

TP 3 : Cybersécurité Haute disponibilité et répartition des charges des serveurs.



<u>I) Mise en place d'un solution haute disponibilité HeartBeat</u>	2
<u>II) Test du Cluster avec la solution de haute disponibilité HeartBeat</u>	6
<u>III) Mise en place d'un Répartiteur avec IPVS</u>	8

I) Mise en place d'une solution haute disponibilité HeartBeat

Définition

Dans une solution de haute disponibilité, le "heartbeat" est comme une vérification régulière des échanges de signaux entre les différents éléments du système. Ces signaux sont essentiels pour s'assurer que chaque composant fonctionne. Si l'un de ces éléments cesse d'émettre ces signaux, les autres éléments le repèrent et prennent le relais pour garantir que le service continue sans interruption. Cela assure une sorte de filet de sécurité pour maintenir le système opérationnel, même en cas de panne d'un élément.

- Création de deux serveurs SRV1 et SRV2

Vérification de la source list du serveur pour permettre l'installation d'Apache 2 et Heartbeat.

Apache2 installation

```
administrateur@Debian-12-Bookworm: ~
Lecture des listes de paquets... Fait
Construction de l'arbre des dépendances... Fait
Lecture des informations d'état... Fait
0 mis à jour, 0 nouvellement installés, 0 à enlever et 37 non mis à jour.
root@Debian-12-Bookworm:/home/administrateur# sudo apt-get install apache2
Lecture des listes de paquets... Fait
Construction de l'arbre des dépendances... Fait
Lecture des informations d'état... Fait
Les paquets supplémentaires suivants seront installés :
  apache2-data apache2-utils
Paquets suggérés :
  apache2-doc apache2-suexec-pristine | apache2-suexec-custom
Les NOUVEAUX paquets suivants seront installés :
  apache2 apache2-data apache2-utils
0 mis à jour, 3 nouvellement installés, 0 à enlever et 37 non mis à jour.
Il est nécessaire de prendre 577 ko dans les archives.
Après cette opération, 1 890 ko d'espace disque supplémentaires seront utilisés.
Souhaitez-vous continuer ? [O/n] o
Réception de :1 http://deb.debian.org/debian bookworm/main amd64 apache2-data al
1 2.4.57-2 [160 kB]
Réception de :2 http://deb.debian.org/debian bookworm/main amd64 apache2-utils a
amd64 2.4.57-2 [202 kB]
Réception de :3 http://deb.debian.org/debian bookworm/main amd64 apache2 amd64 2
.4.57-2 [215 kB]
```

HeartBeat installation

```
administrateur@Debian-12-Bookworm: ~
enabling module filter.
enabling module deflate.
enabling module status.
enabling module reqtimeout.
enabling conf charset.
enabling conf localized-error-pages.
enabling conf other-vhosts-access-log.
enabling conf security.
enabling conf serve-cgi-bin.
enabling site 000-default.
created symlink /etc/systemd/system/multi-user.target.wants/apache2.service -> /l
b/systemd/system/apache2.service.
created symlink /etc/systemd/system/multi-user.target.wants/apache-htcacheclean.
service -> /lib/systemd/system/apache-htcacheclean.service.
traitement des actions différées (« triggers ») pour man-db (2.11.2-2) ...
root@Debian-12-Bookworm:/home/administrateur# apt-get install heartbeat
Lecture des listes de paquets... Fait
Construction de l'arbre des dépendances... Fait
Lecture des informations d'état... Fait
Les paquets supplémentaires suivants seront installés :
  cluster-glue corosync fence-agents gawk iptables libcfg7 libcib27 libcmapi4
  libcorosync-common4 libcpq4 libcrmcluster29 libcrmcommon34 libcrmservice28
  libbest0 libfastjson4 libheartbeat2 libip6tc2 libknet1 liblognorm5 liblrm2
  liblrm28 libnet-telnet-perl libnet1 libnozzle1 libopenhpi3 libopenipmi0
```

- Après avoir installé le package APACHE ET HEARTBEAT nous pouvons alors accéder à la page HTML du serveur Apache pour la modifier et stipuler que nous sommes sur SRV1 ou 2.

```
Lecture des listes de paquets... Fait
root@Debian-12-Bookworm:/home/administrateur# nano /var/www/html/index.html
root@Debian-12-Bookworm:/home/administrateur#
```

```

GNU nano 7.2 /var/www/html/index.html
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
  <meta http-equiv="Content-Type" content="text/html; charset=UTF-8" />
  <title>Apache2 SRV 1 Debian Default Page: It works</title>
  <style type="text/css" media="screen">
* {
  margin: 0px 0px 0px 0px;
  padding: 0px 0px 0px 0px;
}

body, html {
  padding: 3px 3px 3px 3px;

  background-color: #D8DBE2;

  font-family: Verdana, sans-serif;
  font-size: 11pt;
  text-align: center;
}
  
```

^G Aide ^O Écrire ^W Chercher ^K Couper ^T Exécuter ^C Emplacement
 ^X Quitter ^R Lire fich. ^\ Remplacer ^U Coller ^J Justifier ^/ Aller ligne

