

Jourdan Alix
Savic César

BTS SIO2

TP 3 : Implémentation de Heartbeat pour la Haute Disponibilité et la Répartition de Charge des Serveurs



Sommaire :

Sommaire :	2
Objectifs :	3
Définition de Heartbeat :	3
schémas réseaux :	3
Création des serveur :	4
Configuration du cluster :	7
Test du cluster :	10
Conclusion :	1

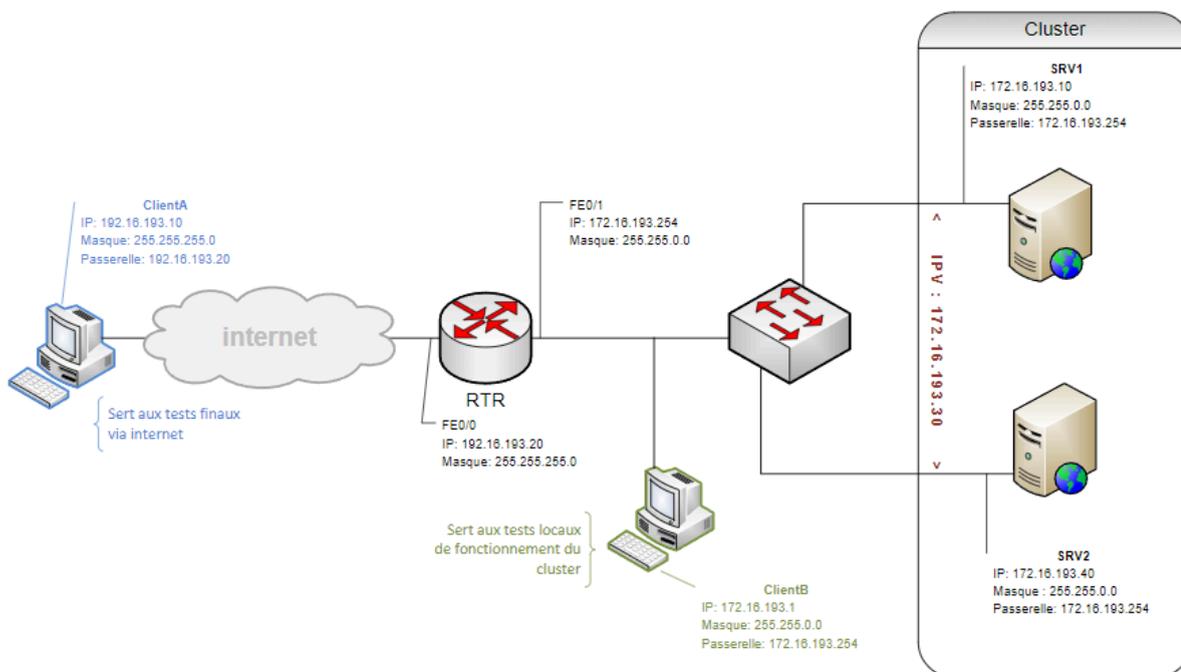
Objectifs :

Dans ce TP, nous allons découvrir le service *Heartbeat* et le mettre en place. Ce service permet de créer un cluster, c'est-à-dire un groupe de serveurs qui partagent une même adresse IP. Grâce à ce cluster, si une machine tombe en panne, les autres serveurs continuent d'assurer le service. Le client n'est ainsi pas impacté par la panne, car il voit toujours le service en fonctionnement.

Définition de Heartbeat :

Heartbeat est un service open source utilisé pour mettre en place une solution de haute disponibilité. En pratique, il surveille l'état de plusieurs serveurs dans un cluster et attribue dynamiquement une adresse IP partagée aux serveurs en fonctionnement. Si un serveur tombe en panne, un autre prend automatiquement le relais, ce qui garantit la continuité du service sans interruption pour l'utilisateur.

schémas réseaux :



Création des serveur :

Tout d'abord, nous allons installer les serveurs sur notre ESXi :

Installation de Debian 12: Sur le serveur ESXi, nous avons créé les deux serveurs qui feront partie du cluster.



Installation d'Apache2: Pour installer Apache2 sur les serveurs, nous devons exécuter la commande suivante :

```
root@debian:/home/ # apt install apache2
Lecture des listes de paquets... Fait
Construction de l'arbre des dépendances... Fait
Lecture des informations d'état... Fait
apache2 est déjà la version la plus récente (2.4.62-1~deb12u2).
0 mis à jour, 0 nouvellement installés, 0 à enlever et 17 non mis à jour.
```



Apache2 Ubuntu Default Page



It works!

