

La Haute Disponibilité

Sommaire :

I. Heartbeat

1. Objectifs :
2. Schéma réseau
3. Qu'est-ce que c'est Heartbeat ?
 - a. L'avantage d'Heartbeat
4. Prérequis
5. Installation et Configuration de Heartbeat
 - a. Configuration du fichier "ha.cf"
 - b. Configuration du fichier "authkeys"
 - c. Configuration du fichier "haresources"
 - a. Configuration du fichier "/etc/hosts"
6. Vérifier le bon fonctionnement
 - a. En cas de problème

II. Load Balancing

1. Objectifs :
2. Schéma du réseau
3. Qu'est-ce que le Load Balancing ?
4. Prérequis
5. Installation et Configuration du serveur "lb"
 - a. Configurer le serveur Load Balancing "lb"
 - b. Activation du routage
 - i. Vérification de l'activation routage
6. Installation Ipvsadm
7. Configuration des fichiers ipvsadm
 - a. 1er fichier à configurer "ipvsadm"
 - b. 2eme fichier à configurer "ipvsadm.rules"
 - c. Vérification du paramétrage
7. Tester le fonctionnement
 - A. En cas d'erreur

III. Ajout d'une machine Web3

1. Schéma réseau
2. Mise en place
3. Configuration du fichier ipvsadm.rules

IV. Heartbeat et Load Balancing

1. Schéma réseau
2. Ajouter le serveur LB2
3. Mise en place des systèmes
 - a. Modifier les interfaces réseaux des serveurs web
 - b. Installer heartbeat et load balancing sur lb2
 - c. Configuration de heartbeat sur les serveurs lb1 et lb2
 - d. Activer le routing
 - e. Configuration de heartbeat sur les serveurs lb1 et lb2

[f. Répartition des charges 'Round Robin'](#)

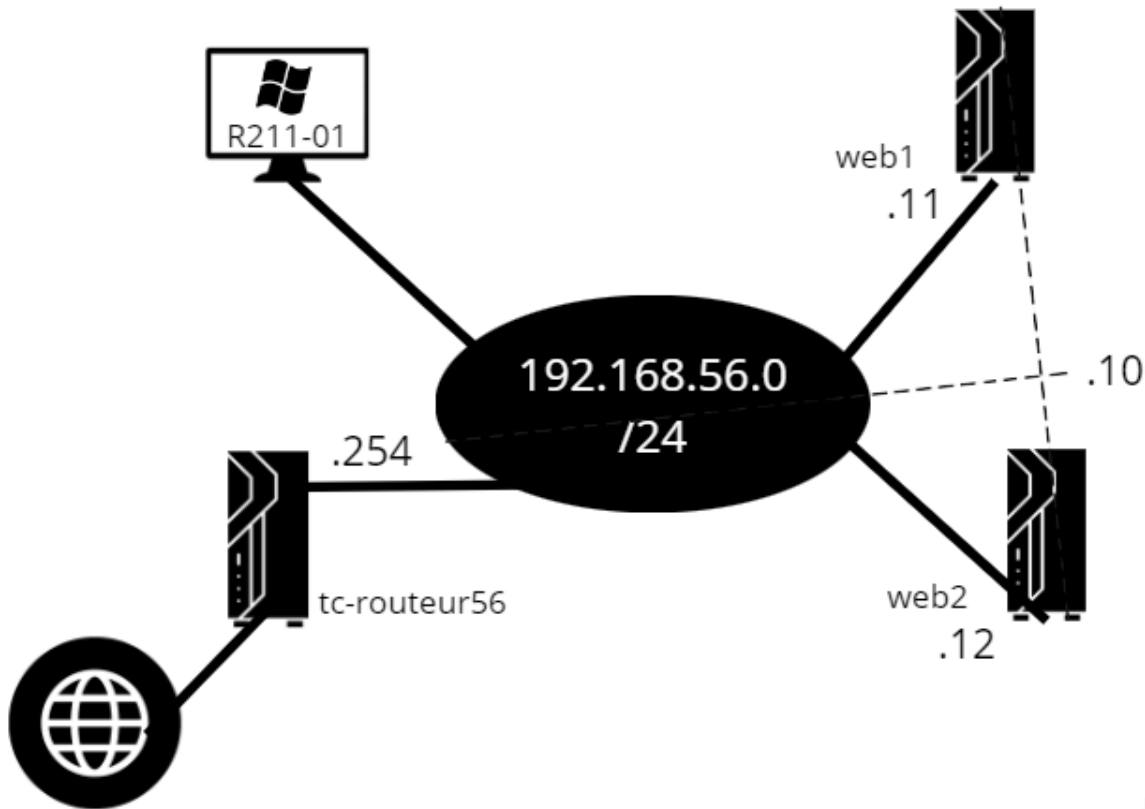
[4. Vérification du bon fonctionnement](#)

