



## Table des matières

Qu'est-ce que Hearbeat ? .....	3
Qu'est ce que le Load Balancing ? .....	3
Schéma de notre réseau :.....	4
Configuration des serveurs web1 et web2 .....	4
Installation de Apache sur web1 et web2 .....	5
Installation et configuration de Heartbeat sur web1 et web2.....	7
Test de Hearbeat .....	10
Configuration des serveurs web1 et web2 .....	13
Configuration de Load Balancing sur lb1 .....	15
Installation et configuration de ipvsadm sur lb1 .....	16
Configuration de lb1 et lb2 sur du Load Balancing .....	20
Installation de Heartbeat sur lb1 et lb2 .....	20

## Qu'est-ce que Heartbeat ?

En résumé, le concept de "heartbeat" en informatique repose sur l'idée d'échanger des signaux périodiques entre les différents composants d'un système afin de surveiller leur état de fonctionnement et de prendre des mesures appropriées en cas de défaillance. Cela contribue à garantir la disponibilité, la fiabilité et la stabilité des systèmes informatiques.

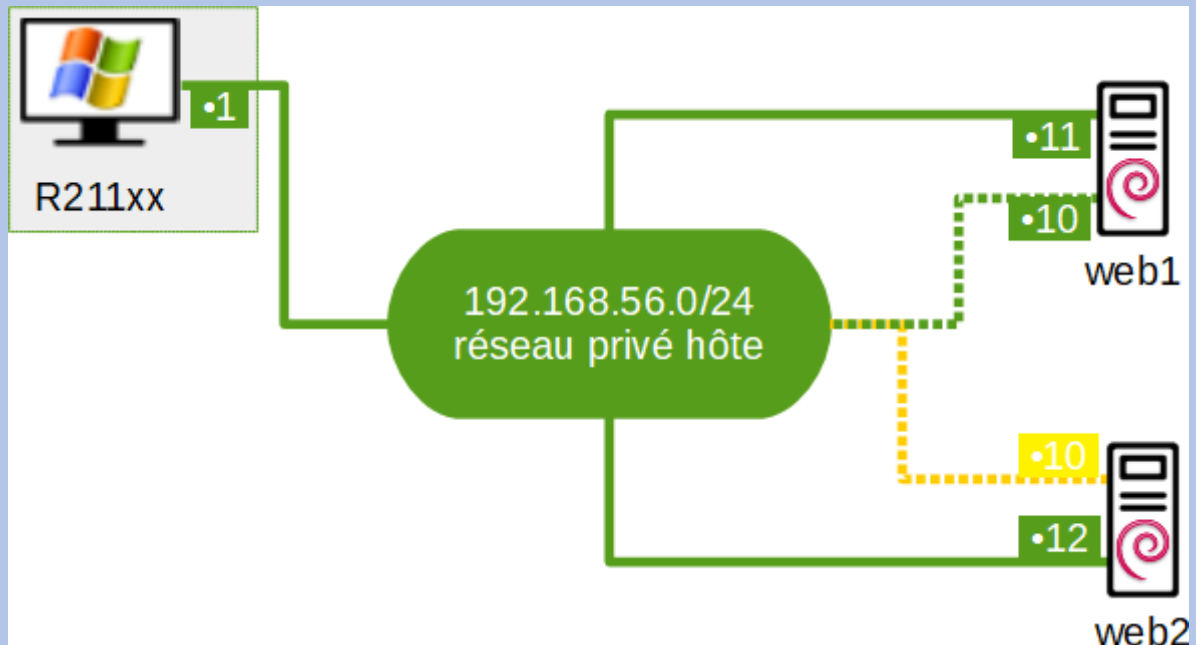
## Qu'est ce que le Load Balancing ?

Le "load balancing" (équilibrage de charge en français) est une technique utilisée dans les réseaux informatiques et les systèmes distribués pour distribuer la charge de travail de manière équitable entre plusieurs composants, tels que des serveurs, afin d'optimiser les performances, d'assurer la disponibilité et d'éviter les surcharges.

Le but principal du load balancing est de garantir que toutes les ressources disponibles sont utilisées de manière efficace, d'éviter la congestion sur certains composants du système, et de fournir une répartition équitable du trafic entre les différents serveurs. Cela améliore la capacité de traitement, la stabilité et la fiabilité du système. Voici quelques points clés liés au load balancing :

1. **Répartition du trafic** : Le load balancer distribue le trafic entrant entre plusieurs serveurs en fonction de différents critères, tels que la charge actuelle de chaque serveur, sa disponibilité, ou d'autres métriques pertinentes.
2. **Amélioration des performances** : En distribuant la charge de manière équilibrée, le load balancing permet d'exploiter efficacement les ressources disponibles, évitant ainsi la surcharge d'un serveur spécifique et garantissant des temps de réponse plus rapides pour les utilisateurs.
3. **Disponibilité accrue** : En redirigeant automatiquement le trafic vers des serveurs disponibles en cas de panne ou de défaillance d'un serveur, le load balancing contribue à maintenir la disponibilité du service sans interruption notable.
4. **Scalabilité** : Le load balancing facilite l'ajout de nouveaux serveurs au fur et à mesure que la demande de services augmente, ce qui permet de faire évoluer l'infrastructure de manière transparente.
5. **Gestion de la charge dynamique** : Certains systèmes de load balancing peuvent ajuster dynamiquement la répartition de la charge en fonction des changements de conditions du système, tels que des variations de trafic ou des modifications de la capacité des serveurs.

Schéma de notre réseau :



## Configuration des serveurs web1 et web2

Dans un premier temps, nous allons configurer l'adressage ip de nos deux serveurs soit web1 et web2.

Web 1 :

```
GNU nano 3.2
# This file describes the network interfaces available on your system
# and how to activate them. For more information, see the man pages
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 192.168.56.11/24
    gateway 192.168.56.254
```

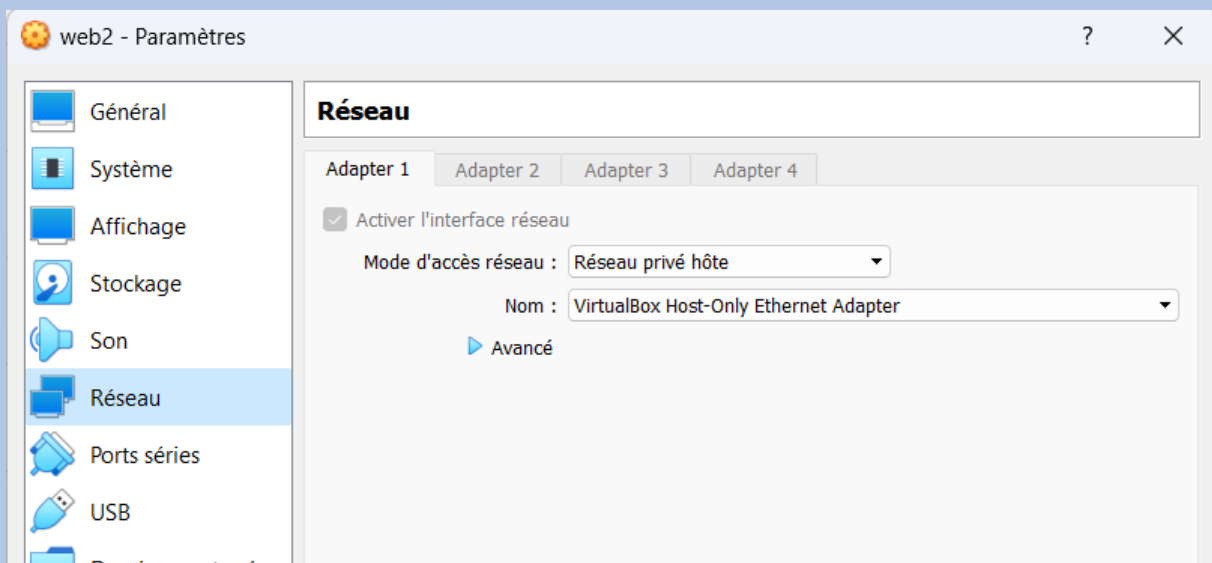
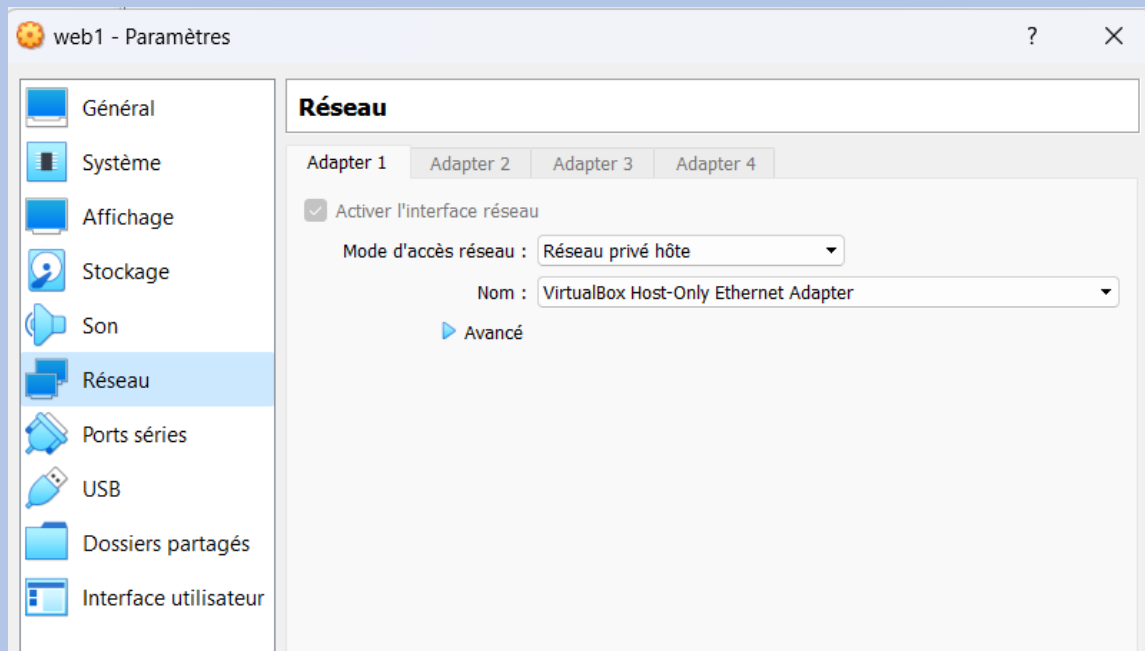
Web 2 :

```
GNU nano 3.2 /etc/network/interfaces
# This file describes the network interfaces available on your system
# and how to activate them. For more information, see the man pages
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 192.168.56.12/24
    gateway 192.168.56.254
```

Puis vérifions leurs cartes réseaux sur VirtualBox.



