modification du fichier HTML `/var/www/html/index.html`).

on peut aussi activer le service `ssh` pour s'en servir plus tard.

Le premier nous renvoi ceci:



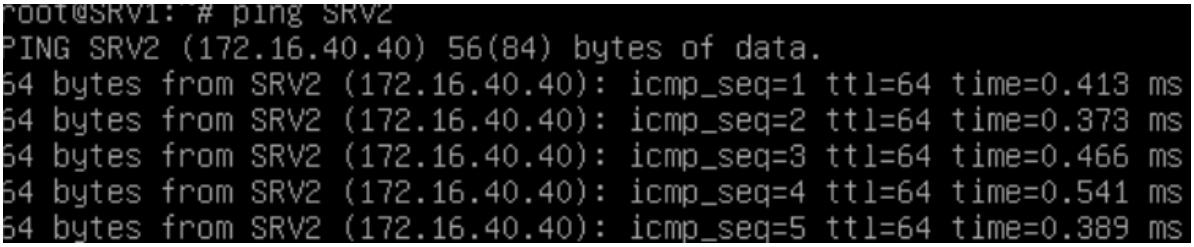
Le deuxième nous renvoi donc:

Ensuite avec la commande `nano /etc/hostname` on modifie le nom des serveurs, on fait ça sur les deux machines:

On peut maintenant passer à l'installation de HeartBeat avec la commande `apt install heartbeat` en tant qu'admin (su).

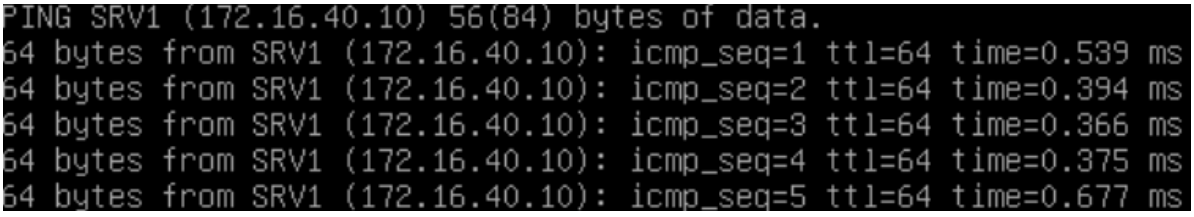
Ensuite, on va simplement modifier le ficher `hosts` avec la commande `nano /etc/hosts` pour déclarer le nom et l'IP de l'autre serveur du cluster.

On peut tester de `ping` avec `ping SRV2`.



Sur le second serveur on refait la même manipulation.

On teste alors avec la commande `ping SRV1`.



Cela permet d'éviter de rentrer les adresses IP, et donc de les remplacer par leurs noms.

On peut commencer la mise en place du service.

Les fichiers de configurations sont au nombre de 3 et dans le dossier `/etc/ha.d`

- ha.cf contient les réglages de la machine réelle.
- haresources contient les réglages du cluster (la machine à simuler).
- Authkeys contient les modes d'authentification (utile si on utilise le réseau de production).

Il faut donc créer ces fichiers.

Avec `nano ha.cf`, on peut commencer la configuration du fichier.

```
bcast ens18

debugfile /var/log/heartbeat
logfile /var/log/heartbeat

keepalive 2
deadtime 10
initdead 10

node SRV1
node SRV2

auto_failback off
```

Pour savoir sur quel nom mettre la première ligne, il suffit de faire `ip a` et de prendre le premier mot de l'interface (souvent ens18 ou enp1s0).

On passe au fichier `authkeys`, pareil, avec `nano authkeys` on met simplement:

```
auth 1
1 md5 macle
```

On peut enregistrer, et mettre les droits avec la commande `chmod 600 /etc/ha.d/authkeys`.

On fini avec la commande `nano haresources`, pour simplement mettre dedans:

```
SRV1 Ipaddr::172.16.40.30/16/ens18:0
```

Il faut maintenant faire la même manipulation sur `SRV2`, au lieu de tout réécrire, on peut utiliser l'utilitaire `scp`

```
scp -r /etc/ha.d user@172.16.40.40:/etc/ha.d
-r              : recrusive (pour la récursivité du dossier)
/etc/ha.d       : le dossier à envoyer
user            : l'utilisateur distant qui a les droits d'écriture
172.16.40.40    : IP serveur distant (ici SRV2) remplaçable par SRV2
/etc/ha.d       : dossier destinataire
```

Les droits sont normalement mit, sinon, refaire la commande `chmod 600 /etc/ha.d/authkeys` sur `SRV2`

Une fois cette étape terminée, sur les 2 serveurs on fait la commande `service heartbeat restart`

et `service heartbeat status`

```
root@SRV2:/etc/ha.d# service heartbeat status
● heartbeat.service - Heartbeat High Availability Cluster Communication and Membership
   Loaded: loaded (/lib/systemd/system/heartbeat.service; disabled; vendor preset: enabled)
   Active: active (running) since Fri 2022-12-09 15:01:26 CET; 22s ago
```

Passons aux tests du cluster, depuis le client, on voit si l'adresse virtuelle `172.16.40.30` est visible pour nous, et que nous puissions s'y connecter.

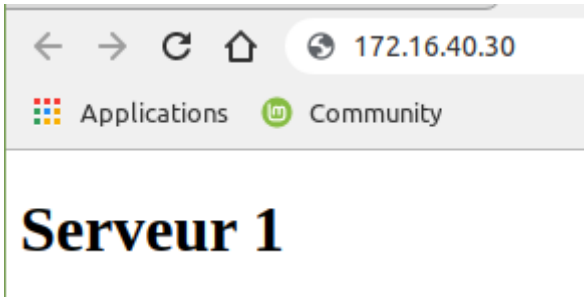
Avec la commande `ping 172.16.40.30` tout d'abord, un succès:

```
C:\Users\user>ping 172.16.40.10

Envoi d'une requête 'Ping'  172.16.40.10 avec 32 octets de données :
Réponse de 172.16.40.10 : octets=32 temps<1ms TTL=64
Réponse de 172.16.40.10 : octets=32 temps=1 ms TTL=64
Réponse de 172.16.40.10 : octets=32 temps<1ms TTL=64
Réponse de 172.16.40.10 : octets=32 temps<1ms TTL=64
```

(Les pings entre Client et SRV1 et SRV2 sont toujours possible!)

Il suffit maintenant de voir si avec le navigateur, la page s'affiche bien.



L'adresse virtuelle `172.16.40.30` nous retourne bien un serveur Apache2

En éteignant complètement le Serveur 1 (SRV1), et en testant une nouvelle fois de se connecter à l'IP virtuelle sur le navigateur depuis le client, on voit cela:

En rallumant le Serveur 1 (SRV1) mais en éteignant le Serveur 2 (SRV2), on obtient ceci:

On en conclu que lors d'une panne d'un des 2 serveurs, HeartBeat permet de faire de la Haute Disponibilité, quasiment instantanément, et cela sans qu'on change d'adresse IP.

En effet l'adresse IP Virtuelle, permet de garder une IP fixe, tout en ayant plusieurs serveurs, et donc plusieurs IP par conséquents, le Cluster, tant qu'au moins 1 serveur est disponible, permet un rendu applicatif optimal et toujours disponible.

L'utilisateur ne voit aucune différence dans son expérience auprès du service, nous avons en tant que technicien, qu'à remettre en état le serveur défaillant, pendant qu'un second prend en charge les utilisateurs.