This is the default welcome page used to test the correct operation of the Apache2 server after installation on Ubuntu systems. It is based on the equivalent page on Debian, from which the Ubuntu Apache packaging is derived. If you can read this page, it means that the Apache HTTP server installed at this site is working properly. You should **replace this file** (located at `/var/www/html/index.html`) before continuing to operate your HTTP server.

If you are a normal user of this web site and don't know what this page is about, this probably means that the site is currently unavailable due to maintenance. If the problem persists, please contact the site's administrator.

Configuration Overview

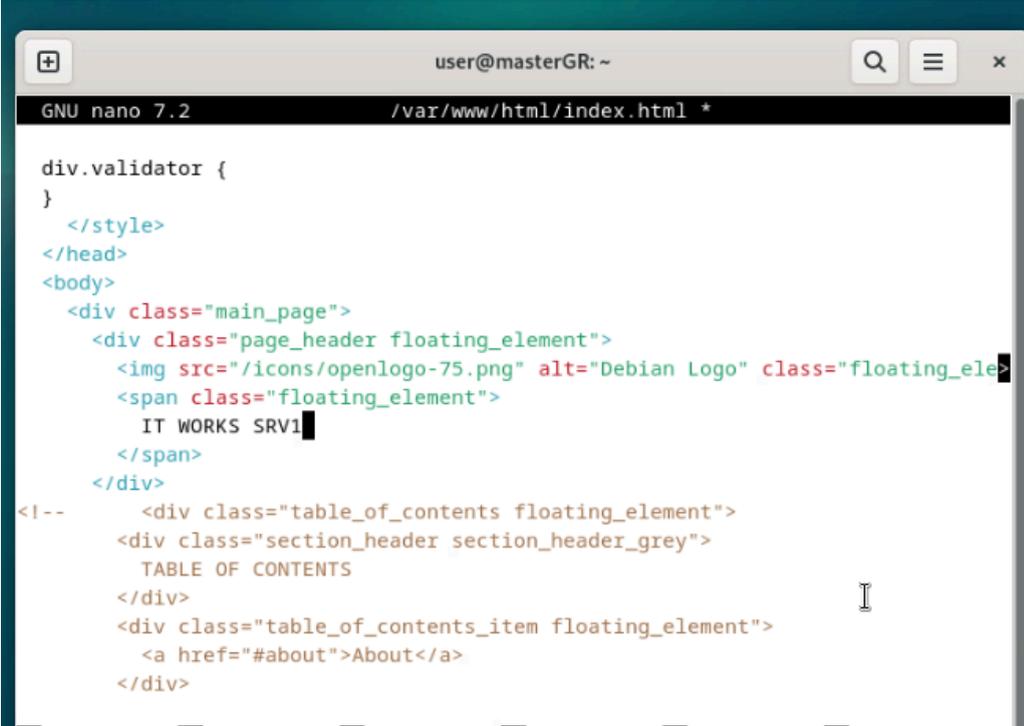
Ubuntu's Apache2 default configuration is different from the upstream default configuration, and split into several files optimized for interaction with Ubuntu tools. The configuration system is **fully documented in `/usr/share/doc/apache2/README.Debian.gz`**. Refer to this for the full documentation. Documentation for the web server itself can be found by accessing the **manual** if the `apache2-doc` package was installed on this server.

The configuration layout for an Apache2 web server installation on Ubuntu systems is as follows:

```
/etc/apache2/
|-- apache2.conf
|   |-- ports.conf
|-- mods-enabled
|   |-- *.load
|   |-- *.so
```

modification du titre la page web :