Nous renommons le serveur avec la commande nano /etc/hostname pour donner le nom SRV1 ou SRV2.

```

GNU nano 7.2 /etc/hostname
SRV1
  
```

nom du fichier à écrire: /etc/hostname

- Ping entre SRV 1 et SRV2

The image shows two terminal windows side-by-side. The left window, titled 'debian 12 serveur cyberssec hualte dispo', shows the configuration of network interfaces. It lists the loopback interface 'lo' with IP 127.0.0.1, and the ethernet interface 'ens192' with IP 192.168.50.2. It also shows the configuration of the 'link' interface with IP 192.168.50.1. The right window, titled 'Cybersecuirté serveur 2 Haute debit', shows the configuration of the 'link' interface with IP 192.168.50.1. Both windows show the output of the 'ping' command, indicating successful connectivity between the two servers.

Nous avons donc pour mission de mettre en place la configuration du cluster entre les deux serveurs SRV1 et SRV2.

nous éditons alors le fichier “ nano /etc/hosts “ : nous déclarons alors sur SRV1 l’adresse et le nom du SRV2 pour que le cluster

The image shows a terminal window titled 'remy@debian: ~' with the file '/etc/hosts' open in nano. The file contains the following content:

```
127.0.0.1 localhost
127.0.1.1 debian

# The following lines are desirable for IPv6 capable hosts
127.0.0.1 localhost
127.0.0.1 SRV1
192.168.50.2 SRV2
```

Nous remarquons alors que les Ping SRV1 et SRV2 fonctionnent en ayant remplacé l’adresse IP par un nom.

The image shows two terminal windows side-by-side. The left window, titled 'SRV2 RB linux 12', shows the output of the 'ping' command from SRV2 to SRV1 using the hostname 'SRV1'. The right window, titled 'SRV1 RB linux 12', shows the output of the 'ping' command from SRV1 to SRV2 using the hostname 'SRV2'. Both windows show successful ping results, indicating that the cluster configuration is working.

- Configuration de Heartbeat copié sur les 2 serveurs SRV1 et SRV2 pour accéder à la page de configuration des fichiers fondamentaux nous faisons la commande

The image shows two terminal windows side-by-side. The left window is titled 'SRV2 RB linux 12' and the right window is titled 'SRV1 linux 12 RB'. Both windows show the output of the 'cat /etc/ha.d/ha.cf' command in a nano 7.2 editor. The configuration is identical on both servers:

```
bcast enp11s0
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
```

- Mise en place de l'adresse Virtuelle équivalente pour permettre une redondance entre SRV1 et SRV2 en cas de panne.

The image shows two terminal windows side-by-side. The left window is titled 'SRV2 RB linux 12' and the right window is titled 'SRV1 linux 12 RB'. Both windows show the output of the 'cat /etc/ha.d/haresources' command in a nano 7.2 editor. The configuration is identical on both servers:

```
SRV2 IPaddr::192.168.50.10/24/enp11s0
```

on the left and

```
SRV1 IPaddr::192.168.50.10/24/enp11s0
```

on the right.