I. Heartbeat

1. Objectifs :

- ☐ **Installation du système d'exploitation :**
 - Installer Debian Buster sur deux serveurs.
 - Configurer une adresse IP statique pour chaque serveur.
- ☐ **Installation d'Apache2 :**
 - Sur chaque serveur, installer Apache2 en utilisant les commandes `sudo apt-get update` et `sudo apt-get install apache2`.
- ☐ **Configuration d'Apache2 :**
 - Assurer une configuration identique d'Apache2 sur les deux serveurs.
 - Synchroniser les fichiers de configuration au besoin.
- ☐ **Installation de Heartbeat :**
 - Sur chaque serveur, installer Heartbeat en utilisant la commande `sudo apt-get install heartbeat`.
- ☐ **Configuration de Heartbeat :**
 - Éditer le fichier `/etc/ha.d/ha.cf` sur les deux serveurs pour spécifier les paramètres de configuration tels que les adresses IP des serveurs.
- ☐ **Configuration de l'interface Heartbeat :**
 - Sur le serveur 1, éditer le fichier `/etc/ha.d/haresources` pour spécifier l'adresse IP flottante et les paramètres d'interface.
 - Sur le serveur 2, effectuer la même configuration que sur le serveur 1.
- ☐ **Démarrage de Heartbeat :**
 - Sur chaque serveur, démarrer Heartbeat en utilisant la commande `sudo service heartbeat start`.
- ☐ **Test de redondance :**
 - Éteindre Apache2 sur l'un des serveurs.
 - Vérifier que l'adresse IP flottante se déplace vers le serveur en ligne.
 - Assurer que le service Apache2 continue de fonctionner sans interruption.

2. Schéma réseau



3. Qu'est-ce que c'est Heartbeat ?

Heartbeat est un logiciel open source de **haute disponibilité** utilisé pour garantir la continuité des services sur des serveurs. Ses principales caractéristiques incluent la surveillance des **nœuds**, le basculement automatique en cas de défaillance, l'utilisation d'une adresse **IP virtuelle flottante**, une configuration simple, la gestion de divers services, et son intégration avec d'autres outils. **Heartbeat** permet de maintenir la disponibilité des applications en surveillant l'état des serveurs et en transférant automatiquement les services vers un serveur de secours en cas de besoin, assurant ainsi une haute disponibilité et une résilience aux pannes.

a. L'avantage d'Heartbeat

L'utilisation de "**heartbeat**" contribue à la **stabilité**, à la **disponibilité** et à la **gestion proactive** des systèmes informatiques, en permettant une détection rapide des problèmes et la mise en œuvre de stratégies de récupération ou de redondance.

4. Prérequis

- Renommer les machines (**Debainweb1** et **Debianweb2**)