avec la commande nano /var/www/html/index.html



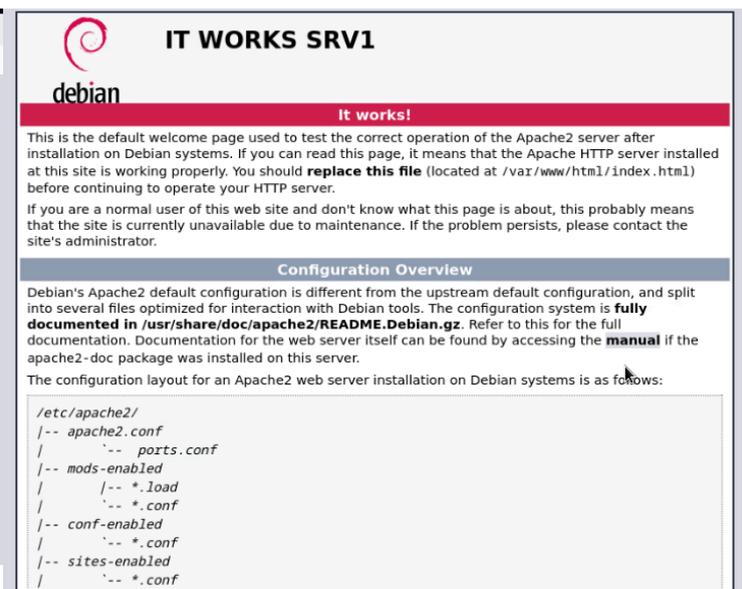
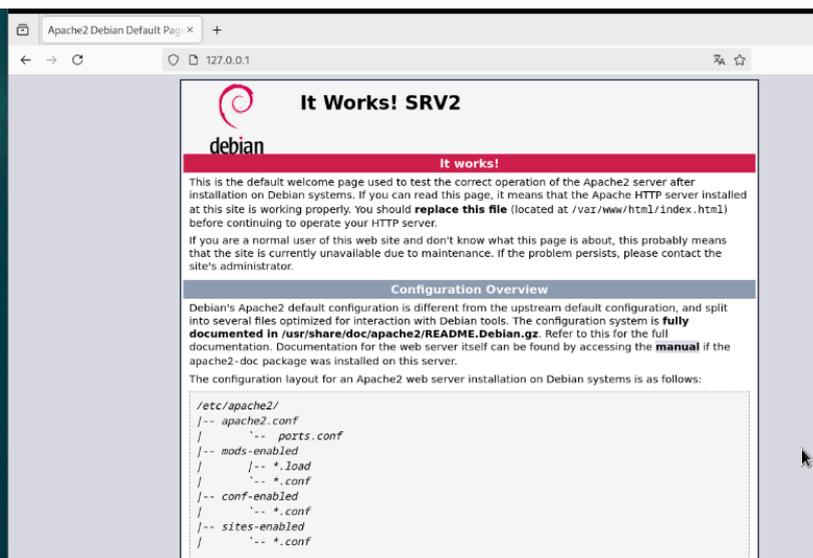
```

GNU nano 7.2 /var/www/html/index.html *

div.validator {
}
</style>
</head>
<body>
  <div class="main_page">
    <div class="page_header floating_element">
      
        IT WORKS SRV1
      </span>
    </div>
  <!--
    <div class="table_of_contents floating_element">
      <div class="section_header section_header_grey">
        TABLE OF CONTENTS
      </div>
      <div class="table_of_contents_item floating_element">
        <a href="#about">About</a>
      </div>
  </div>

```

Vérification de l'affichage après modification du titre de la page :



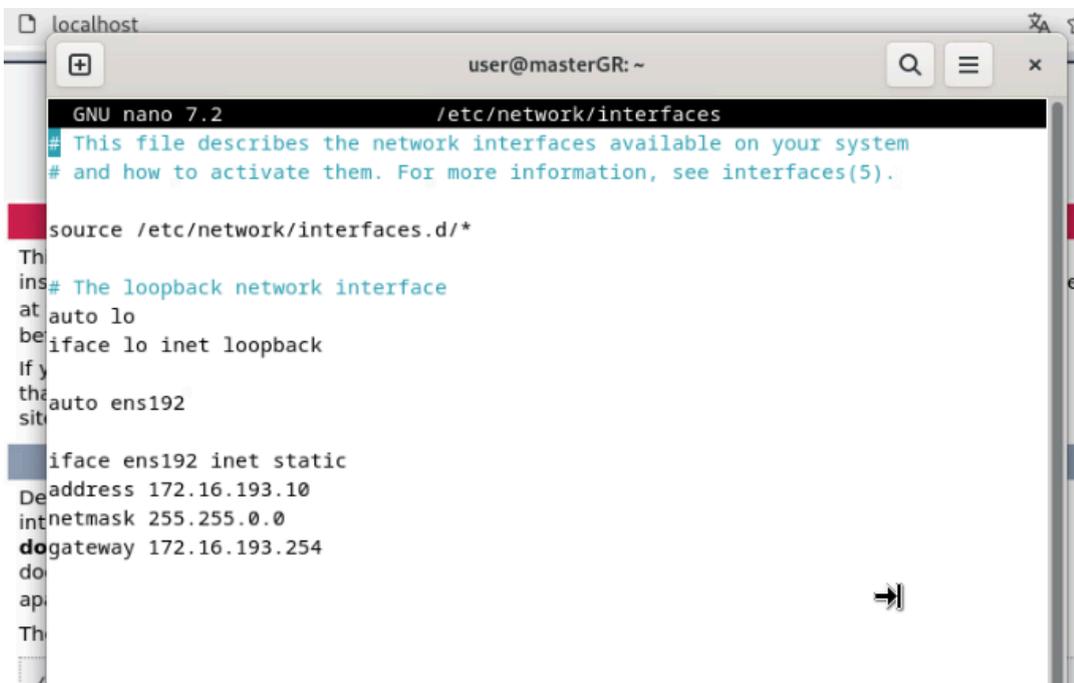
installation du service heartbeat sur chaque serveur web :

voici la commande pour l'installer apt-get install heartbeat

```
root@debian:/home/alix# apt install heartbeat
Lecture des listes de paquets... Fait
Construction de l'arbre des dépendances... Fait
Lecture des informations d'état... Fait
heartbeat est déjà la version la plus récente (1:3.0.6-13+b2).
0 mis à jour, 0 nouvellement installés, 0 à enlever et 17 non mis à jour.
```