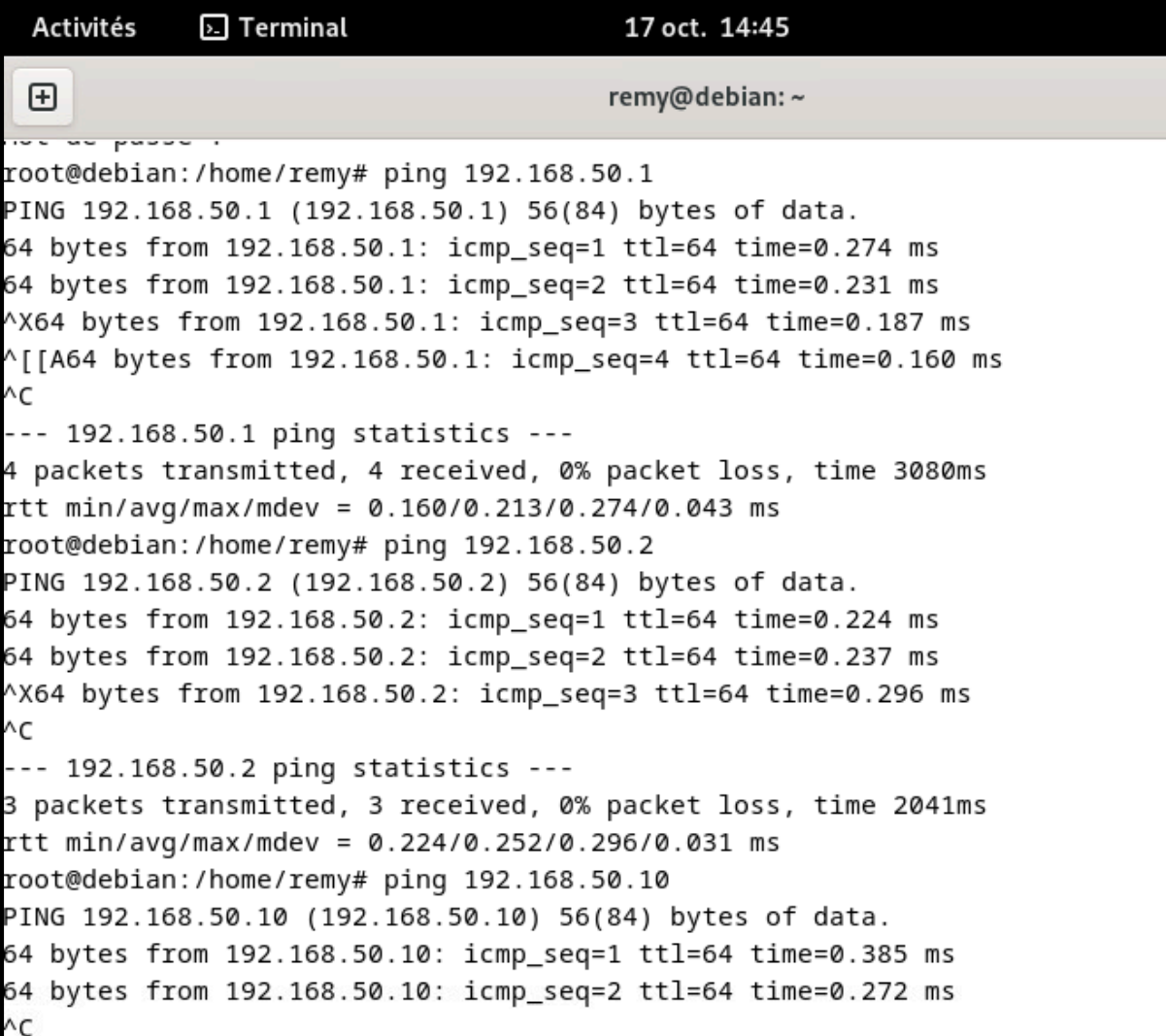
The image shows a terminal window titled 'SRV2 RB linux 12'. It shows the output of the 'cat /etc/ha.d/authkeys' command in a nano 7.2 editor. The configuration is as follows:

```
auth 1
1 md5 "123456"
2 crc
```

```
root@debian:/home/remy# nano /etc/ha.d/authkeys
root@debian:/home/remy# chmod 600 /etc/ha.d/authkeys
```

II) Test du Cluster avec la solution de haute disponibilité HeartBeat :

- Test de ping entre le nouveau client et le serveur SRV1 et SRV2 :



The screenshot shows a terminal window titled "Terminal" with the date and time "17 oct. 14:45". The user is logged in as "remy@debian: ~". The terminal output shows the following commands and results:

```
root@debian:/home/remy# ping 192.168.50.1
PING 192.168.50.1 (192.168.50.1) 56(84) bytes of data.
64 bytes from 192.168.50.1: icmp_seq=1 ttl=64 time=0.274 ms
64 bytes from 192.168.50.1: icmp_seq=2 ttl=64 time=0.231 ms
^X64 bytes from 192.168.50.1: icmp_seq=3 ttl=64 time=0.187 ms
^[[A64 bytes from 192.168.50.1: icmp_seq=4 ttl=64 time=0.160 ms
^C
--- 192.168.50.1 ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 3080ms
rtt min/avg/max/mdev = 0.160/0.213/0.274/0.043 ms
root@debian:/home/remy# ping 192.168.50.2
PING 192.168.50.2 (192.168.50.2) 56(84) bytes of data.
64 bytes from 192.168.50.2: icmp_seq=1 ttl=64 time=0.224 ms
64 bytes from 192.168.50.2: icmp_seq=2 ttl=64 time=0.237 ms
^X64 bytes from 192.168.50.2: icmp_seq=3 ttl=64 time=0.296 ms
^C
--- 192.168.50.2 ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 2041ms
rtt min/avg/max/mdev = 0.224/0.252/0.296/0.031 ms
root@debian:/home/remy# ping 192.168.50.10
PING 192.168.50.10 (192.168.50.10) 56(84) bytes of data.
64 bytes from 192.168.50.10: icmp_seq=1 ttl=64 time=0.385 ms
64 bytes from 192.168.50.10: icmp_seq=2 ttl=64 time=0.272 ms
^C
```