nano /etc/hostname

- changer les adresses IP

nano /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
    gateway 192.168.56.254
```

(Faire la même chose pour la Debian 2 : **192.168.56.12**)

5. Installation et Configuration de Heartbeat

On va mettre en place une configuration de haute disponibilité avec des tests de Heartbeat avec Apache2.

Nous allons manipuler les deux serveurs DebianWeb1 et DebianWeb2 où nous allons tout d'abord installer **Apache2**.

apt update

apt install apache2

Pour vérifier que apache à bien été installé :

systemctl status apache2

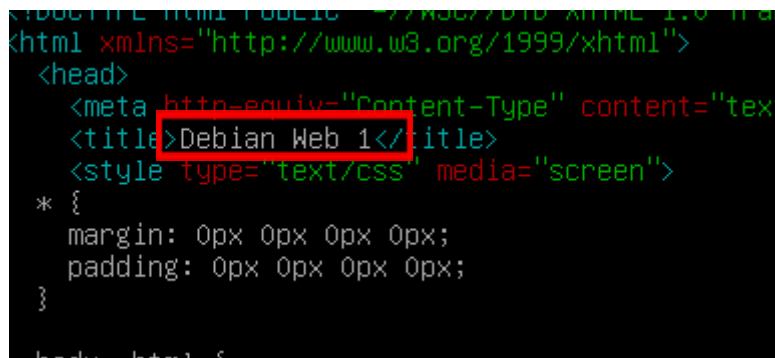
si besoin (systemctl start apache2)

Installer Heartbeat avec la commande :

apt update
apt install heartbeat


Modifier la page par **défaut d'Apache** et y inclure des informations spécifiques à chaque serveur, vous pouvez ajouter une section dans le fichier HTML qui affiche des détails tels que le nom du serveur. **On pourra voir le changement lors du test.**

nano /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>Debian Web 1</title>
  <style type="text/css" media="screen">
    * {
      margin: 0px 0px 0px 0px;
      padding: 0px 0px 0px 0px;
    }
  </style>
</head>
<body>
  <h1>Debian Web 1</h1>
</body>
</html>
```

Changer le titre en DebianWeb1 et pareil pour le web2



```
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
  <meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
  <title>Debian Web 2</title>
  <style type="text/css" media="screen">
    * {
      margin: 0px 0px 0px 0px;
      padding: 0px 0px 0px 0px;
    }
  </style>
</head>
<body>
  <h1>Debian Web 2</h1>
</body>
</html>
```

Après l'installation de Heartbeat, pensez à créer les 3 fichiers de configurations :

- **/etc/ha.d/ha.cf**
- **/etc/ha.d/authkeys**
- **/etc/ha.d/haresources**

a. Configuration du fichier "ha.cf"

Dans le serveur DebianWeb1 et DebianWeb2, effectuer la configuration suivante dans le fichier ha.cf.

```
bcast enp0s3
deadtime 5
keepalive 1
node DebianWeb1 DebianWeb2
```

- **bcast :**
Il spécifie l'interface réseau à utiliser pour la diffusion (broadcast). Cela indique à Heartbeat sur quelle interface il doit écouter les diffusions de statut des autres nœuds du cluster.
- **deadtime :**
Il spécifie le temps en secondes pendant lequel Heartbeat attend avant de considérer un nœud comme mort (défaillant) après avoir cessé de recevoir des signaux de ce nœud. Dans votre exemple, deadtime 5 signifie que si Heartbeat ne reçoit pas de signaux du nœud pendant 5 secondes, il considérera ce nœud comme mort.
- **keepalive :**
Il spécifie la fréquence à laquelle Heartbeat envoie des signaux de vie (keepalive) à d'autres nœuds pour indiquer qu'il est toujours en vie. keepalive 1 signifie qu'un signal de vie est envoyé toutes les 1 seconde.
- **node :**
Il spécifie les noms des nœuds dans le cluster. Dans votre exemple, node serveur1 serveur2 indique qu'il y a deux nœuds dans le cluster avec les noms "serveur1" et "serveur2". Ces noms doivent correspondre aux noms de nœuds que vous utilisez dans le cluster.

b. Configuration du fichier "authkeys"

Le service heartbeat exige une protection de ce fichier sinon il ne démarrera pas et serait visible par n'importe qui.

```
auth1
1 md5 motdepasse
```

clé partagée entre les serveurs de la grappe (même chose sur les 2 serveurs donc...). Ce fichier détermine la clé et le protocole de protection utilisé.

Autre configuration possible :

auth1
1 sha1 MaClefSecrete

Assurez-vous que le fichier **authkeys** est accessible uniquement par les utilisateurs autorisés:

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

Cela définit les permissions du fichier **authkeys** de manière à ce qu'il soit accessible en **lecture** et **écriture** uniquement par le **propriétaire** du fichier. Assurez-vous d'ajuster les permissions en fonction de vos besoins de **sécurité spécifiques**.