## Installation de Apache sur web1 et web2

Avant toute installation, nous mettons à jour les paquets :

```
root@buster:~# apt update
```

Puis pouvons continuer avec l'installation de Apache 2 :

```
root@buster:~# apt install apache2_
```

Vérifions qu'Apache 2 est bien actif :

## Okan

```
root@buster:~# systemctl status apache2
• apache2.service - The Apache HTTP Server
  Loaded: loaded (/lib/systemd/system/apache2.service; enabled; vendor preset: enabled)
  Active: active (running) since Wed 2023-12-13 17:44:12 CET; 12s ago
    Docs: https://httpd.apache.org/docs/2.4/
 Main PID: 1070 (apache2)
   Tasks: 55 (limit: 1149)
  Memory: 9.5M
   CGroup: /system.slice/apache2.service
           └─1070 /usr/sbin/apache2 -k start
             └─1072 /usr/sbin/apache2 -k start
               └─1073 /usr/sbin/apache2 -k start

déc. 13 17:44:12 buster systemd[1]: Starting The Apache HTTP Server...
déc. 13 17:44:12 buster apachectl[1059]: AH00558: apache2: Could not reliably determine the se
déc. 13 17:44:12 buster systemd[1]: Started The Apache HTTP Server.
lines 1-15/15 (END)
```

Après avoir vérifié qu'Apache 2 fonctionne bien, nous pouvons rentrer l'adresse ip de nos deux serveurs web dans la barre de recherche du navigateur de notre machine hôte



Faire la même chose pour l'autre serveur web.



## Installation et configuration de Heartbeat sur web1 et web2

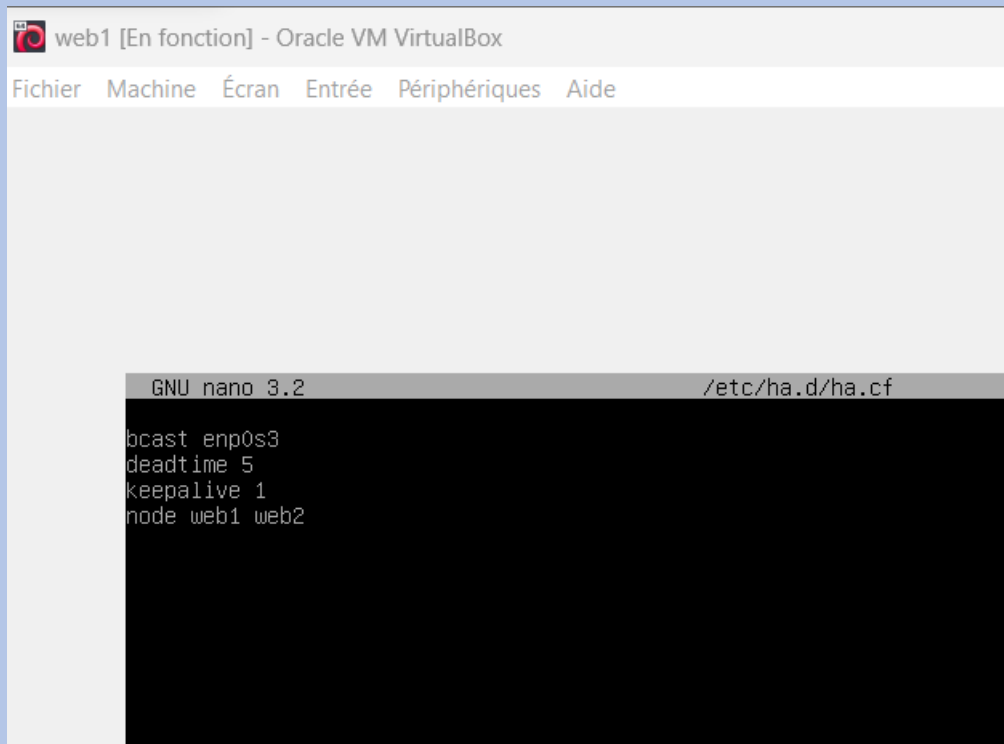
Nous allons maintenant installer heartbeat sur nos deux serveurs webs :

```
root@buster:~# apt install heartbeat_
```

Nous poursuivons avec la création d'un fichier qu'on nomme « ha.cf »

```
root@buster:~# nano /etc/ha.d/ha.cf
```

Après avoir bien créer le fichier de configuration ci-dessus, nous allons y rentrer en tout premier l'interface réseau soit enp0s3 dans notre cas, puis le temps mort du nœud qu'on fixera a 5 secondes, l'intervalle entre 2 battements de cœur qu'on fixe a 1 secondes et enfin le nœud c'est-à-dire les machines utilisées pour la haute disponibilité.

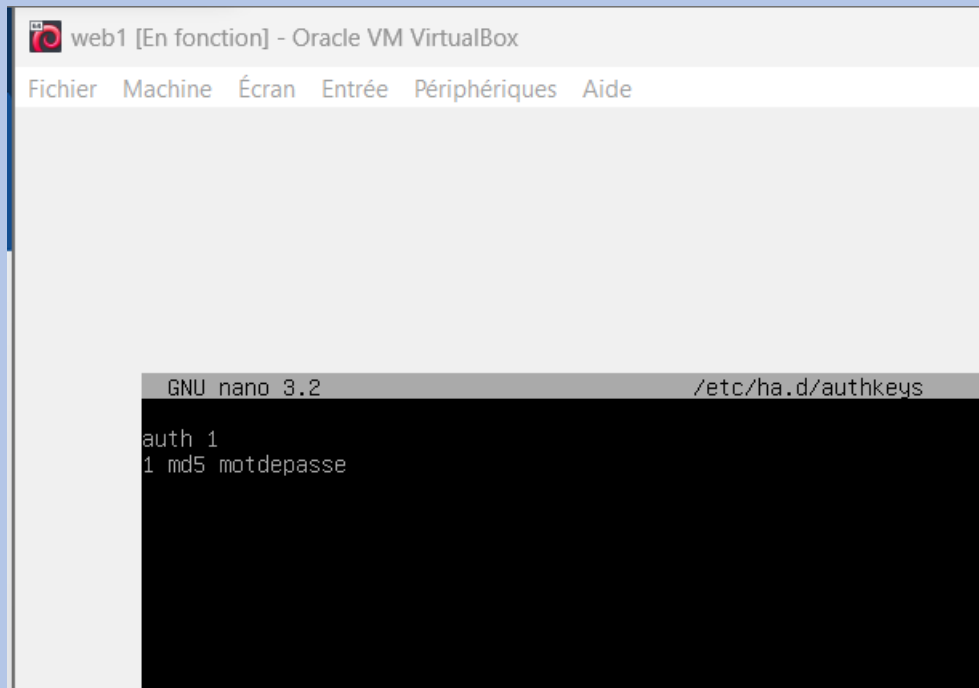


```
web1 [En fonction] - Oracle VM VirtualBox
Fichier  Machine  Écran  Entrée  Périphériques  Aide

GNU nano 3.2 /etc/ha.d/ha.cf
bcast enp0s3
deadtime 5
keepalive 1
node web1 web2
```

Nous créons maintenant le fichier « authkeys » toujours dans le même répertoire.

Nous y rentrons la clé partagée entre les serveurs ainsi que le protocole de protection.



The screenshot shows a terminal window titled "web1 [En fonction] - Oracle VM VirtualBox". The window has a menu bar with "Fichier", "Machine", "Écran", "Entrée", "Périphériques", and "Aide". The terminal content shows the GNU nano 3.2 editor editing the file /etc/ha.d/authkeys. The file content is as follows:

```
auth 1
1 md5 motdepasse
```

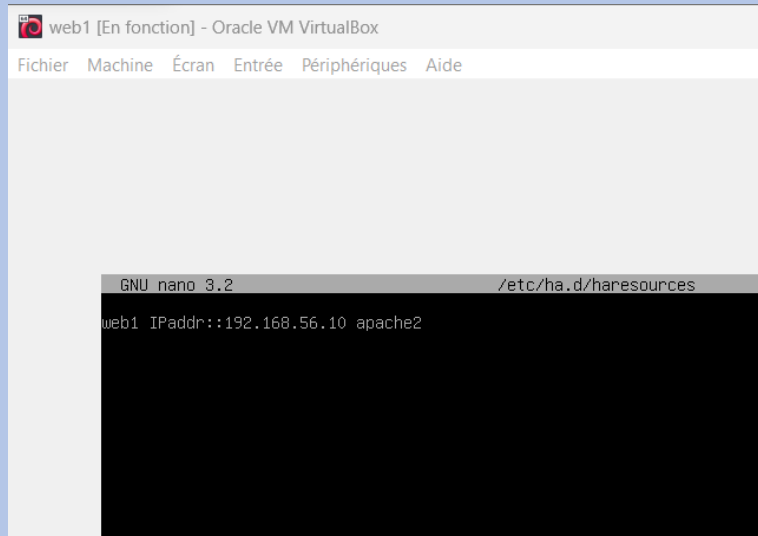
Le service heartbeat exige une protection de ce fichier, pour cela nous lui mettons le droit de lecture et écriture seulement à l'utilisateur :

```
root@buster:~# chmod 600 /etc/ha.d/authkeys
```

Et enfin le 3eme et dernier fichier a créé est le fichier « haresources » où nous rentrons l'adresse virtuelle utilisée sur notre réseau.