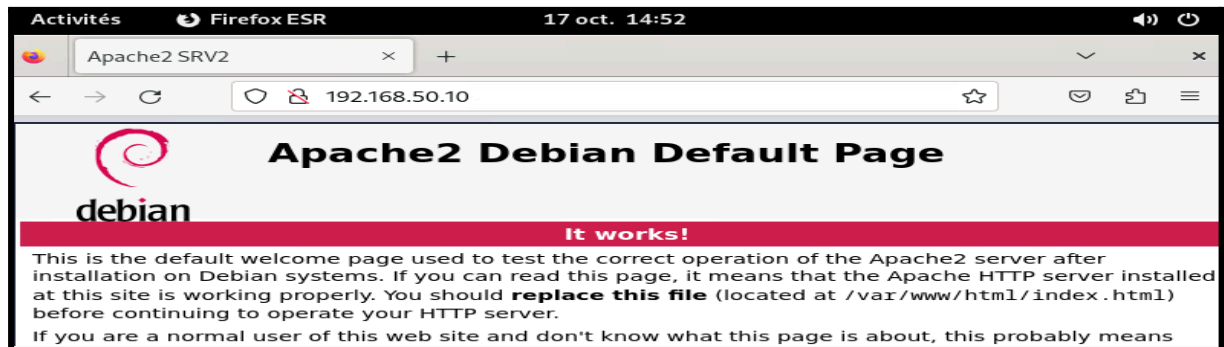
Test de ping entre le client et l'adresse Virtuel du cluster :

```
PING 192.168.50.10 (192.168.50.10) 56(84) bytes of data.  
64 bytes from 192.168.50.10: icmp_seq=1 ttl=64 time=0.385 ms  
64 bytes from 192.168.50.10: icmp_seq=2 ttl=64 time=0.272 ms  
^C  
--- 192.168.50.10 ping statistics ---  
2 packets transmitted, 2 received, 0% packet loss, time 1009ms  
rtt min/avg/max/mdev = 0.272/0.328/0.385/0.056 ms  
root@debian:/home/remy#
```

Simulation de la panne du serveur SRV 1 pour pouvoir alors observer que le cluster fonctionne correctement et que l'adresse IP virtuel attribuée au cluster redirige vers le serveur SRV 2.

```
root@SRV1:/home/remy# sudo systemctl stop networking
```

Après avoir mis le serveur SRV1 down on va alors essayer de se connecter au serveur en passant par l'adresse virtuel et nous obtenons au final une redirection vers le SRV2



III) Mise en place d'un Répartiteur avec IPVS

Définition :

Le répartiteur de charge, c'est comme un planificateur de tâches pour les serveurs. Il répartit le travail entre eux pour éviter qu'un seul serveur soit surchargé, assurant ainsi que les applications en ligne restent accessibles et rapides pour les utilisateurs. Si un serveur tombe en panne, le répartiteur redirige le trafic vers les autres serveurs pour éviter les interruptions de service.

Pour revenir à IPVS c'est à dire IP virtual service en informatique c'est une technologie de répartition de charge qui une fois configurée redirige selon les préférences de l'algorithme les paquets et donc le serveur choisit pour équilibrer les charges en fonction de la configuration choisie au préalable.

IPVS utilise des méthodes de répartition de charge telles que le "round-robin", la répartition basée sur la charge actuelle des serveurs, ou d'autres algorithmes selon la configuration. Comme vu dans le TP suivant.

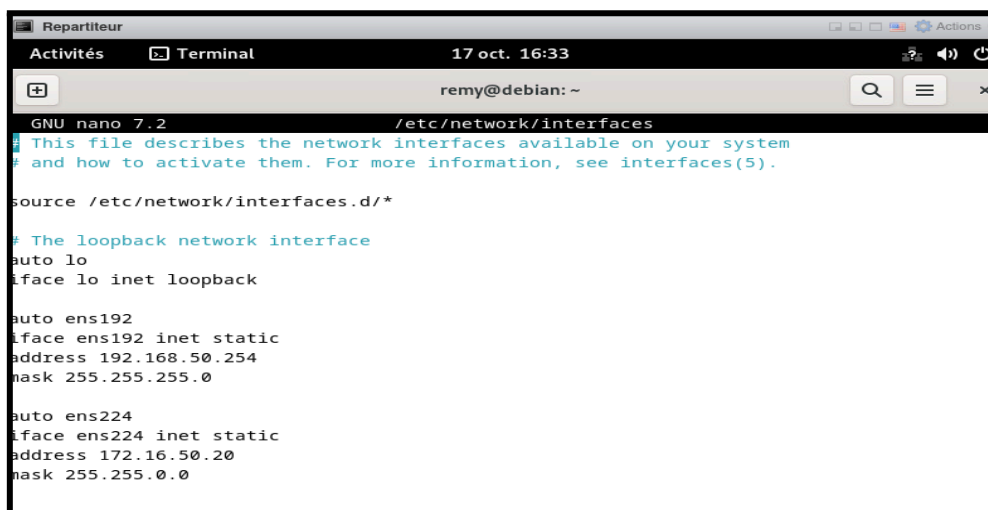
installation du package IPVS sur le répartiteur :

```

root@debian:/home/remy# apt-get install ipvsadm
Lecture des listes de paquets... Fait
Construction de l'arbre des dépendances... Fait
Lecture des informations d'état... Fait
ipvsadm est déjà la version la plus récente (1:1.31-1+b1).
0 mis à jour, 0 nouvellement installés, 0 à enlever et 10 non mis à jour.
root@debian:/home/remy# apt upgrade
Lecture des listes de paquets... Fait
Construction de l'arbre des dépendances... Fait

