LoadBalancing & HeartBeat

- Installer Load Balancing et le configurer
- Installer HeartBeat et le configurer

Installer Load Balancing et le configurer

Le Load Balancing c'est quoi ?

Le **load balancing** (ou équilibrage de charge) répartit le trafic entre plusieurs serveurs pour garantir disponibilité, performance et fiabilité.

Un **load balancer** agit comme intermédiaire : il reçoit les requêtes des utilisateurs et les redirige vers les serveurs disponibles en fonction de règles (par exemple, le moins chargé ou à tour de rôle).

Cela permet de :

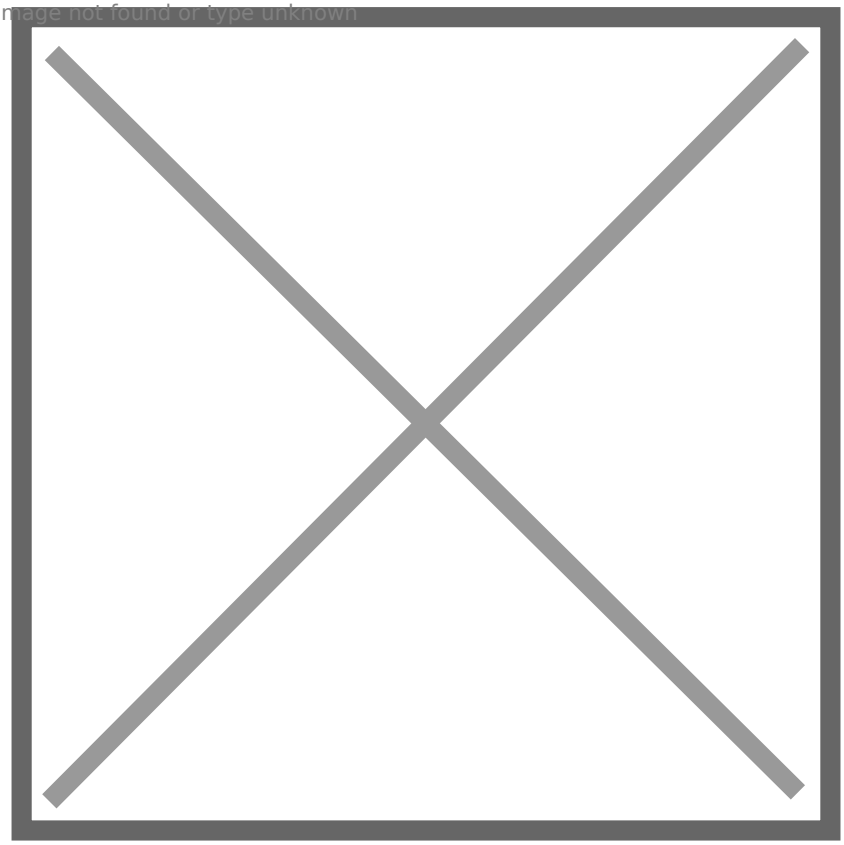
- **Éviter les surcharges** des serveurs.
- **Assurer la continuité** du service, même en cas de panne.
- **Optimiser les performances** en répartissant le travail équitablement.

On peut le faire via des solutions matérielles, logicielles ou cloud, comme HAProxy, Nginx, ou AWS Elastic Load Balancer.