Ici le serveur web1 est maître.



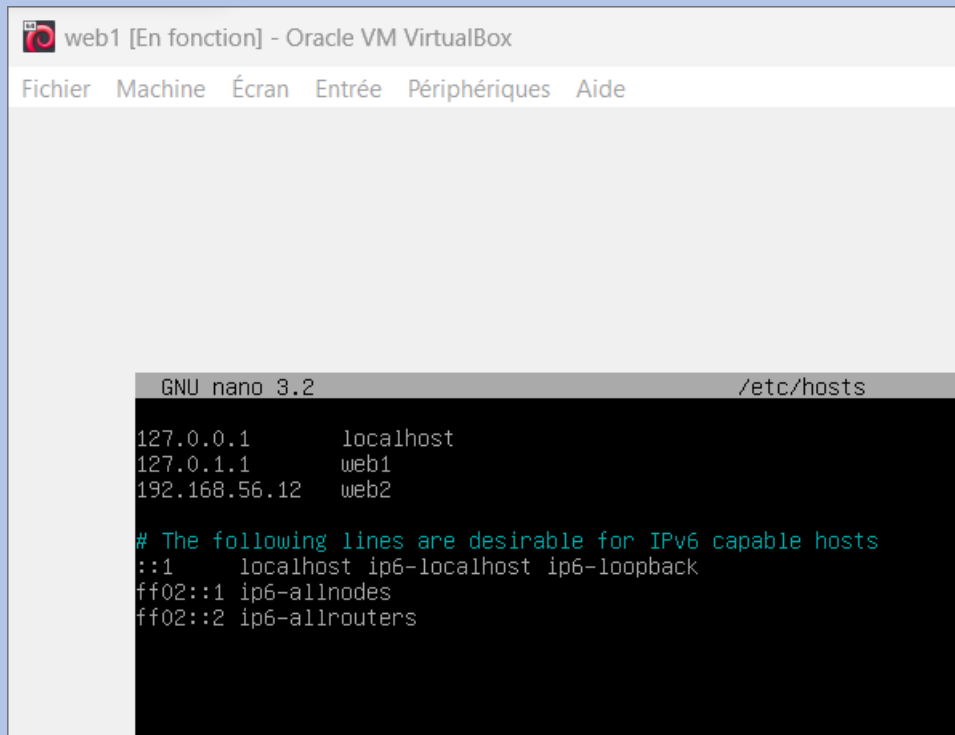


```
web1 [En fonction] - Oracle VM VirtualBox
Fichier Machine Écran Entrée Périphériques Aide

GNU nano 3.2 /etc/ha.d/haresources
web1 IPaddr::192.168.56.10 apache2
```

Il faut bien évidemment réaliser chacune des manipulations ci-dessus pour web2 aussi.

En nous rendant dans le fichier `/etc/hosts`, nous allons donc déclarer web1 et web2.

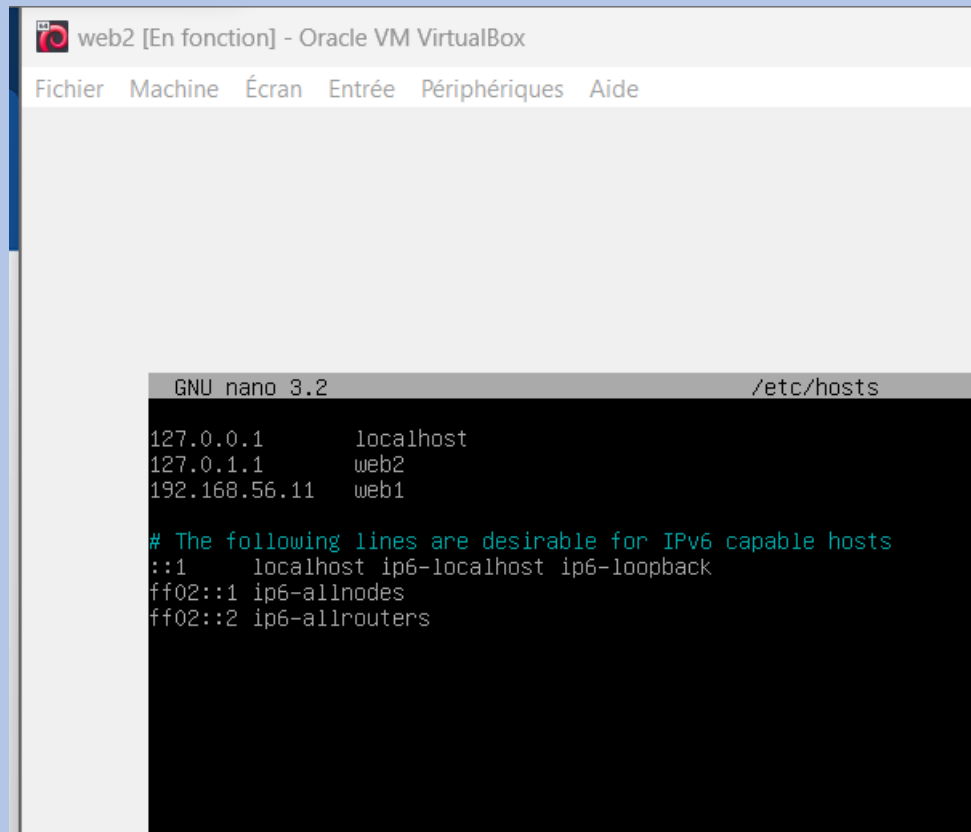


```
web1 [En fonction] - Oracle VM VirtualBox
Fichier Machine Écran Entrée Périphériques Aide

GNU nano 3.2 /etc/hosts
127.0.0.1    localhost
127.0.1.1    web1
192.168.56.12 web2

# The following lines are desirable for IPv6 capable hosts
::1        localhost ip6-localhost ip6-loopback
ff02::1    ip6-allnodes
ff02::2    ip6-allrouters
```

Faire la même chose sur web2.



```
web2 [En fonction] - Oracle VM VirtualBox
Fichier  Machine  Écran  Entrée  Périphériques  Aide

GNU nano 3.2 /etc/hosts
127.0.0.1    localhost
127.0.1.1    web2
192.168.56.11 web1

# The following lines are desirable for IPv6 capable hosts
::1        localhost ip6-localhost ip6-loopback
ff02::1    ip6-allnodes
ff02::2    ip6-allrouters
```

## Test de Hearbeat

Pour pouvoir nous repérer durant la phase de test, il faut modifier le code html des pages par défaut pour web1 et web2.

Nous allons stopper apache2 sur les deux serveurs webs :

```
root@web1:~# systemctl stop apache2.service
```

```
root@web2:~# systemctl stop apache2.service
```

Nous allons maintenant le désactiver complètement car Heartbeat le démarrera lui-même.

```
root@web1:~# systemctl disable apache2.service
```

```
root@web2:~# systemctl disable apache2.service
```

Et enfin redémarrons Heartbeat sur les deux serveurs webs :

```
root@web1:~# systemctl restart heartbeat.service
```

```
root@web2:~# systemctl restart heartbeat.service
```

Vérifions que Heartbeat est bien actif après son redémarrage.

## Okan

```

root@web1:~# systemctl restart heartbeat.service
root@web1:~# systemctl status heartbeat.service
• heartbeat.service - Heartbeat High Availability Cluster Communication and Membership
   Loaded: loaded (/lib/systemd/system/heartbeat.service; enabled; vendor preset: enabled)
   Active: active (running) since Wed 2023-12-13 18:06:53 CET; 54s ago
     Docs: man:heartbeat(8)
           http://www.linux-ha.org/wiki/Documentation
  Main PID: 1124 (heartbeat)
    Tasks: 4 (limit: 1149)
   Memory: 10.5M
   CGroup: /system.slice/heartbeat.service
           └─1124 heartbeat: master control process
             └─1127 heartbeat: FIFO reader
               └─1128 heartbeat: write: bcast enp0s3
                 └─1129 heartbeat: read: bcast enp0s3

```

```

root@web2:~# systemctl status heartbeat.service
• heartbeat.service - Heartbeat High Availability Cluster Communication and Membership
   Loaded: loaded (/lib/systemd/system/heartbeat.service; enabled; vendor preset: enabled)
   Active: active (running) since Wed 2023-12-13 18:07:06 CET; 1min 51s ago
     Docs: man:heartbeat(8)
           http://www.linux-ha.org/wiki/Documentation
  Main PID: 1139 (heartbeat)
    Tasks: 4 (limit: 1149)
   Memory: 10.2M
   CGroup: /system.slice/heartbeat.service
           └─1139 heartbeat: master control process
             └─1142 heartbeat: FIFO reader
               └─1143 heartbeat: write: bcast enp0s3
                 └─1144 heartbeat: read: bcast enp0s3