```

Répartiteur configuration des deux carte réseau :



```

Repartiteur
17 oct. 16:33
remy@debian: ~
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/*

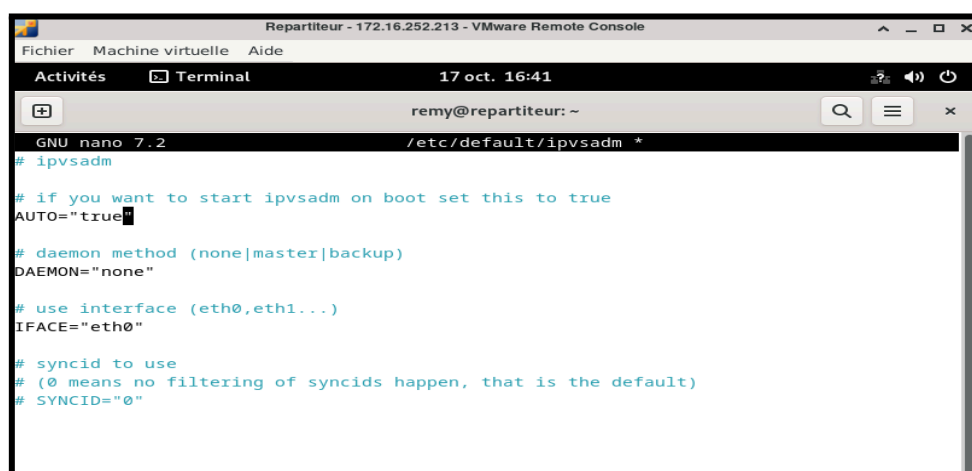
# The loopback network interface
auto lo
iface lo inet loopback

auto ens192
iface ens192 inet static
address 192.168.50.254
mask 255.255.255.0

auto ens224
iface ens224 inet static
address 172.16.50.20
mask 255.255.0.0

```

Nous éditons le fichier de configuration ipvs : on applique "true" pour "Auto" pour pouvoir répartir les requêtes .



```

Repartiteur - 172.16.252.213 - VMware Remote Console
17 oct. 16:41
remy@repartiteur: ~
GNU nano 7.2 /etc/default/ipvsadm *
# ipvsadm

# if you want to start ipvsadm on boot set this to true
AUTO="true"

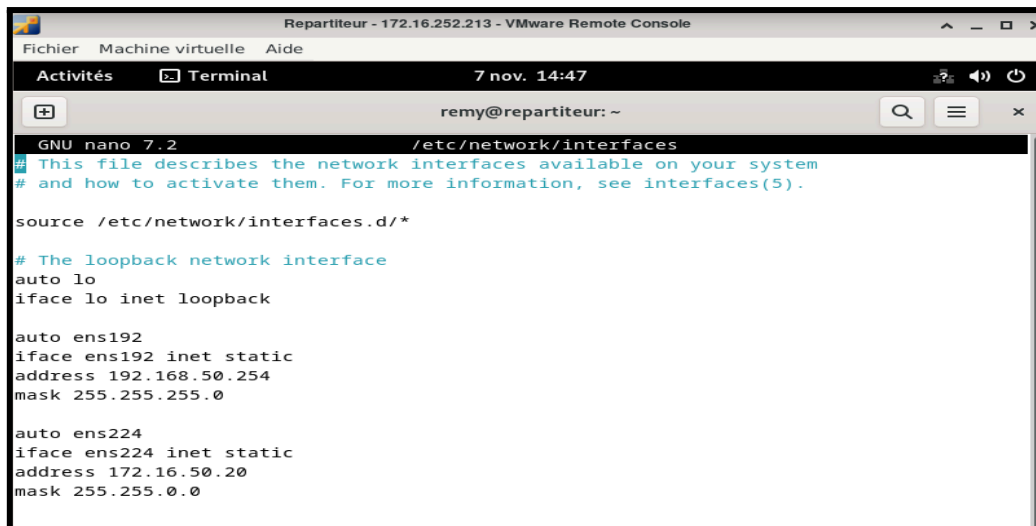
# daemon method (none|master|backup)
DAEMON="none"

# use interface (eth0,eth1...)
IFACE="eth0"

# syncid to use
# (0 means no filtering of syncids happen, that is the default)
# SYNCID="0"