Configuration de l'adresse IP définitive sur chacun des deux serveurs :

plan d'adressage serveur 1 :



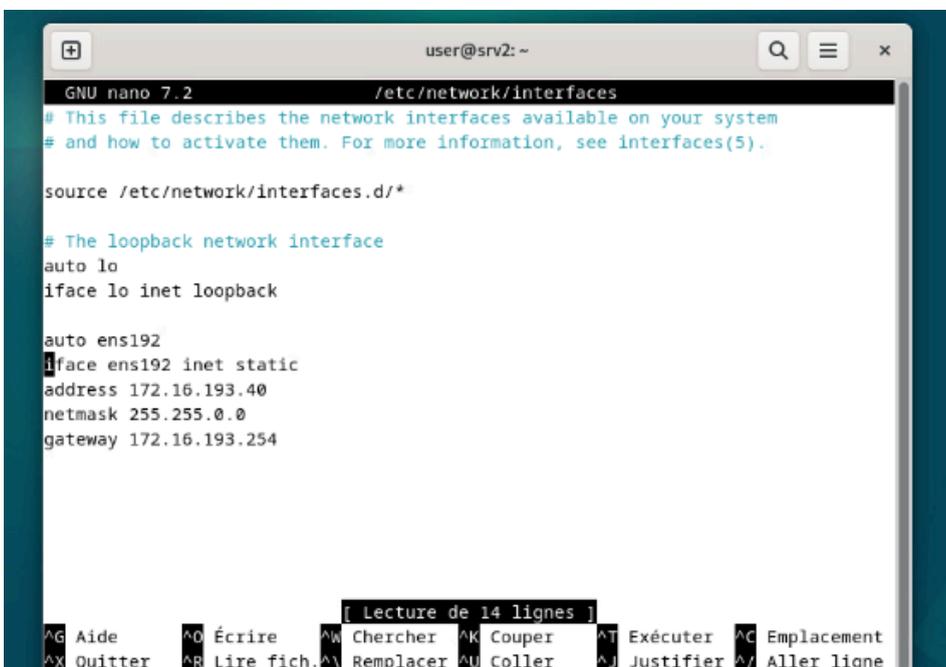
```
localhost
user@masterGR: ~
GNU nano 7.2 /etc/network/interfaces
# This file describes the network interfaces available on your system
# and how to activate them. For more information, see interfaces(5).

source /etc/network/interfaces.d/*

# The loopback network interface
auto lo
iface lo inet loopback

auto ens192
iface ens192 inet static
address 172.16.193.10
netmask 255.255.0.0
gateway 172.16.193.254
```

plan d'adressage serveur 2 :



```
user@srv2: ~
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

auto ens192
iface ens192 inet static
address 172.16.193.40
netmask 255.255.0.0
gateway 172.16.193.254

[ Lecture de 14 lignes ]
^G Aide      ^O Écrire    ^W Chercher  ^K Couper     ^T Exécuter   ^C Emplacement
^X Quitter   ^R Lire fich.^M Remplacer  ^U Coller    ^J Justifier  ^_ Aller ligne
```

Séparation des serveurs du réseau principal du lycée :
Création de switch virtuel :