```

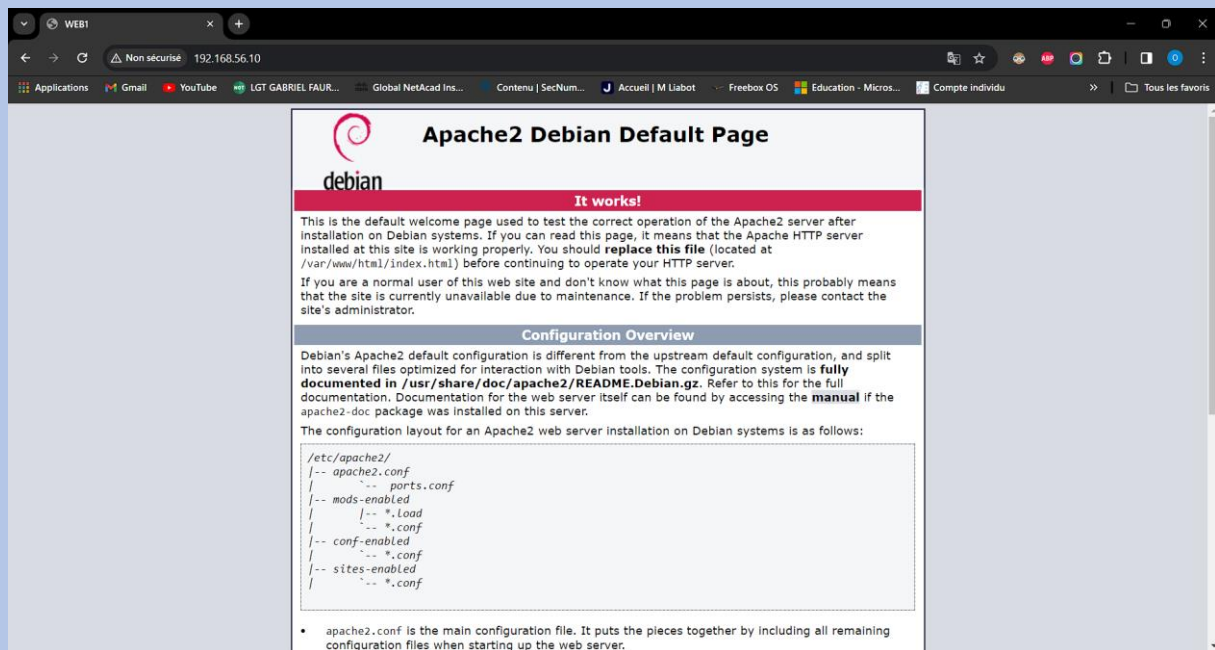
En exécutant un « ip a », nous pouvons voir que l'adresse ip virtuelle « 192.168.56.10 » est bien présente.

```

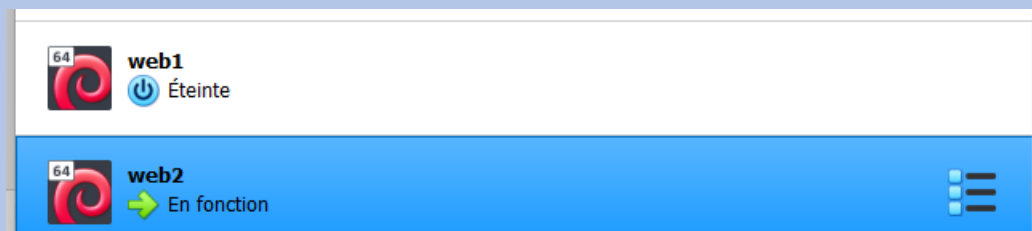
    valid_lft forever preferred_lft forever
2: enp0s3: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state UP group
000
    link/ether 08:00:27:a7:60:5f brd ff:ff:ff:ff:ff:ff
    inet 192.168.56.11/24 brd 192.168.56.255 scope global enp0s3
        valid_lft forever preferred_lft forever
    inet 192.168.56.10/24 brd 192.168.56.255 scope global secondary enp0s3:0
        valid_lft forever preferred_lft forever
    inet6 fe80::a00:27ff:fea7:605f/64 scope link
        valid_lft forever preferred_lft forever
root@web1:~# _

```

Nous allons nous connecter à la page d'apache par défaut avec l'adresse ip virtuelle.



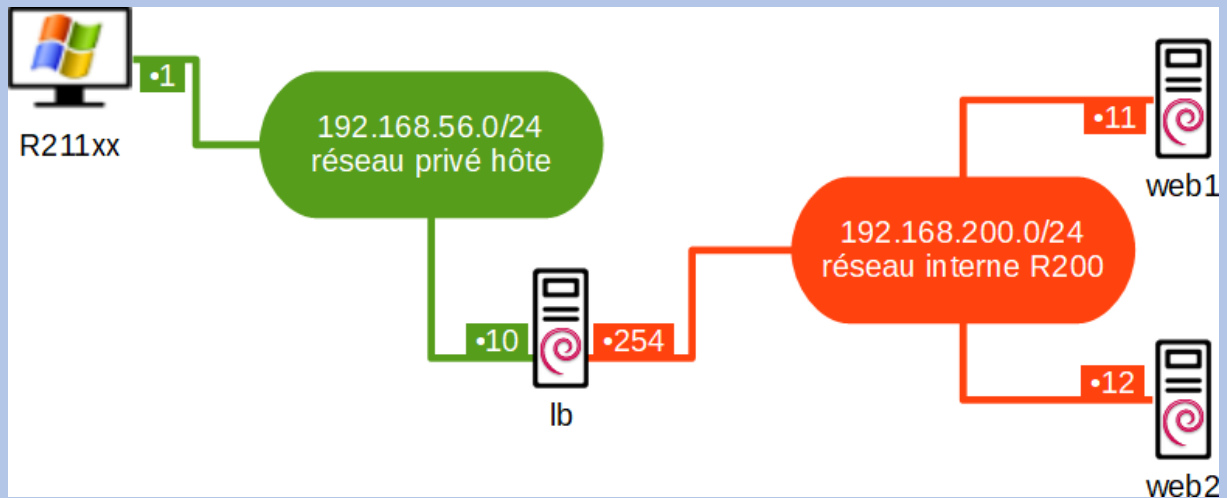
En éteignant web1, nous allons voir que web2 devrait prendre le relais.



C'est bien le cas, web2 prend bien le relais.

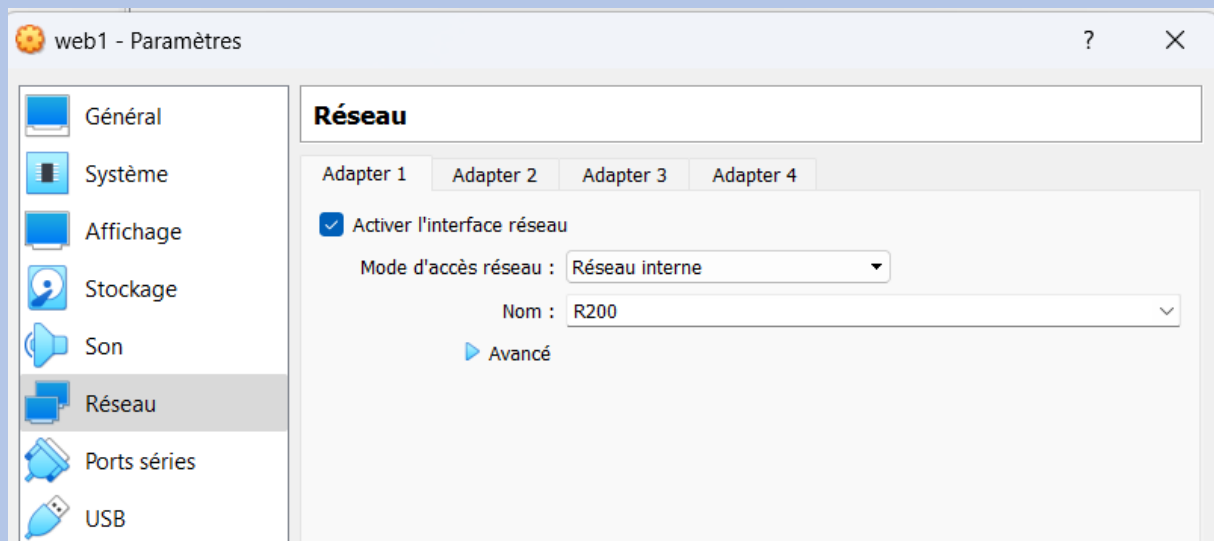


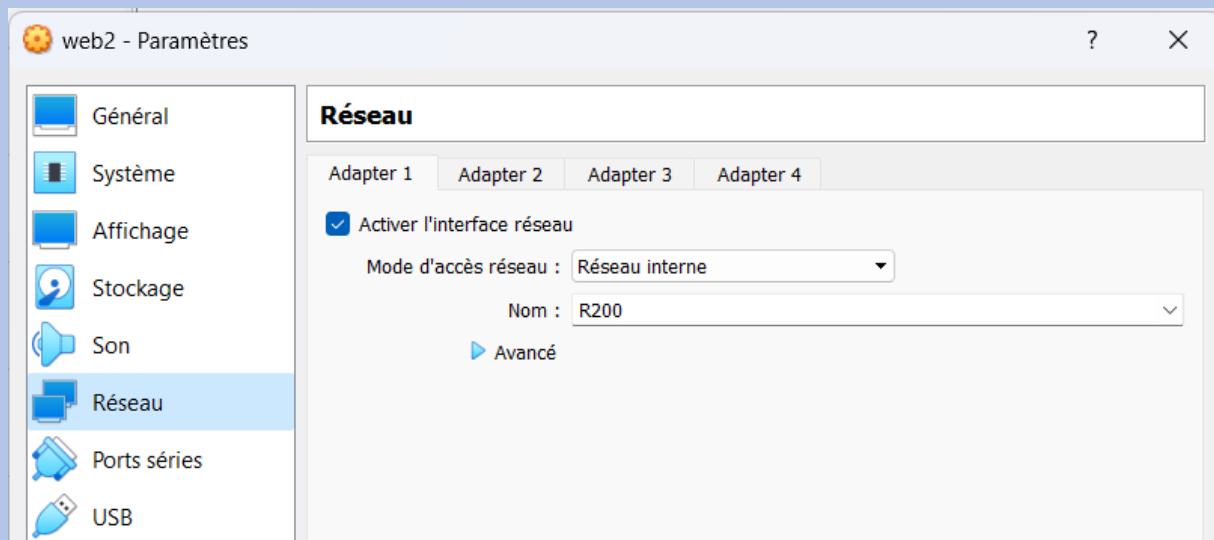
Schéma de notre réseau :



## Configuration des serveurs web1 et web2

Nous configurons web1 et web2 en sorte qu'ils soient en réseau interne.