```

Mise en place de la configuration des cartes réseau :



```
Repartiteur - 172.16.252.213 - VMware Remote Console
Fichier Machine virtuelle Aide
Activités Terminal 7 nov. 14:47
remy@repartiteur: ~
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).

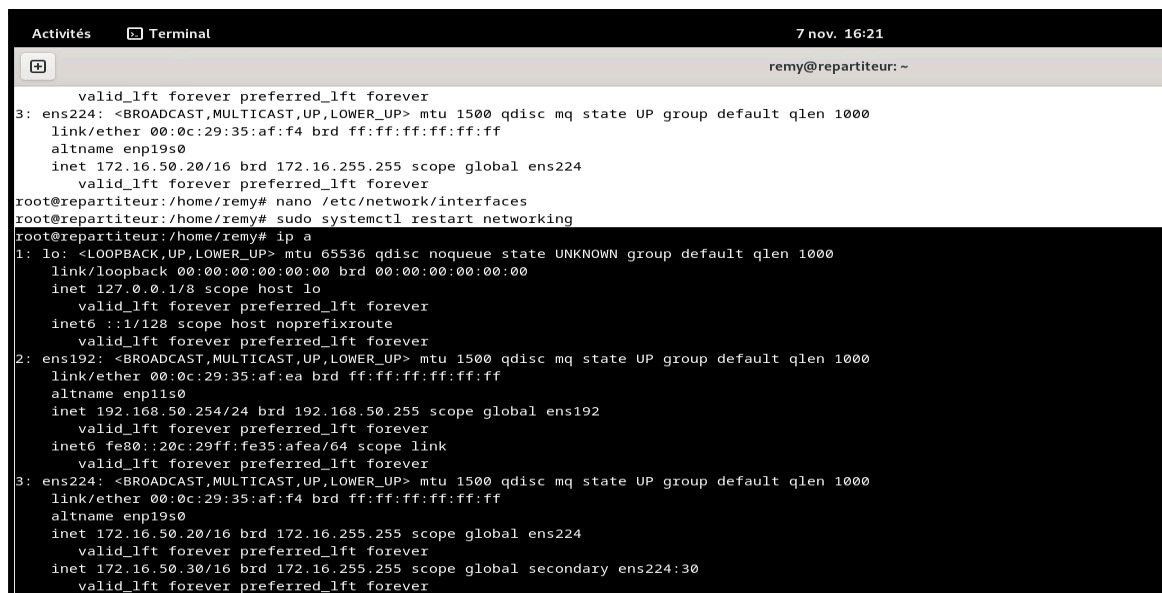
source /etc/network/interfaces.d/*

# The loopback network interface
auto lo
iface lo inet loopback

auto ens192
iface ens192 inet static
address 192.168.50.254
mask 255.255.255.0

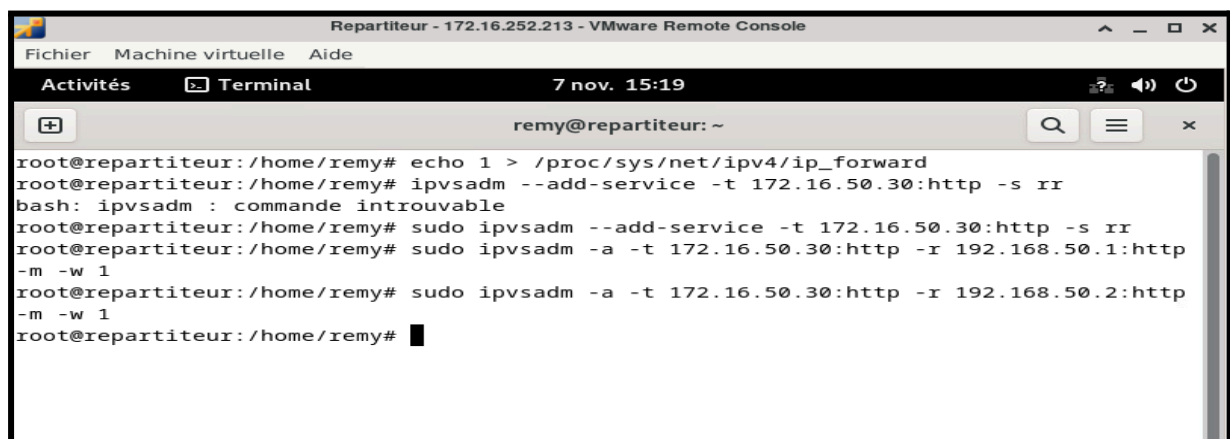
auto ens224
iface ens224 inet static
address 172.16.50.20
mask 255.255.0.0
```

Création de l'adresse virtuel ens224:30 pour le Load Balancer:



```
Activités Terminal 7 nov. 16:21
remy@repartiteur: ~
valid_lft forever preferred_lft forever
3: ens224: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc mq state UP group default qlen 1000
   link/ether 00:0c:29:35:af:f4 brd ff:ff:ff:ff:ff:ff
   altnam enp19s0
   inet 172.16.50.20/16 brd 172.16.255.255 scope global ens224
       valid_lft forever preferred_lft forever
root@repartiteur:/home/remy# nano /etc/network/interfaces
root@repartiteur:/home/remy# sudo systemctl restart networking
root@repartiteur:/home/remy# 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 mq state UP group default qlen 1000
   link/ether 00:0c:29:35:af:ea brd ff:ff:ff:ff:ff:ff
   altnam enp11s0
   inet 192.168.50.254/24 brd 192.168.50.255 scope global ens192
       valid_lft forever preferred_lft forever
   inet6 fe80::20c:29ff:fe35:afea/64 scope link
       valid_lft forever preferred_lft forever
3: ens224: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc mq state UP group default qlen 1000
   link/ether 00:0c:29:35:af:f4 brd ff:ff:ff:ff:ff:ff
   altnam enp19s0
   inet 172.16.50.20/16 brd 172.16.255.255 scope global ens224
       valid_lft forever preferred_lft forever
   inet 172.16.50.30/16 brd 172.16.255.255 scope global secondary ens224:30
       valid_lft forever preferred_lft forever
```