 switch TP.3 cybersécurité AC

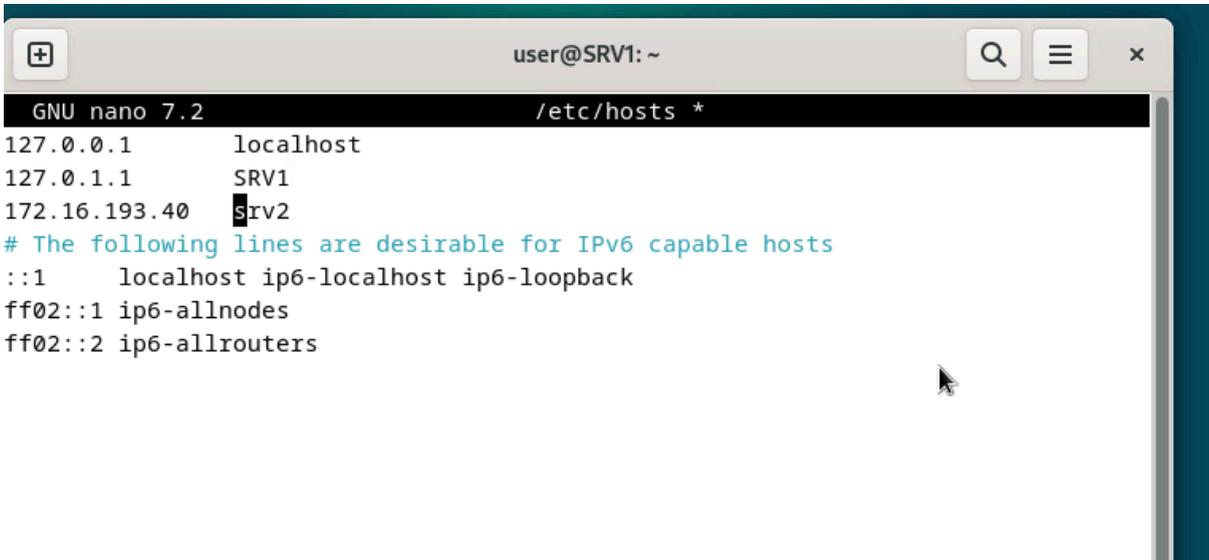
Nous avons séparé nos deux serveurs du lycée pour éviter un conflit d'adressage.

Configuration du cluster :

Modification du fichier de résolution :

Grâce à cette modification, il est désormais possible de ping les serveurs en utilisant leurs noms d'hôtes.

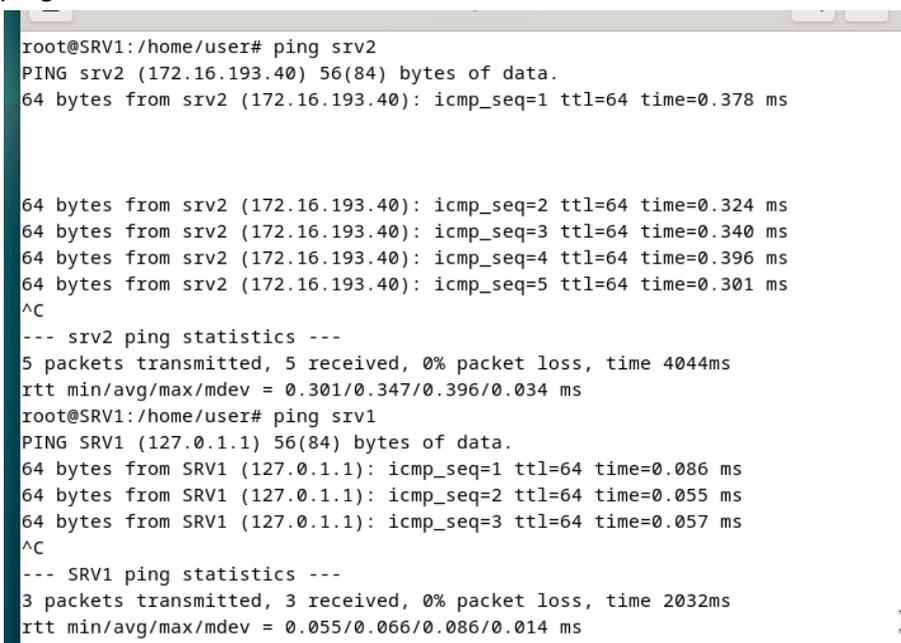
Serveur 1 :



```

GNU nano 7.2 /etc/hosts *
127.0.0.1    localhost
127.0.1.1    SRV1
172.16.193.40  srv2
# The following lines are desirable for IPv6 capable hosts
::1        localhost ip6-localhost ip6-loopback
ff02::1    ip6-allnodes
ff02::2    ip6-allrouters
  
```

ping srv1 :



```

root@SRV1:/home/user# ping srv2
PING srv2 (172.16.193.40) 56(84) bytes of data:
64 bytes from srv2 (172.16.193.40): icmp_seq=1 ttl=64 time=0.378 ms

64 bytes from srv2 (172.16.193.40): icmp_seq=2 ttl=64 time=0.324 ms
64 bytes from srv2 (172.16.193.40): icmp_seq=3 ttl=64 time=0.340 ms
64 bytes from srv2 (172.16.193.40): icmp_seq=4 ttl=64 time=0.396 ms
64 bytes from srv2 (172.16.193.40): icmp_seq=5 ttl=64 time=0.301 ms
^C
--- srv2 ping statistics ---
5 packets transmitted, 5 received, 0% packet loss, time 4044ms
rtt min/avg/max/mdev = 0.301/0.347/0.396/0.034 ms
root@SRV1:/home/user# ping srv1
PING SRV1 (127.0.1.1) 56(84) bytes of data:
64 bytes from SRV1 (127.0.1.1): icmp_seq=1 ttl=64 time=0.086 ms
64 bytes from SRV1 (127.0.1.1): icmp_seq=2 ttl=64 time=0.055 ms
64 bytes from SRV1 (127.0.1.1): icmp_seq=3 ttl=64 time=0.057 ms
^C
--- SRV1 ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 2032ms
rtt min/avg/max/mdev = 0.055/0.066/0.086/0.014 ms
  