Installation du LoadBalancing sur notre réseau

Dans notre cas nous allons suivre ce schema

Image not found or type unknown



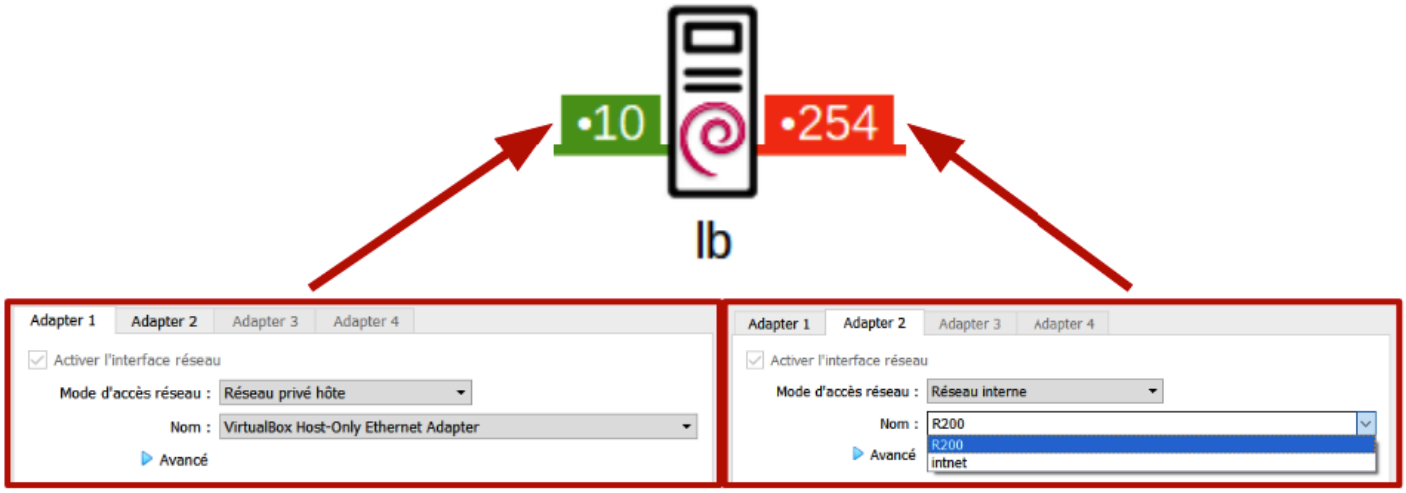
Pour commencer nous allons crée un serveur Maitre nomée " LoadBalancing "

On fais un update des packets et nous installations ipvsadm

```
apt update
apt install ipvsadm
```

Puis on ajouter sur virtualbox ou autre virtualiseur 2 interfaces réseau

- Public (Réseau privé hôte)
- Privée (R200)



En suite nous accédons a la configuration network du serveur LoadBalancing afin de configuré le réseau

```
nano /etc/network/interfaces
```

Et on applique le réseau ci dessous

```
# 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

# The 2nd network interfaces enp0s3

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

Après nous allons configurer ipvsadm

Nous allons dans le fichier **systctl.conf**

```
nano /etc/sysctl.conf
```

et on active Net.ipv4.ip_forward=1

```
#
# /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

###################################################################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://lwn.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
```

Et on vérifie que l'ip forward est bien activé cela doit vous afficher 1 (si cela ne s'afficher pas relancé la VM)

```
cat /proc/sys/net/ipv4/ip_forward
```

Maintenant nous allons dans le fichiers ipvsadm et nous mettons la configuration ci dessous