Création du cluster du Load balancing et mise en application des commandes suivantes pour permettre de faire connaître les adresses au service IPVS :



```
Repartiteur - 172.16.252.213 - VMware Remote Console
Fichier Machine virtuelle Aide
Activités Terminal 7 nov. 15:19
remy@repartiteur: ~
root@repartiteur:/home/remy# echo 1 > /proc/sys/net/ipv4/ip_forward
root@repartiteur:/home/remy# ipvsadm --add-service -t 172.16.50.30:http -s rr
bash: ipvsadm : commande introuvable
root@repartiteur:/home/remy# sudo ipvsadm --add-service -t 172.16.50.30:http -s rr
root@repartiteur:/home/remy# sudo ipvsadm -a -t 172.16.50.30:http -r 192.168.50.1:http
-m -w 1
root@repartiteur:/home/remy# sudo ipvsadm -a -t 172.16.50.30:http -r 192.168.50.2:http
-m -w 1
root@repartiteur:/home/remy#
```

Durant le paramétrage on configure alors le IPVS avec l'option " - rr " c'est-à-dire la méthode de round robin permet de répartir de manière aléatoire la charge entre les deux serveurs SRV1 et SRV2.

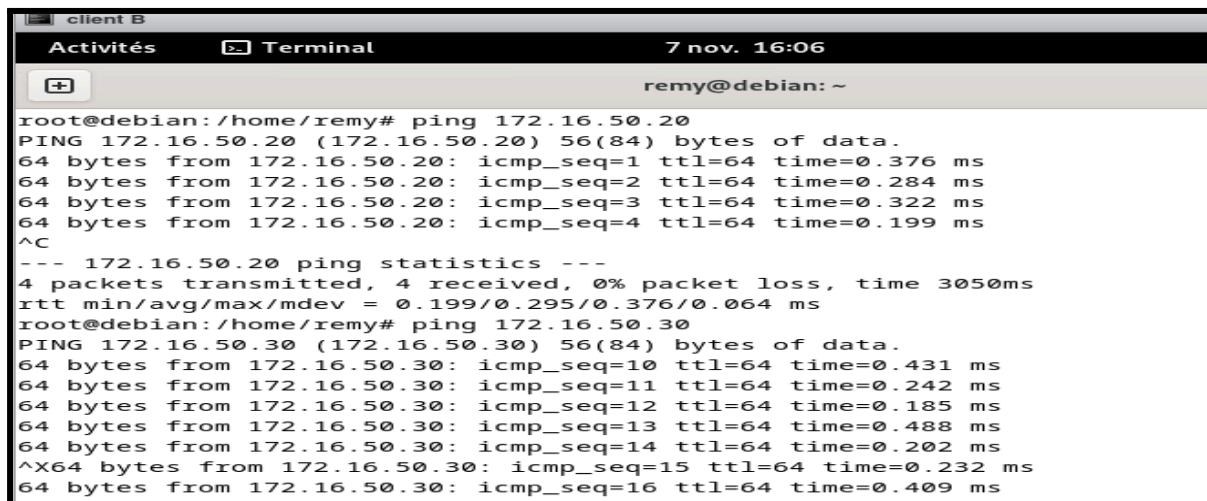
On utilise aussi l'option "-m" est utilisée pour masquer l'adresse du réseau derrière le répartiteur.

l'option "-w" permet de définir le poids à définir pour chaque serveur et leur répartition il est couplé à l'option "-rr" donc on utilise la commande "-wrr" nommée la méthode de "Weighted Round Robin"

Vérification du paramétrage IPVS mise en place pour le répartiteur :

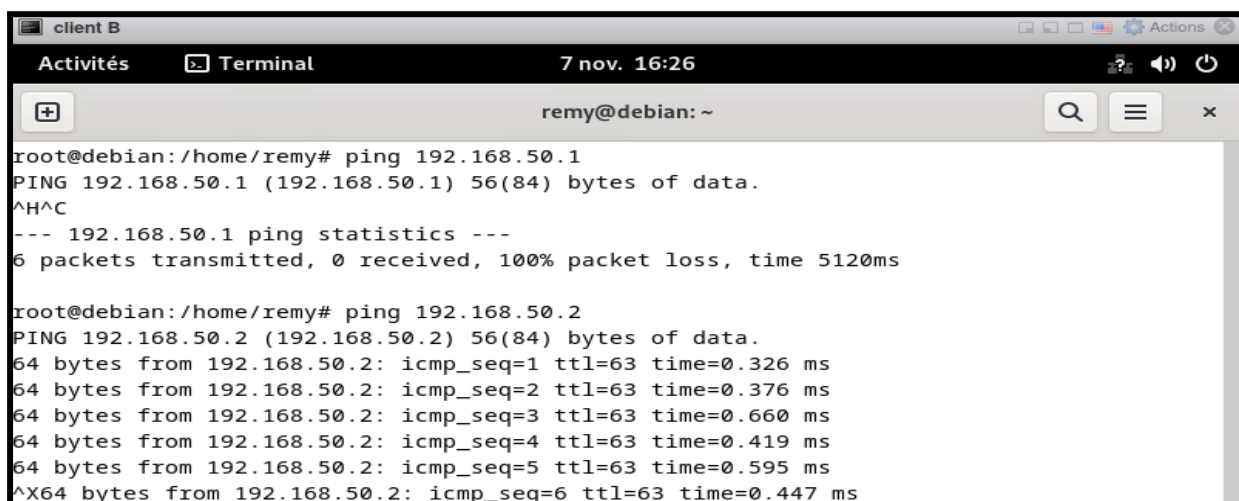
```
root@repartiteur:/home/remy# sudo ipvsadm -L
IP Virtual Server version 1.2.1 (size=4096)
Prot LocalAddress:Port Scheduler Flags
  -> RemoteAddress:Port          Forward Weight ActiveConn InActConn
TCP  172.16.50.30:http rr
  -> 192.168.50.1:http            Masq    1      0      0
  -> 192.168.50.2:http            Masq    1      0      0
```