Passons à la configuration de l'adressage ip de nos serveurs webs.

```
web1 [En fonction] - Oracle VM VirtualBox
Fichier Machine Écran Entrée Périphériques Aide

GNU nano 3.2 /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 192.168.200.11/24
    gateway 192.168.200.254
```

```
web2 [En fonction] - Oracle VM VirtualBox
Fichier Machine Écran Entrée Périphériques Aide

GNU nano 3.2 /etc/network/interfaces
# This file describes the network interfaces available on your system
# and how to activate them. For more information, see interfaces(5).

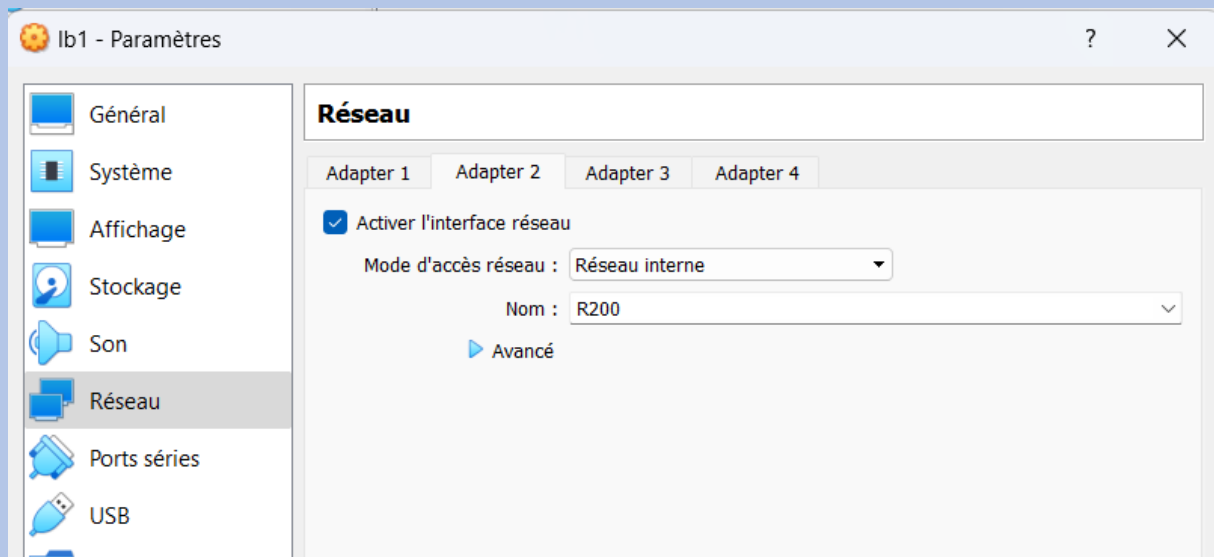
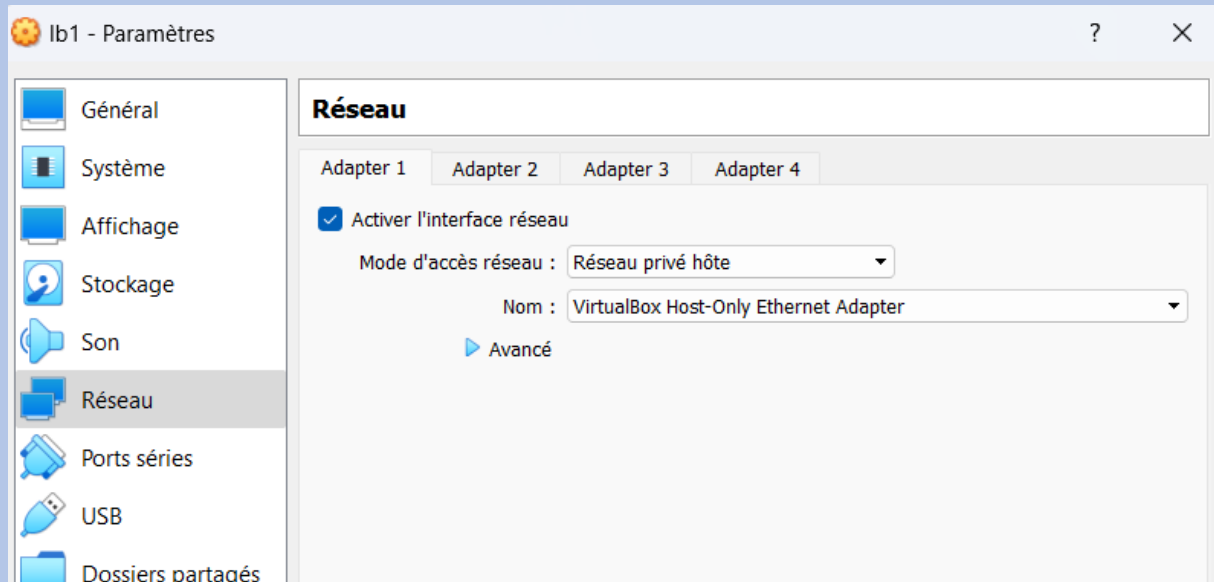
source /etc/network/interfaces.d/*

# The loopback network interface
auto lo
iface lo inet loopback

# The primary network interface
allow-hotplug enp0s3
iface enp0s3 inet static
    address 192.168.200.12/24
    gateway 192.168.200.254
```

## Configuration de Load Balancing sur lb1

Nous allons tout d'abord configurer les cartes réseaux de notre lb1.



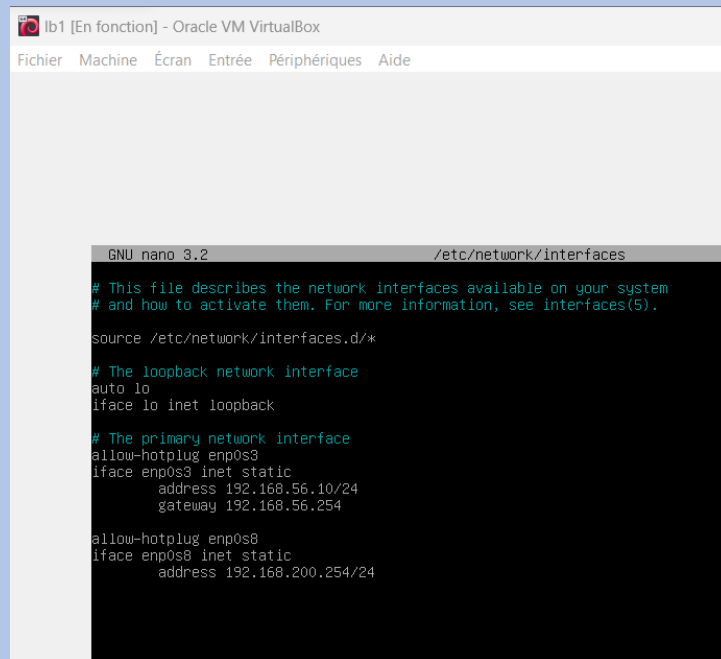
Okan

## Installation et configuration de ipvsadm sur lb1

Nous allons installer le répartiteur de charge « ipvsadm ».

```
root@buster:~# apt install ipvsadm
```

Nous allons maintenant configurer les deux interfaces sur le lb1.



```
GNU nano 3.2 /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/*

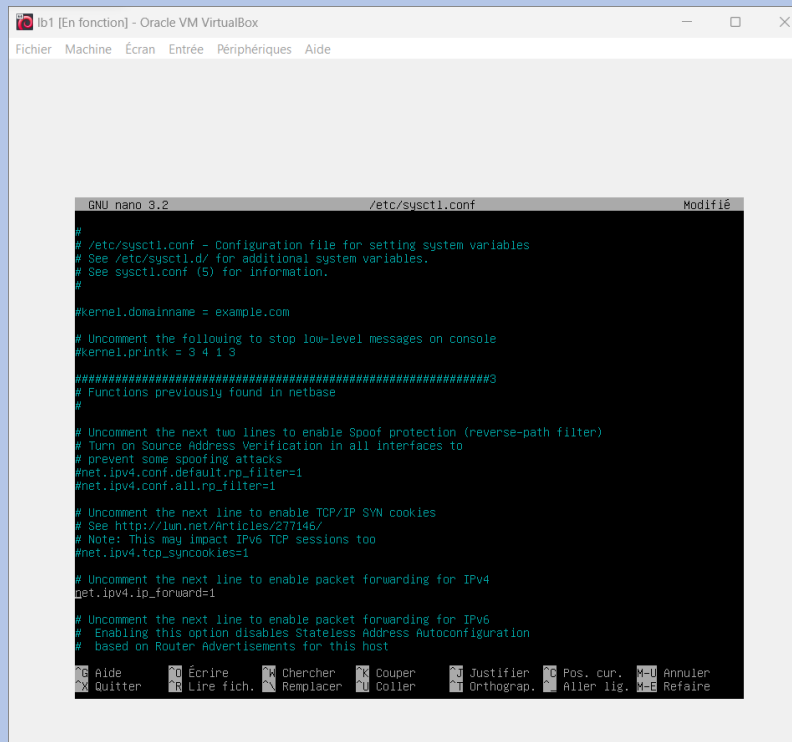
# The loopback network interface
auto lo
iface lo inet loopback

# The primary network interface
allow-hotplug enp0s3
iface enp0s3 inet static
    address 192.168.56.10/24
    gateway 192.168.56.254

allow-hotplug enp0s8
iface enp0s8 inet static
    address 192.168.200.254/24
```

Puis, activons le routage en supprimant le # (qui représente un commentaire) dans le fichier de configuration /etc/sysctl.conf.