```
nano /etc/default/ipvsadm
```

```
# 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"
```

NB :

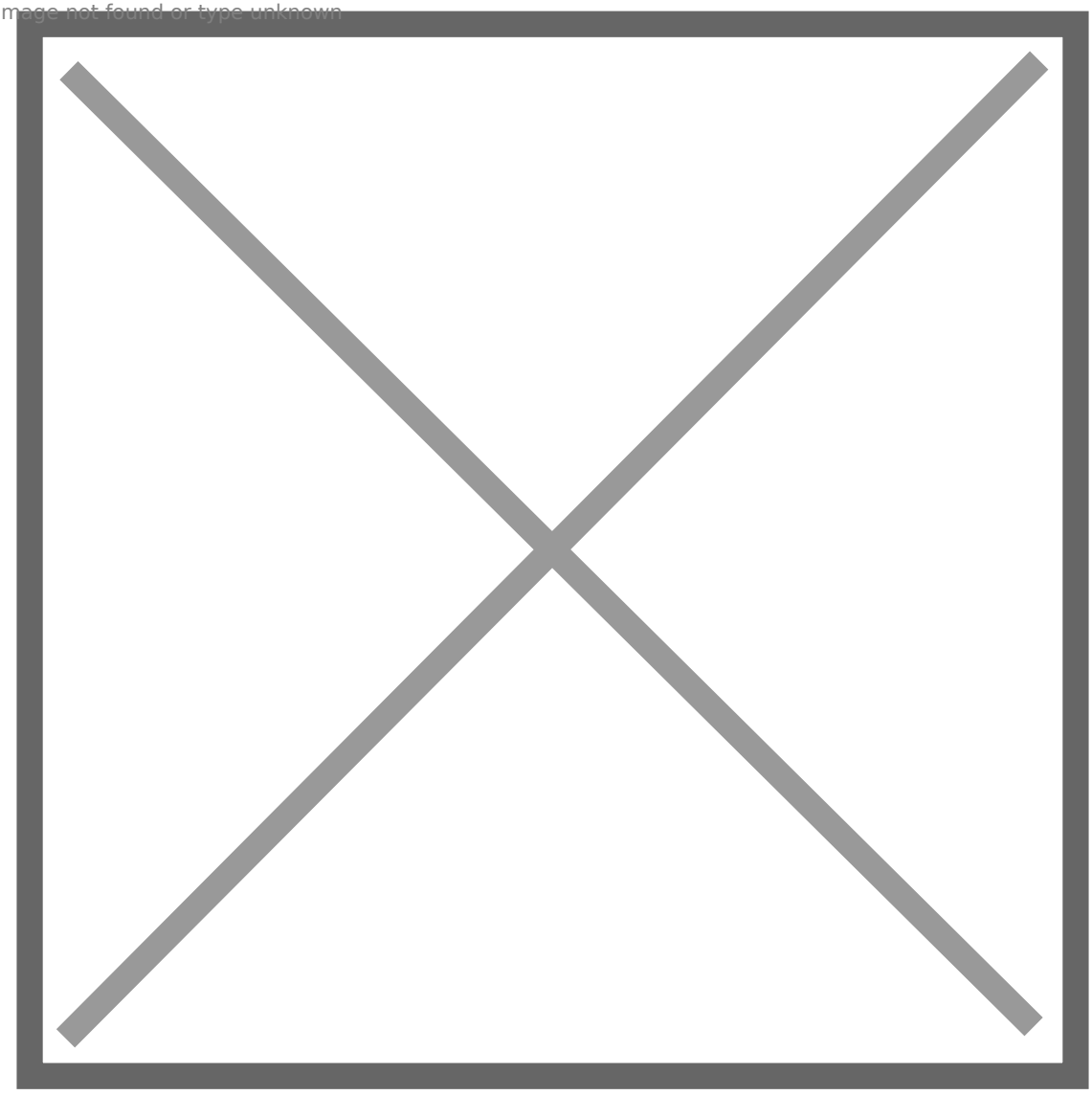
- Ligne 1 : Chargement de l'application et des règles au démarrage
- Ligne 2 : " Maitre " par défaut puisqu'il est le seul load balancer
- Ligne 3 : C'est par cette interface qu'arrivent les requêtes vers la grappe de serveurs Web

Puis nous allons dans le fichiers rules afin de mettre nos serveurs web