Ping de mon client B en 172.16.50.15 vers mon cluster en 172.16.50.30 :



```
client B
7 nov. 16:06
remy@debian: ~
root@debian:/home/remy# ping 172.16.50.20
PING 172.16.50.20 (172.16.50.20) 56(84) bytes of data.
64 bytes from 172.16.50.20: icmp_seq=1 ttl=64 time=0.376 ms
64 bytes from 172.16.50.20: icmp_seq=2 ttl=64 time=0.284 ms
64 bytes from 172.16.50.20: icmp_seq=3 ttl=64 time=0.322 ms
64 bytes from 172.16.50.20: icmp_seq=4 ttl=64 time=0.199 ms
^C
--- 172.16.50.20 ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 3050ms
rtt min/avg/max/mdev = 0.199/0.295/0.376/0.064 ms
root@debian:/home/remy# ping 172.16.50.30
PING 172.16.50.30 (172.16.50.30) 56(84) bytes of data.
64 bytes from 172.16.50.30: icmp_seq=10 ttl=64 time=0.431 ms
64 bytes from 172.16.50.30: icmp_seq=11 ttl=64 time=0.242 ms
64 bytes from 172.16.50.30: icmp_seq=12 ttl=64 time=0.185 ms
64 bytes from 172.16.50.30: icmp_seq=13 ttl=64 time=0.488 ms
64 bytes from 172.16.50.30: icmp_seq=14 ttl=64 time=0.202 ms
^X64 bytes from 172.16.50.30: icmp_seq=15 ttl=64 time=0.232 ms
64 bytes from 172.16.50.30: icmp_seq=16 ttl=64 time=0.409 ms
```

Nous pouvons alors Ping avec le client B en 172.16.50.15 le SRV 1 en 192.168.50.2 en passant par le répartiteur ayant son adresse virtuel en 172.16.50.3.



```
client B
7 nov. 16:26
remy@debian: ~
root@debian:/home/remy# ping 192.168.50.1
PING 192.168.50.1 (192.168.50.1) 56(84) bytes of data.
^H^C
--- 192.168.50.1 ping statistics ---
6 packets transmitted, 0 received, 100% packet loss, time 5120ms

root@debian:/home/remy# ping 192.168.50.2
PING 192.168.50.2 (192.168.50.2) 56(84) bytes of data.
64 bytes from 192.168.50.2: icmp_seq=1 ttl=63 time=0.326 ms
64 bytes from 192.168.50.2: icmp_seq=2 ttl=63 time=0.376 ms
64 bytes from 192.168.50.2: icmp_seq=3 ttl=63 time=0.660 ms
64 bytes from 192.168.50.2: icmp_seq=4 ttl=63 time=0.419 ms
64 bytes from 192.168.50.2: icmp_seq=5 ttl=63 time=0.595 ms
^X64 bytes from 192.168.50.2: icmp_seq=6 ttl=63 time=0.447 ms
```

Test de l'URL avec le client B nous pouvons alors voir que le SRV2 est utilisé par la répartition de charge et le SRV 1 est inactif.

The screenshot shows two windows side-by-side. The left window is a terminal titled 'repartiteur' with the user 'remy@repartiteur:~'. It displays the output of the command 'ipvsadm -ln', showing a load balancer configuration for port 80. The right window is a web browser titled 'client B' showing the 'Apache2 Debian Default Page' at the URL '172.16.50.30'. The page includes the Debian logo, the text 'It works!', and a 'Configuration Overview' section.

Terminal Output:

```
Toutes les 1,0s: ipvsadm -ln repartiteur: Thu Nov 9 20:12:12 2023
IP Virtual Server version 1.2.1 (size=4096)
Prot LocalAddress:Port Scheduler Flags
  -> RemoteAddress:Port Forward Weight ActiveConn InActConn
TCP 172.16.50.30:80 rr
  -> 192.168.50.1:80 Masq 1 0 0
  -> 192.168.50.2:80 Masq 1 0 1
```

Web Browser Content:

Apache2 Debian Default Page

debian

It works!

This is the default welcome page used to test the correct operation of the Apache2 server after installation on Debian systems. 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

Debian's Apache2 default configuration is different from the upstream default configuration, and split into several files optimized for interaction with Debian 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 Debian systems is as follows:

`/etc/apache2/`