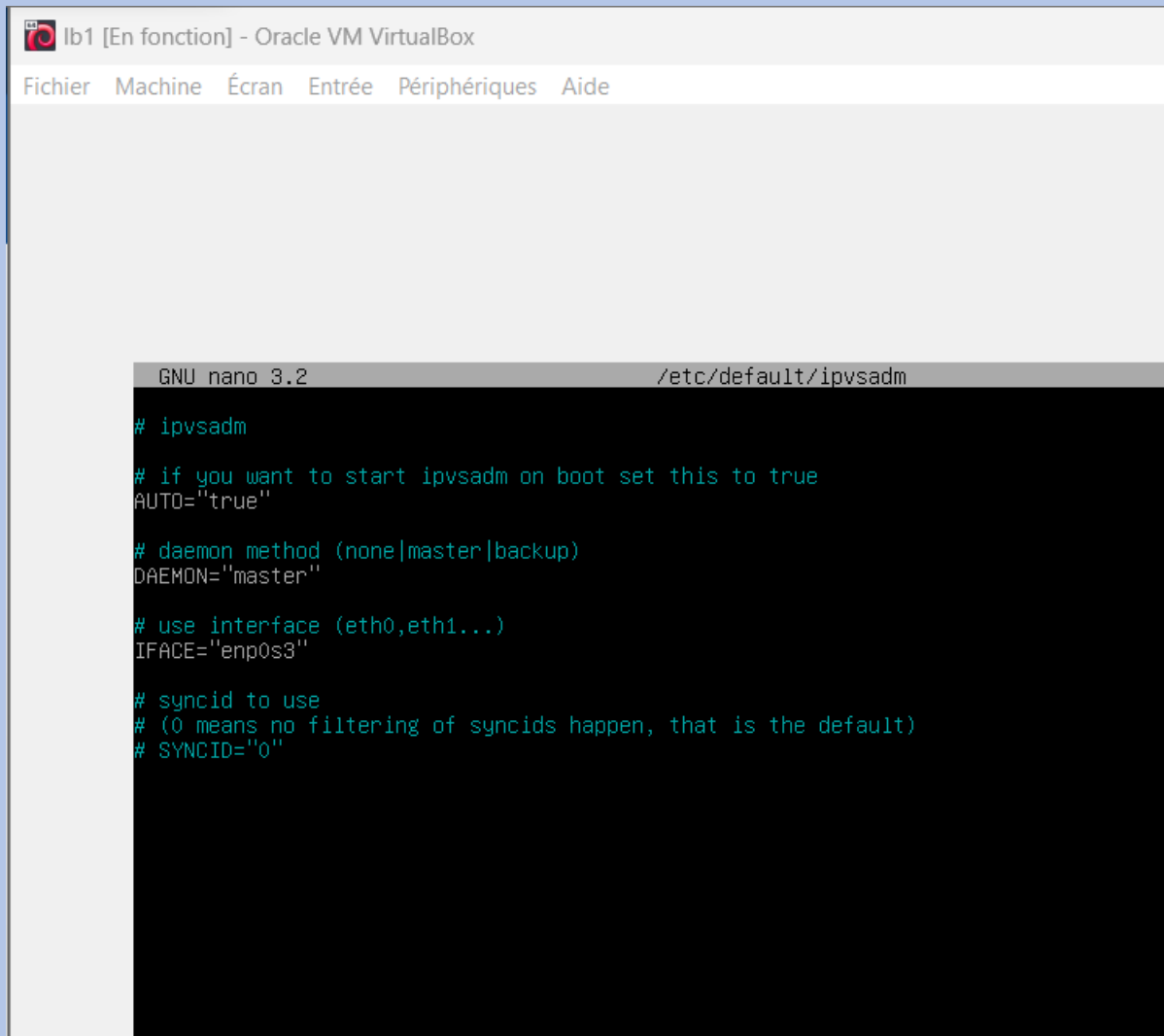
```
GNU nano 3.2 /etc/sysctl.conf Modifié
#
# /etc/sysctl.conf - Configuration file for setting system variables
# See /etc/sysctl.d/ for additional system variables.
# See sysctl.conf(5) for information.
#
#kernel.domainname = example.com
# Uncomment the following to stop low-level messages on console
#kernel.printk = 3 4 1 3
#####
# Functions previously found in netbase
#
# Uncomment the next two lines to enable Spoof protection (reverse-path filter)
# Turn on Source Address Verification in all interfaces to
# prevent some spoofing attacks
#net.ipv4.conf.default.rp_filter=1
#net.ipv4.conf.all.rp_filter=1
#
# Uncomment the next line to enable TCP/IP SYN cookies
# See http://lun.net/Articles/277146/
# Note: This may impact IPv6 TCP sessions too
#net.ipv4.tcp_syncookies=1
#
# Uncomment the next line to enable packet forwarding for IPv4
net.ipv4.ip_forward=1
#
# Uncomment the next line to enable packet forwarding for IPv6
# Enabling this option disables Stateless Address Autoconfiguration
# based on Router Advertisements for this host
```

Vérifions que le routage est bien actif.

```
root@buster:~# cat /proc/sys/net/ipv4/ip_forward
1
```

En nous rendant dans le fichier de configuration d'ipvsadm ci-dessous, nous allons le mettre en « maître » et rentrer l'interface où les requêtes arrivent.

Okan



The screenshot shows a VirtualBox window titled 'lb1 [En fonction] - Oracle VM VirtualBox'. The menu bar includes 'Fichier', 'Machine', 'Écran', 'Entrée', 'Périphériques', and 'Aide'. The main area displays a terminal window running 'GNU nano 3.2' editing '/etc/default/ipvsadm'. The file content is as follows:

```
# ipvsadm

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

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

# use interface (eth0,eth1...)
IFACE="enp0s3"

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

Et enfin, configurons le dernier fichier :

```
# Définition du service
ipvsadm -A -t 192.168.56.10:80 -s rr

# Membres du clusters
ipvsadm -a -t 192.168.56.10:80 -r 192.168.200.11:80 -m
ipvsadm -a -t 192.168.56.10:80 -r 192.168.200.12:80 -m
```

Vérifions avec la commande suivante le paramétrage effectué.

```
root@buster:~# ipvsadm -ln
IP Virtual Server version 1.2.1 (size=4096)
Prot LocalAddress:Port Scheduler Flags
  -> RemoteAddress:Port      Forward Weight ActiveConn InActConn
TCP  192.168.56.10:80 rr
  -> 192.168.200.11:80        Masq    1      0          0
  -> 192.168.200.12:80        Masq    1      0          0
```

Connectons-nous à la page par défaut d'apache avec l'adresse ip virtuelle.

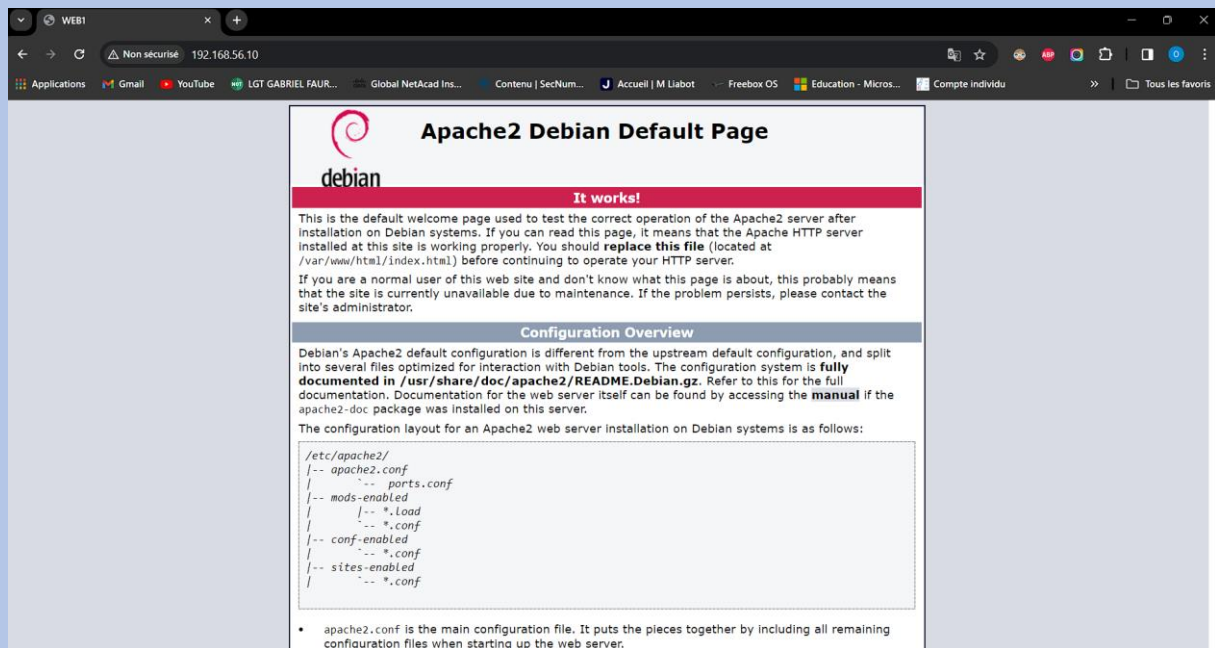
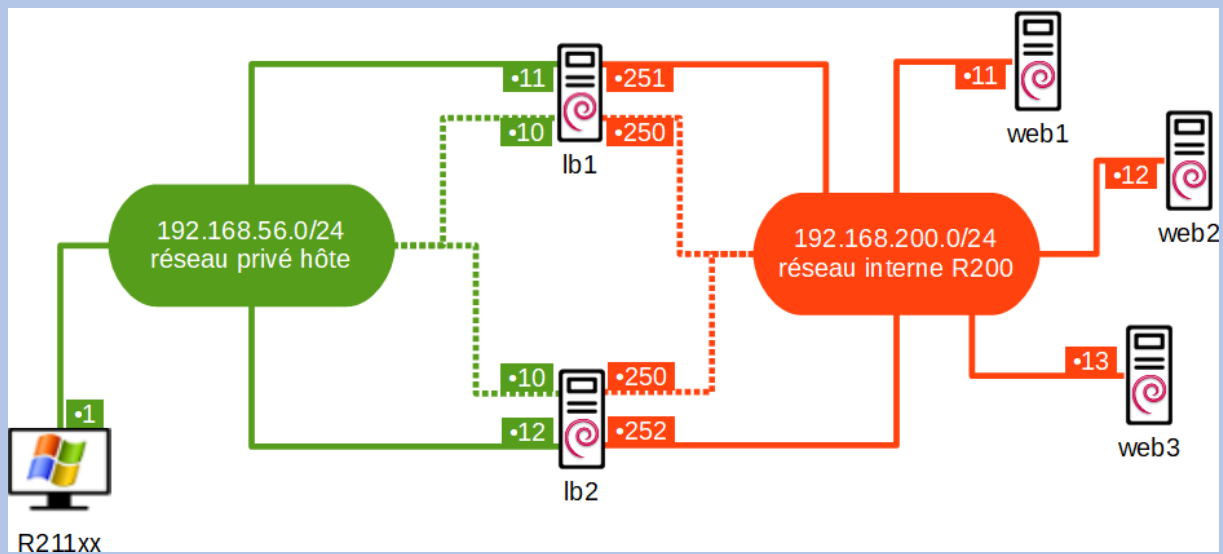


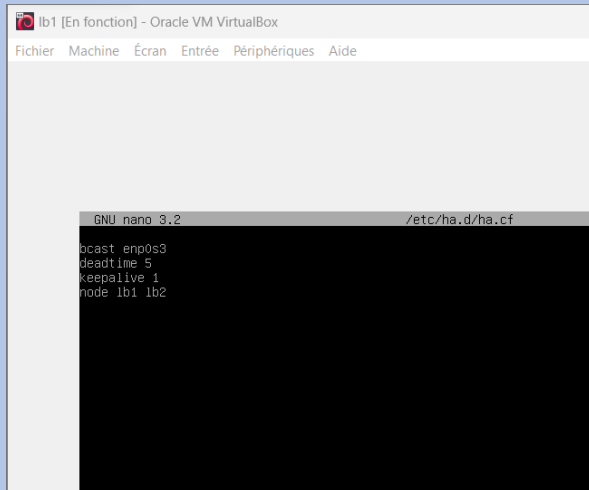
Schéma de notre réseau :



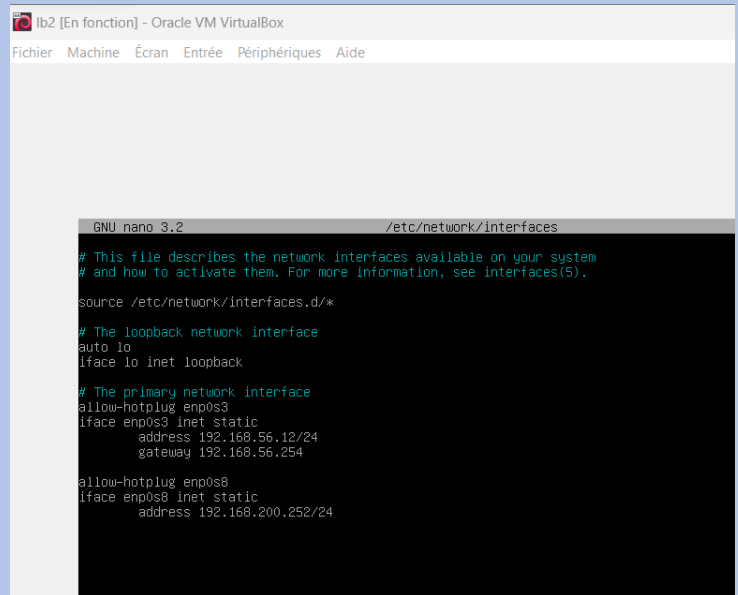
## Okan

## Configuration de lb1 et lb2 sur du Load Balancing

Nous configurons tout d'abord l'adressage ip de notre lb1, puis de notre lb2.