```
nano /etc/ipvsadm.rules
```

et nous mettons la configuration suivante :

Image not found or type unknown



NB Définition du service :

- -A ajoute un service, les éléments imporatns sont définis après. à savoir Protocol + @IP:PORT + Algorithme

- -t Protocol TCP
- -s Algorithme de Répartition Round Robin

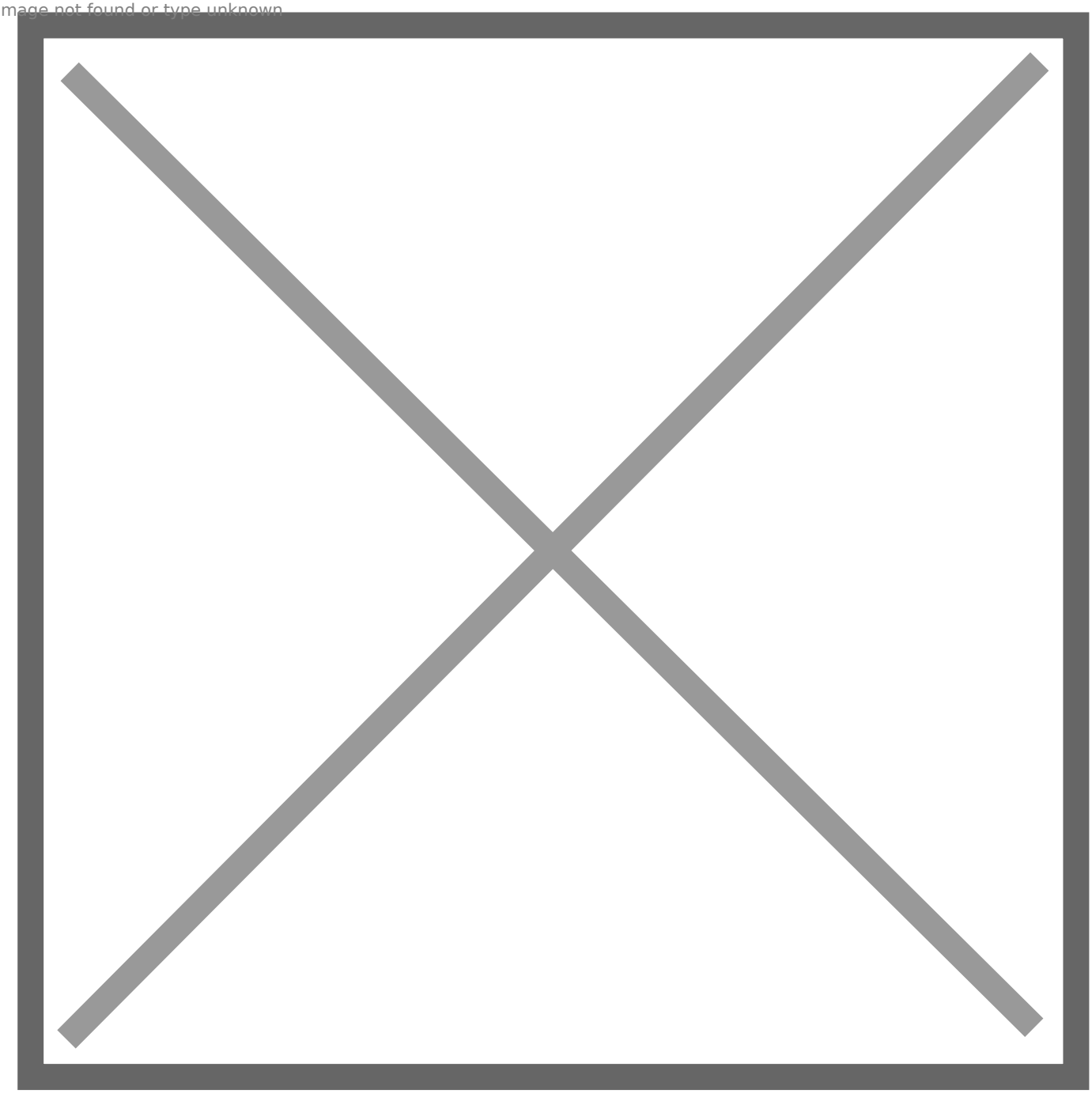
NB Membres du clusters :

- -a ajoute un noeud à un service
- -t:IP Service concerné
- -r:IP Adresse et port du noeud participant au cluster

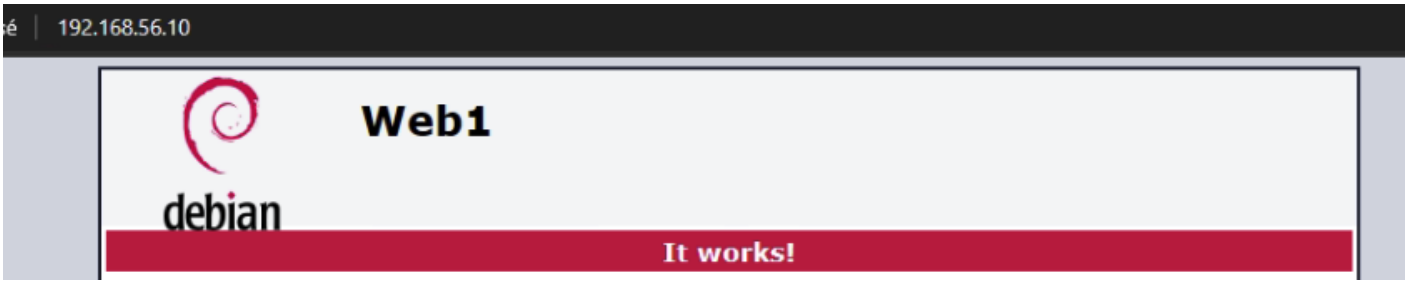
On effectue la commande suivante pour vérifier que tout est correct

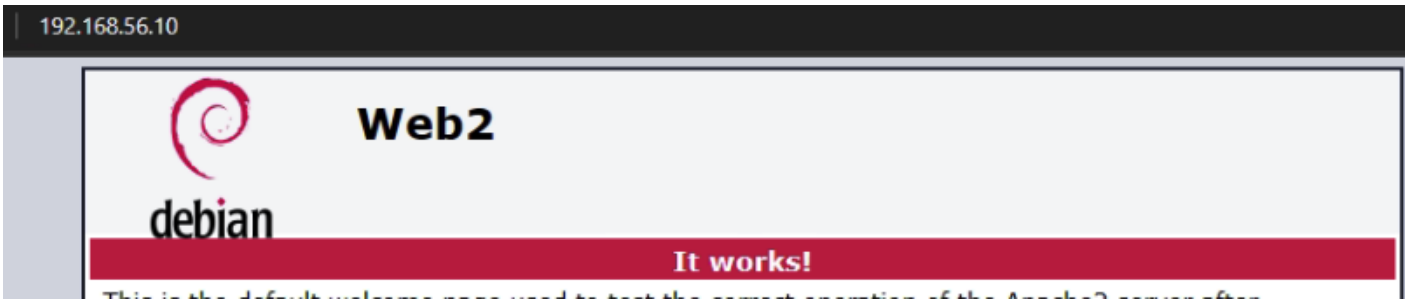
```
ipvsadm -ln
```

Image not found or type unknown



On se connecte au serveur lb 192.168.56.10 est nous devons tombé sur la page Web1 & Web2 si on refresh constament





Maintenant nous allons ajouter un Web3 a notre serveur

Pour cela on clone 1 des serveur Web on change l'ip de ce serveur avec la même interfaces réseau puis le text de la page apache