Vous pouvez vérifier les nouvelles permissions en utilisant la commande.

```
ls -l /etc/ha.d/authkeys
```

c. Configuration du fichier “haresources”

Ajoutez les ressources que vous souhaitez surveiller et gérer. Chaque ligne de ce fichier spécifie une ressource. Par exemple, si vous souhaitez surveiller une adresse IP virtuelle et un service Apache, vous pouvez avoir quelque chose comme ceci :

```
DebianWeb1 IPaddr::192.168.56.10        apache2
```

- **nom_de_la_machine** : Le nom de la machine qui sera activée par défaut au démarrage de Heartbeat. Assurez-vous que ce nom est identique sur les deux machines.
- **IPaddr::192.168.56.10** : Une adresse IP virtuelle flottante que Heartbeat gérera. Assurez-vous que cette adresse est unique et accessible sur le réseau.
- **apache2** : Le service à surveiller. Vous pouvez ajouter d'autres services si nécessaire.

a. Configuration du fichier “/etc/hosts”

Le fichier **/etc/hosts** est utilisé pour associer des adresses IP à des noms d'hôtes sur un système. Si vous n'avez pas de service DNS installé et que vous souhaitez déclarer les hôtes web1 et web2 dans ce fichier, vous pouvez le faire de la manière suivante.

```
GNU nano 3.2 /etc/hosts
127.0.0.1    localhost
127.0.1.1    DebianWeb1
192.168.56.12 DebianWeb2

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

Cela peut être particulièrement utile pour assurer une résolution de noms locale et rapide, en particulier dans des environnements où un serveur DNS central n'est pas disponible ou n'est pas souhaité. Assurez-vous de répéter ces étapes sur chaque machine de votre cluster.

Pensez à inverser pour l'autre serveur.

```
127.0.0.1    localhost
127.0.1.1    DebianWeb2
192.168.56.11 DebianWeb1
# The following lines are desirable for IPv6 capable hosts
::1          localhost ip6-localhost ip6-loopback
ff02::1      ip6-allnodes
ff02::2      ip6-allrouters
```

6. Vérifier le bon fonctionnement

Commençons par stopper le service apache2 sur les 2 machines.

systemctl disable apache2.service

```
root@DebianWeb1:/etc/ha.d# systemctl disable apache2.service
Synchronizing state of apache2.service with SysV service script with /lib/systemd/systemd-sysv-install.
Executing: /lib/systemd/systemd-sysv-install disable apache2
Removed /etc/systemd/system/multi-user.target.wants/apache2.service.
root@DebianWeb1:/etc/ha.d# _
```

Vérifier que le service est bien arrêté. Remplacer “stop” par “status”.

systemctl status apache2.service

Tester que le service heartbeat est bien actif. Le redémarrer avec la commande suivante :

systemctl restart heartbeat.service

a. En cas de problème

Si le service heartbeat ne **redémarre** pas correctement :

- Relire fichier de config
- Redémarrer les vm
- Stopper les services apache2
- Redémarrer heartbeat

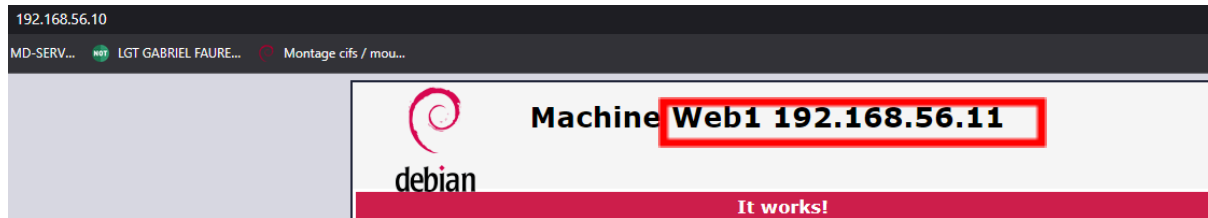
Utilisez la commande `ip a` sur la machine "**web1**" pour vérifier son adresse IP prévue, qui devrait être "**192.168.56.10**".

```
link/ether 08:00:27:2c:9b:f1 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:fe2c:9bf1/64 scope link
    valid_lft forever preferred_lft forever
```

Vérifier également sur la machine "**web2**" pour voir si l'adresse "**192.168.56.10**" s'affiche bien.