```
GNU nano 3.2 /etc/ha.d/ha.cf
#broadcast enp0s3
#deadtime 5
#keepalive 1
node lb1 lb2
```



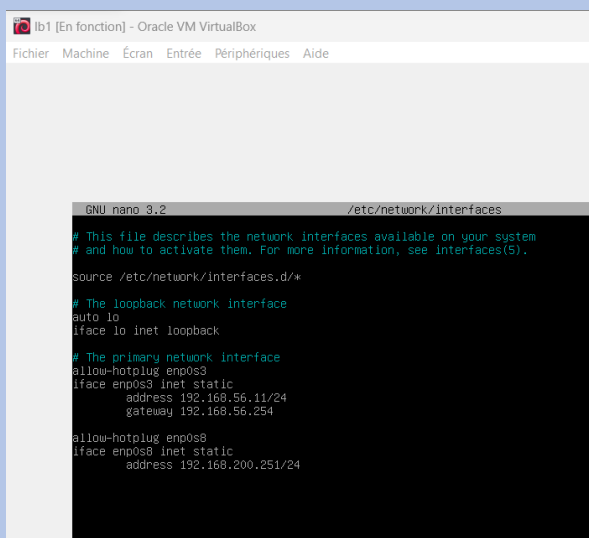
```
GNU nano 3.2 /etc/network/interfaces
# This file describes the network interfaces available on your system
# and how to activate them. For more information, see interfaces(5).

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 192.168.56.12/24
    gateway 192.168.56.254

allow-hotplug enp0s8
iface enp0s8 inet static
    address 192.168.200.252/24
```



```
GNU nano 3.2 /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 192.168.56.11/24
    gateway 192.168.56.254

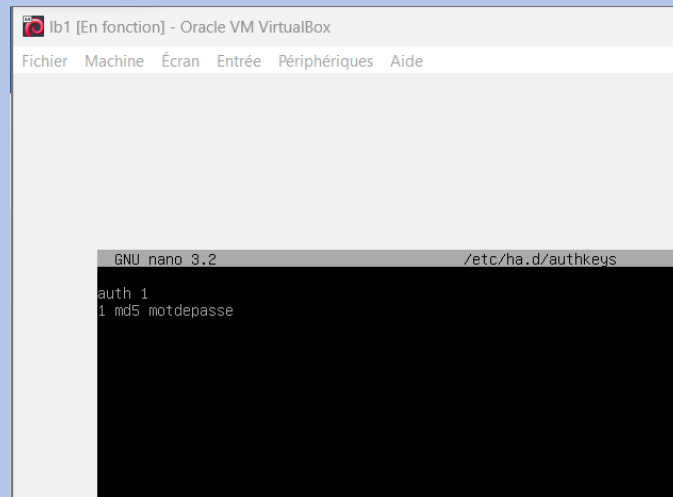
allow-hotplug enp0s8
iface enp0s8 inet static
    address 192.168.200.251/24
```

## Installation de Heartbeat sur lb1 et lb2

Installons Heartbeat sur lb1 et lb2.

```
root@buster:~# apt install heartbeat_
```

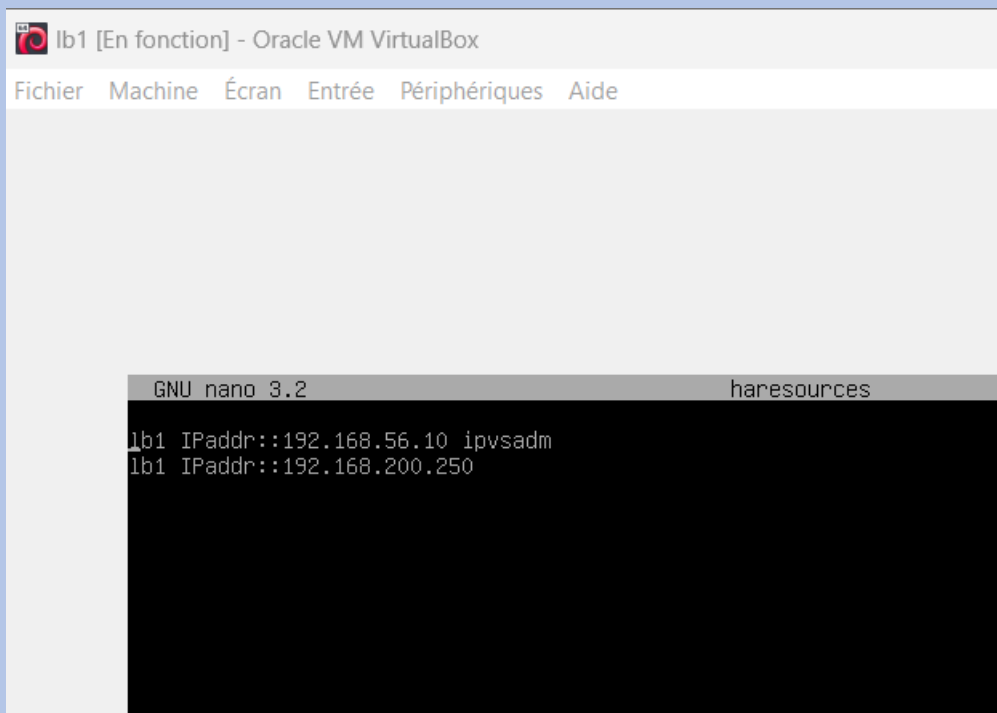
Comme fait précédemment, nous configurons les 3 fichiers de configuration d'Heartbeat.