```

Serveur 2 :

```

GNU nano 7.2 /etc/hosts
127.0.0.1    localhost
127.0.1.1    srv2
172.16.193.10  SRV1
# The following lines are desirable for IPv6 capable hosts
Th:::1      localhost ip6-localhost ip6-loopback
insff02::1  ip6-allnodes
atff02::2  ip6-allrouters
be
If y
tha
sit

```

ping serveur 2 :

```

^C
--- SRV1 ping statistics ---
2 packets transmitted, 2 received, 0% packet loss, time 1020ms
rtt min/avg/max/mdev = 0.325/0.333/0.341/0.008 ms
root@srv2:/home/user# ping SRV2
PING srv2 (127.0.1.1) 56(84) bytes of data.
64 bytes from srv2 (127.0.1.1): icmp_seq=1 ttl=64 time=0.043 ms
64 bytes from srv2 (127.0.1.1): icmp_seq=2 ttl=64 time=0.051 ms
^C
/--- srv2 ping statistics ---
/2 packets transmitted, 2 received, 0% packet loss, time 1000ms
/rtt min/avg/max/mdev = 0.043/0.047/0.051/0.004 ms
/root@srv2:/home/user#
/

```

Création du fichier de configuration Heartbeat :

La configuration sera identique pour les deux serveurs. Utilisez la commande suivante pour accéder au fichier de configuration de *Heartbeat* : `nano /etc/ha.d/ha.cf`

```

GNU nano 7.2 /etc/ha.d/ha.cf
bcast ens192
debugfile /var/log/ha-debug
logfile /var/log/ha-log
logfacility local0
keepalive 2
deadtime 10
warntime 6
initdead 60
udpport 694
node srv1
node srv2
auto_failback off

```

Ce fichier définit le node maître, autrement dit l'adresse IP partagée du cluster :

SRV1 ressources de Heartbeat :

```
GNU nano 7.2 /etc/ha.d/haresources
srv1 IPaddr::172.16.193.30/16/ens192
```

Mise en place de l'authentification :

serveur 1 et 2 :



```
user@SRV1: ~
GNU nano 7.2 /etc/ha.d/authkeys
auth 1
1 md5 "123456"
2 crc
```

Ensuite, nous allons configurer une authentification par mot de passe pour éviter toute éventualité de corruption. Il est ensuite nécessaire de protéger le fichier en réservant l'accès uniquement à l'utilisateur root en utilisant la commande suivante : `chmod 600 /etc/ha.d/authkeys`

Redémarrage du service :

Une fois les étapes effectuées, il est nécessaire de redémarrer le service *Heartbeat*, en commençant par le serveur 1, puis le serveur 2. Utilisez les commandes suivantes pour chaque serveur :

```
service heartbeat stop
```

```
service heartbeat start
```

Vérification de la carte virtuelle :