```
# 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 192.168.200.13/24
gateway 192.168.200.254
```

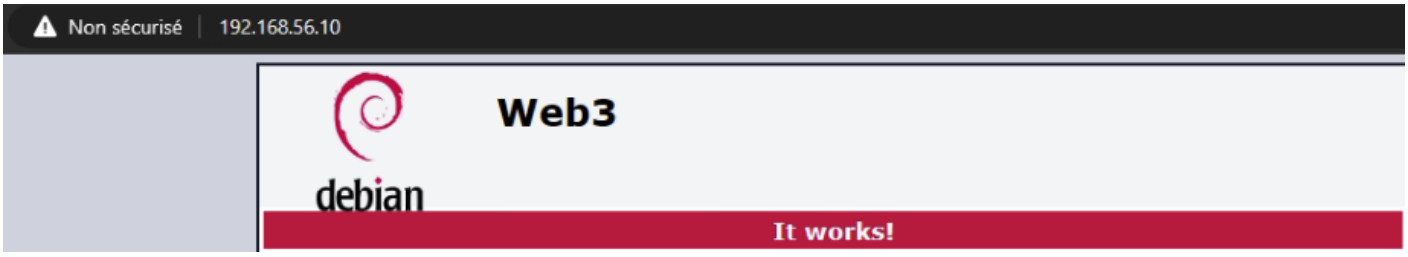
Maintenant on va dans le fichiers rules sur le serveur **LoadBalancing** et on rajoute simplement le 3eme serveur