```
GNU nano 3.2 /etc/ha.d/authkeys
auth 1
1 md5 motdepasse
```

Faire les mêmes manipulations pour lb2 notamment au niveau de l'installation d'ipvsadm.

Bien créer l'ip flottante à mettre dans le fichier de configuration dans lb1 et lb2.



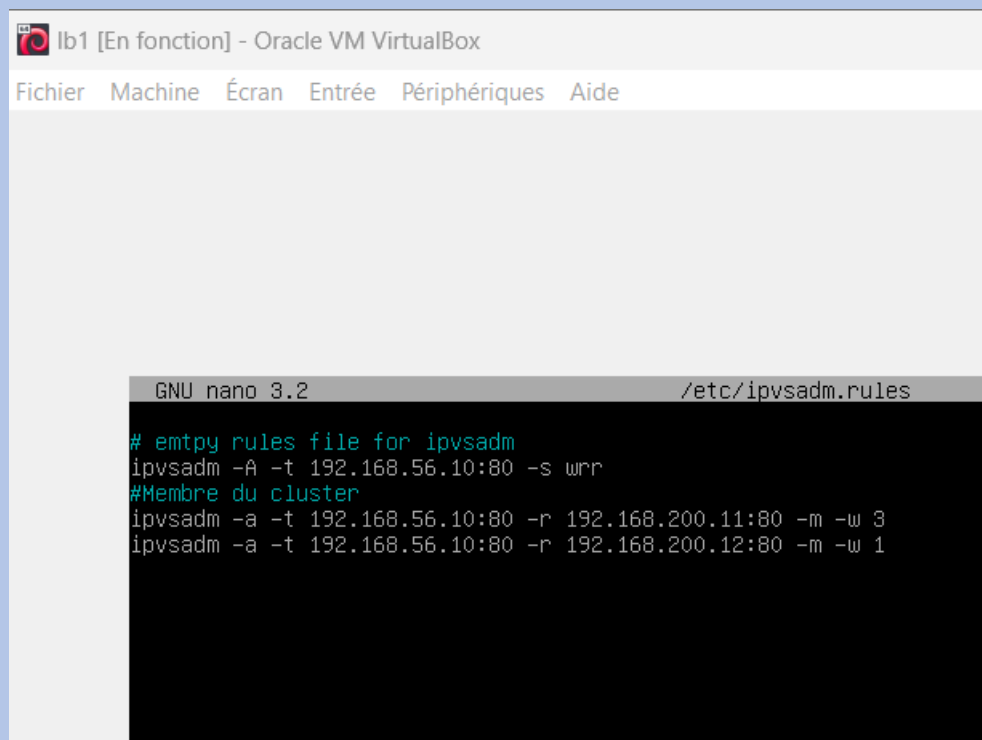
```
GNU nano 3.2 haresources
lb1 IPaddr::192.168.56.10 ipvsadm
lb1 IPaddr::192.168.200.250
```

Après avoir bien configuré heartbeat et ipvsadm, nous pouvons voir que l'ip virtuelle apparaît bien.

## Okan

```
valid_lft forever preferred_lft forever
2: enp0s3: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state UP group default qlen 1000
    link/ether 08:00:27:f0:ed:8c brd ff:ff:ff:ff:ff:ff
    inet 192.168.56.11/24 brd 192.168.56.255 scope global enp0s3
        valid_lft forever preferred_lft forever
    inet 192.168.56.10/24 brd 192.168.56.255 scope global secondary enp0s3:0
        valid_lft forever preferred_lft forever
    inet6 fe80::a00:27ff:fe0:ed8c/64 scope link
        valid_lft forever preferred_lft forever
3: enp0s8: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state UP group default qlen 1000
    link/ether 08:00:27:c2:98:98 brd ff:ff:ff:ff:ff:ff
    inet 192.168.200.251/24 brd 192.168.200.255 scope global enp0s8
        valid_lft forever preferred_lft forever
    inet 192.168.200.250/24 brd 192.168.200.255 scope global secondary enp0s8:0
        valid_lft forever preferred_lft forever
    inet6 fe80::a00:27ff:fec2:9898/64 scope link
        valid_lft forever preferred_lft forever
root@lb1:~#
```

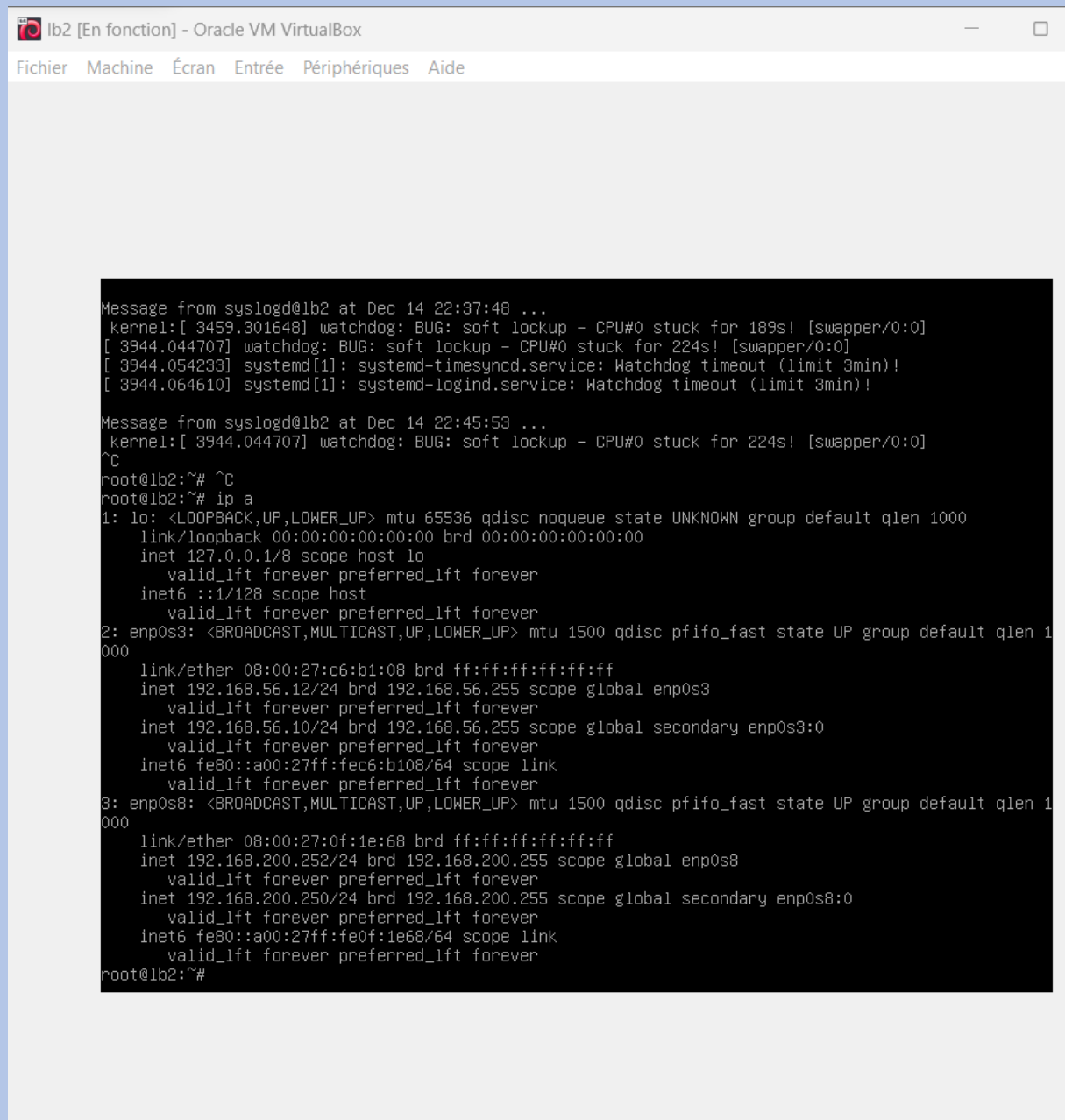
Pour la répartition de charges sur les serveurs web, il faut éditer le fichier de configuration ipvsadm.rules :



The screenshot shows a terminal window titled "lb1 [En fonction] - Oracle VM VirtualBox". The menu bar includes "Fichier", "Machine", "Écran", "Entrée", "Périphériques", and "Aide". The terminal content shows the GNU nano 3.2 editor editing the file /etc/ipvsadm.rules. The file contains the following text:

```
# empty rules file for ipvsadm
ipvsadm -A -t 192.168.56.10:80 -s wrr
#Membre du cluster
ipvsadm -a -t 192.168.56.10:80 -r 192.168.200.11:80 -m -w 3
ipvsadm -a -t 192.168.56.10:80 -r 192.168.200.12:80 -m -w 1
```

L'ip flottante a bien été récupérée par lb2 :



```
lb2 [En fonction] - Oracle VM VirtualBox
Fichier Machine Écran Entrée Périphériques Aide

Message from syslogd@lb2 at Dec 14 22:37:48 ...
kernel:[ 3459.301648] watchdog: BUG: soft lockup - CPU#0 stuck for 189s! [swapper/0:0]
[ 3944.044707] watchdog: BUG: soft lockup - CPU#0 stuck for 224s! [swapper/0:0]
[ 3944.054233] systemd[1]: systemd-timesyncd.service: Watchdog timeout (limit 3min)!
[ 3944.064610] systemd[1]: systemd-logind.service: Watchdog timeout (limit 3min)!

Message from syslogd@lb2 at Dec 14 22:45:53 ...
kernel:[ 3944.044707] watchdog: BUG: soft lockup - CPU#0 stuck for 224s! [swapper/0:0]
^C
root@lb2:~# ^C
root@lb2:~# 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
        valid_lft forever preferred_lft forever
2: enp0s3: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state UP group default qlen 1000
    link/ether 08:00:27:c6:b1:08 brd ff:ff:ff:ff:ff:ff
    inet 192.168.56.12/24 brd 192.168.56.255 scope global enp0s3
        valid_lft forever preferred_lft forever
    inet 192.168.56.10/24 brd 192.168.56.255 scope global secondary enp0s3:0
        valid_lft forever preferred_lft forever
    inet6 fe80::a00:27ff:fec6:b108/64 scope link
        valid_lft forever preferred_lft forever
3: enp0s8: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state UP group default qlen 1000
    link/ether 08:00:27:0f:1e:68 brd ff:ff:ff:ff:ff:ff
    inet 192.168.200.252/24 brd 192.168.200.255 scope global enp0s8
        valid_lft forever preferred_lft forever
    inet 192.168.200.250/24 brd 192.168.200.255 scope global secondary enp0s8:0
        valid_lft forever preferred_lft forever
    inet6 fe80::a00:27ff:fe0f:1e68/64 scope link
        valid_lft forever preferred_lft forever
root@lb2:~#
```