Sur les serveurs, la carte virtuelle avec l'adresse IP partagée apparaît correctement sous le nom `ens192:0`.

```

    valid_lft forever preferred_lft forever
2: ens192: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc mq state UP group de
fault qlen 1000
    link/ether 00:0c:29:16:bf:e9 brd ff:ff:ff:ff:ff:ff
    altname enp11s0
    inet 172.16.193.10/16 brd 172.16.255.255 scope global ens192
        valid_lft forever preferred_lft forever
    inet 172.16.193.30/16 brd 172.16.255.255 scope global secondary ens192:0
        valid_lft forever preferred_lft forever
    inet6 fe80::20c:29ff:fe16:bfe9/64 scope link
        valid_lft forever preferred_lft forever
root@SRV1: /home/user#

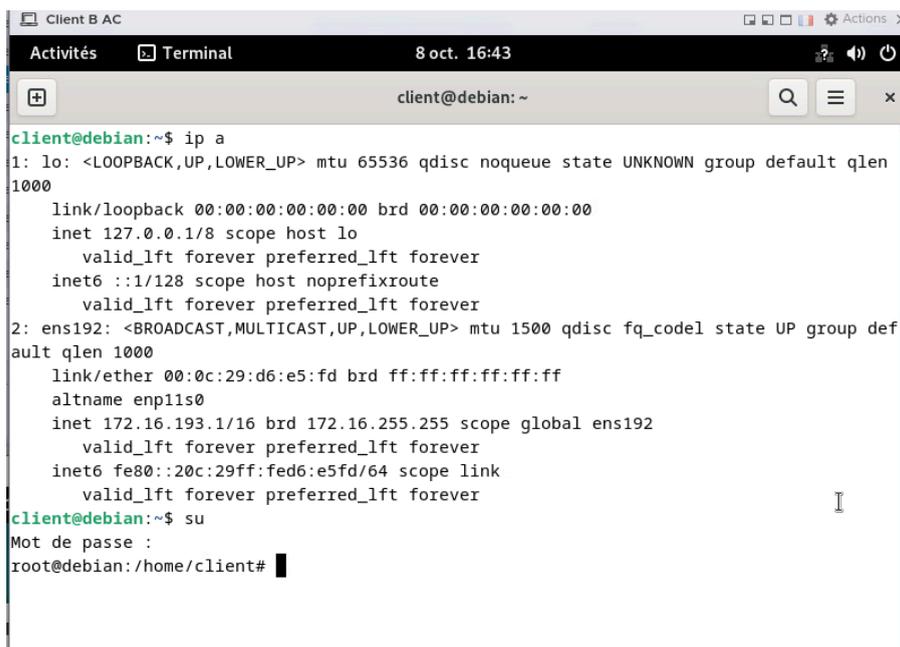
```

Test du cluster :

Dans un premier temps, nous allons tester le fonctionnement du cluster localement avec des stations clientes que nous allons créer.

Lors de la configuration de la machine cliente, nous allons attribuer l'adresse IP selon le schéma réseau et la connecter au switch virtuel afin qu'elle fasse partie du même réseau : local.IP Client B

ClientB
IP: 172.16.193.1
Masque: 255.255.0.0
Passerelle: 172.16.193.254



```

Client B AC
Activités Terminal 8 oct. 16:43
client@debian: ~
client@debian:~$ ip a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen
1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
        valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host noprefixroute
        valid_lft forever preferred_lft forever
2: ens192: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group def
ault qlen 1000
    link/ether 00:0c:29:d6:e5:fd brd ff:ff:ff:ff:ff:ff
    altname enp11s0
    inet 172.16.193.1/16 brd 172.16.255.255 scope global ens192
        valid_lft forever preferred_lft forever
    inet6 fe80::20c:29ff:fed6:e5fd/64 scope link
        valid_lft forever preferred_lft forever
client@debian:~$ su
Mot de passe :
root@debian: /home/client#