```
nano /etc/ipvsadm.rules
```

```
# 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
ipvsadm -a -t 192.168.56.10:80 -r 192.168.200.13:80 -m
```

Après avoir ajouté le 3éme serveur on reboot le serveur lb pour être sur et on vérifie qu'on tombe également sur la page apache de Web3

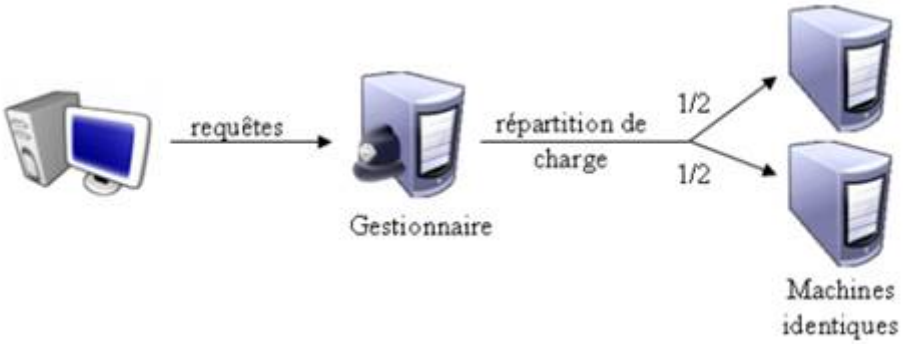


Installer HeartBeat et le configurer

Mise en place d'une infrastructure haute disponibilité

1. Introduction à la haute disponibilité

La haute disponibilité, également appelée High Availability (HA), constitue un pilier fondamental dans l’administration des systèmes informatiques. Elle vise à garantir l’accès continu aux services et aux applications, même en cas de défaillance matérielle ou durant des interventions de maintenance. Pour y parvenir, on met en place divers dispositifs tels que la duplication des composants, le basculement automatique en cas de panne, et un système de surveillance anticipée pour prévenir les incidents.



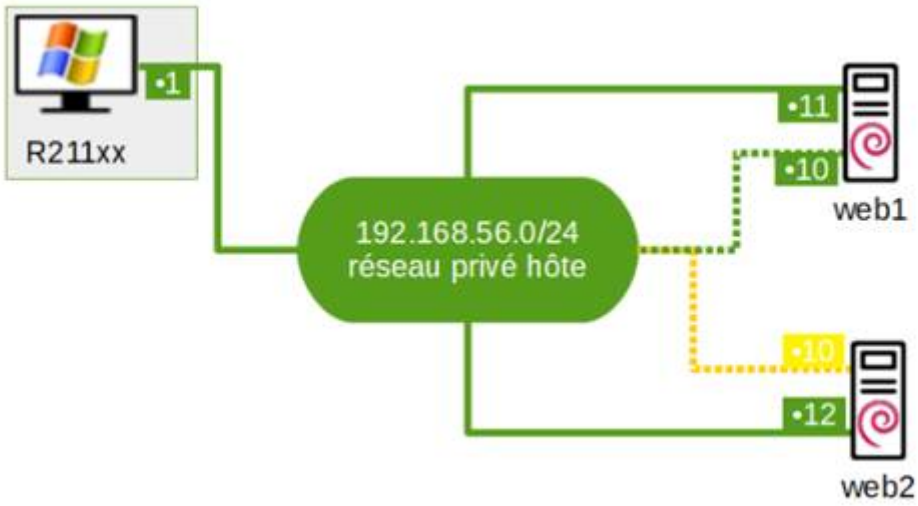
2. Heartbeat et la redondance de serveurs

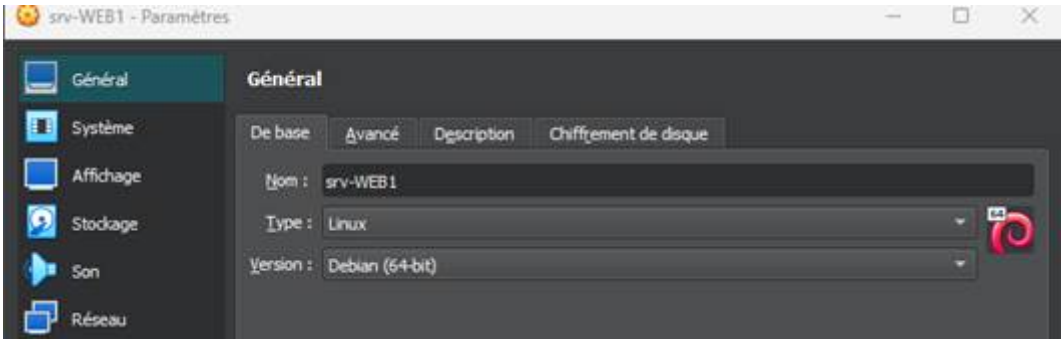
Afin d’assurer la continuité du service, un outil de supervision appelé « heartbeat » est mis en place. Il contrôle en permanence l’état du serveur principal. Si celui-ci devient indisponible, un serveur de secours prend automatiquement le relais. L’ensemble des serveurs, appelés « nœuds »,

forme un cluster. Grâce à une adresse IP virtuelle partagée, les utilisateurs peuvent accéder au service sans se préoccuper de quel nœud est actif à un instant donné.

3. Étapes de mise en œuvre

La première étape consiste à installer Apache2 sur chacun des deux serveurs web. Ensuite, il faut attribuer un nom d’hôte et une adresse IP spécifique à chaque machine. Pour les différencier visuellement, il est recommandé de personnaliser la page d'accueil par défaut d’Apache avec un nom unique à chaque serveur.





Poursuivez en installant le paquet **heartbeat** sur les deux serveurs. Le bon fonctionnement du système repose sur trois fichiers de configuration essentiels : `ha.cf`, `authkeys` et `haresources`. Il est impératif de sécuriser le fichier contenant la clé partagée à l’aide de la commande suivante :