```

Ping du client vers le cluster :

```

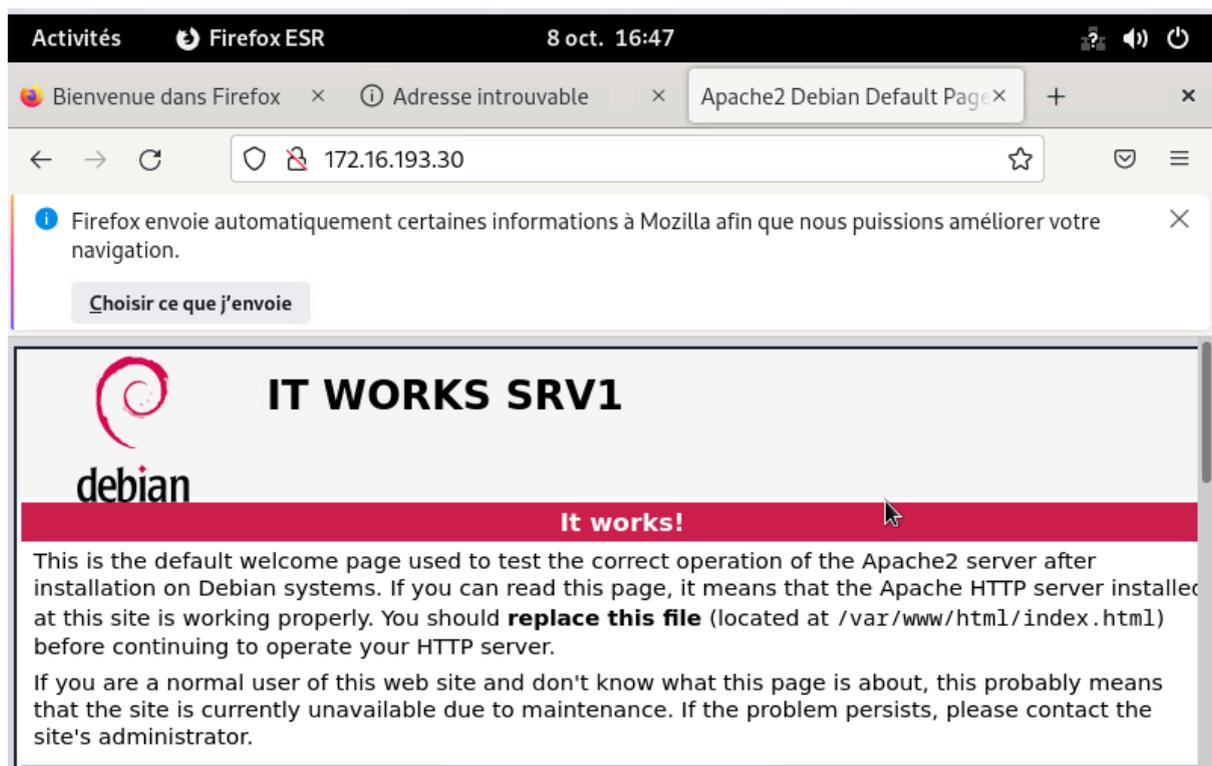
    valid_itt forever preterred_itt forever
client@debian:~$ su
Mot de passe :
root@debian:/home/client# ping 172.16.193.30
PING 172.16.193.30 (172.16.193.30) 56(84) bytes of data.
64 bytes from 172.16.193.30: icmp_seq=1 ttl=64 time=0.750 ms
64 bytes from 172.16.193.30: icmp_seq=2 ttl=64 time=0.328 ms
64 bytes from 172.16.193.30: icmp_seq=3 ttl=64 time=0.393 ms
64 bytes from 172.16.193.30: icmp_seq=4 ttl=64 time=0.397 ms
64 bytes from 172.16.193.30: icmp_seq=5 ttl=64 time=0.383 ms
64 bytes from 172.16.193.30: icmp_seq=6 ttl=64 time=0.362 ms
^C

```

Test de l'adresse IP du cluster sur la page web :

Nous voyons correctement la page du serveur 1 sur le client B .

Affichage de la page web :

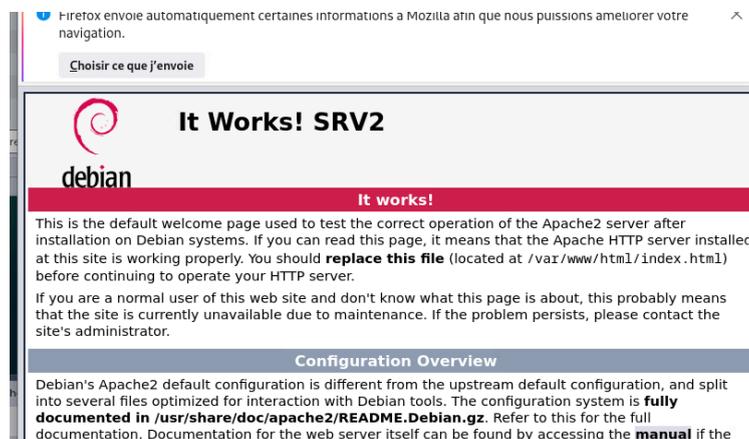


Nous allons provoquer une panne du serveur 1 en éteignant la carte réseau.
avec la commande suivante

```
root@SRV1:/home/user# sudo ifdown ens192
root@SRV1:/home/user# ip a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
        valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host noprefixroute
        valid_lft forever preferred_lft forever
2: ens192: <BROADCAST,MULTICAST> mtu 1500 qdisc mq state DOWN group default qlen 1000
    link/ether 00:0c:29:16:bf:e9 brd ff:ff:ff:ff:ff:ff
    altname enp11s0
root@SRV1:/home/user#
```

Affichage de la page web du serveur 2, car le serveur 1 est en panne :

Étant donné que nous avons provoqué une panne, grâce au service *Heartbeat*, qui est lié au cluster, cela a permis au serveur 2 de prendre le relais, car il partage également la même adresse IP.



Conclusion :

Dans ce TP, nous avons mis en place le service Heartbeat pour créer un cluster de serveurs, assurant une haute disponibilité. Nous avons pu observer qu'un cluster regroupe plusieurs serveurs dans une « bulle » avec une adresse IP partagée, ce qui permet aux deux serveurs de se comporter comme une seule entité. Cette configuration garantit la disponibilité des services, même en cas de défaillance de l'un des serveurs. Heartbeat est ainsi utilisé pour sécuriser la disponibilité des serveurs et minimiser les interruptions de service. Ce TP nous a permis de comprendre le fonctionnement de Heartbeat.