```
chmod 600 /etc/ha.d/authkeys
```

4. Paramétrage détaillé

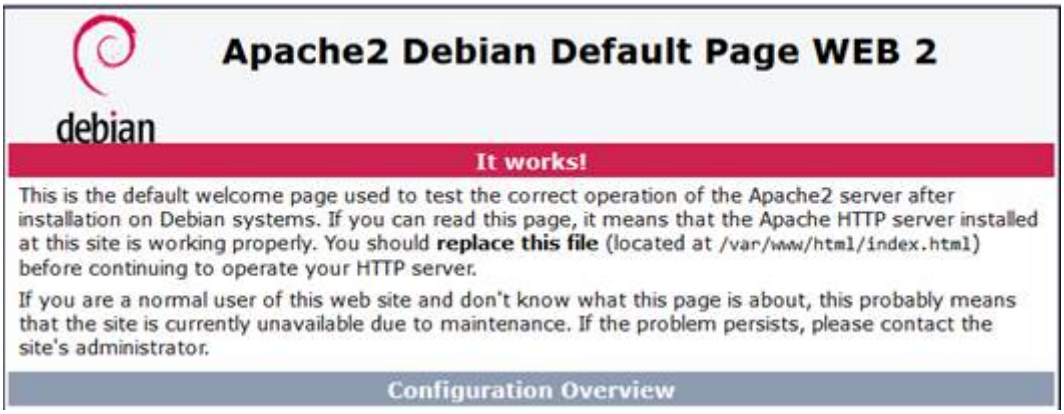
Le fichier `ha.cf` permet de configurer les paramètres de surveillance, notamment l'intervalle entre deux signaux (*keepalive*) et le temps d'attente avant de considérer un nœud comme inactif (*deadtime*). Il doit également contenir la liste complète des nœuds du cluster. De son côté, le fichier `haresources` définit les services à maintenir disponibles ainsi que l'adresse IP virtuelle utilisée pour les connexions clients.

5. Test et validation

Il faut d'abord stopper le service Apache2 sur les deux serveurs, puis empêcher son démarrage automatique. Une fois cela fait, activez **heartbeat** afin qu'il prenne en charge la gestion du service. Vous pouvez ensuite vérifier le bon fonctionnement du mécanisme en simulant l'arrêt du serveur principal : le serveur secondaire doit automatiquement prendre le relais, assurant ainsi la continuité du service sans interruption visible.

6. Notions de répartition de charge

Le **load balancing**, ou répartition de charge, est une technique qui permet de distribuer les requêtes entrantes entre plusieurs serveurs. Cela optimise les performances globales du système et renforce sa résilience face aux pannes. Cette gestion est assurée par un serveur spécifique, appelé **Load Balancer**, qui oriente le trafic selon des règles précises, comme l'algorithme de rotation (round robin) ou une pondération définie.


Apache2 Debian Default Page WEB 2
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