```
link/ether 08:00:27:40:98:54 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:fe40:9854/64 scope link
    valid_lft forever preferred_lft forever
```

En vérifiant l'adresse IP de la grappe, connectez-vous depuis la machine hôte à l'adresse "**192.168.56.10**". Vous devriez être redirigé vers la page par défaut d'Apache sur la machine "**web1**".



Afin de confirmer que "web2" prend effectivement l'IP "192.168.56.10" en cas d'indisponibilité de "web1", arrêtez le service Apache sur "web1" en utilisant la commande suivante :

```
systemctl stop apache2.service
```

Cela simule une coupure du service Apache sur "web1" et permet de vérifier si "web2" prend en charge l'adresse IP "192.168.56.10" comme prévu.

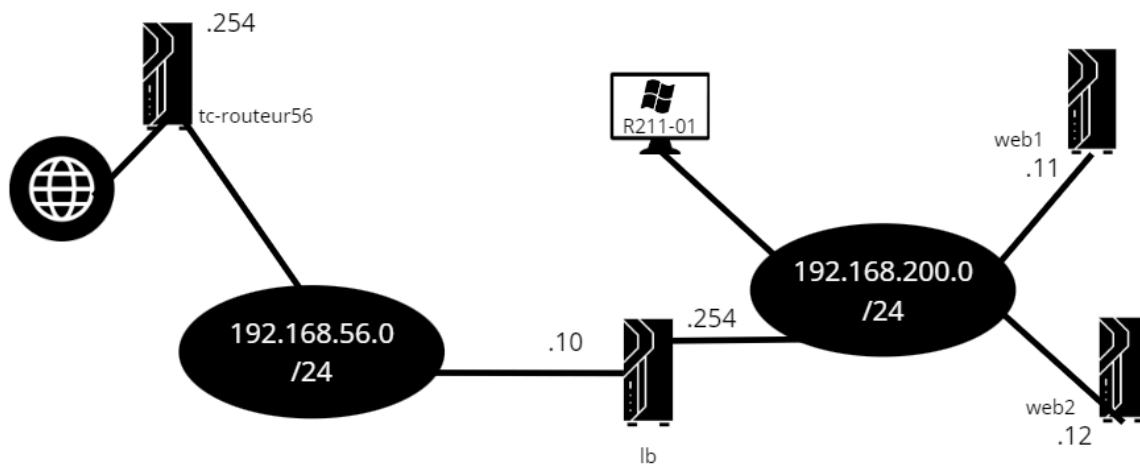
II. Load Balancing

1. Objectifs :

- ☐ **Répartition équilibrée de la charge** : Mettre en place un système de load balancing pour répartir de manière équilibrée le trafic entre deux serveurs Web équipés d'Apache2. Cela vise à optimiser l'utilisation des ressources disponibles et à améliorer les performances du système.
- ☐ **Haute disponibilité** : Assurer une disponibilité élevée du service en configurant le "load balancer" de manière à rediriger le trafic vers un serveur fonctionnel en cas de panne ou d'indisponibilité d'un des serveurs Apache2. Cela garantira une continuité de service même en présence de défaillances matérielles ou logicielles.

- ☐ **Sécurité renforcée** : Utiliser le "load balancer" comme point d'entrée principal pour les clients, masquant ainsi directement les serveurs Apache2. Cela renforce la sécurité en limitant l'exposition des serveurs Web au réseau extérieur, réduisant ainsi la surface d'attaque potentielle.
- ☐ **Gestion des sessions** : Mettre en œuvre une gestion efficace des sessions pour garantir une expérience utilisateur cohérente, même en cas de changement de serveur. Ceci est particulièrement important pour les applications nécessitant une continuité des sessions utilisateur.
- ☐ **Surveillance et gestion des performances** : Configurer des outils de surveillance pour suivre les performances des serveurs Apache2 et du "load balancer". Mettre en place des mécanismes de gestion automatique pour ajuster la répartition de la charge en fonction de la charge système, garantissant ainsi des performances optimales.

2. Schéma du réseau



3. Qu'est-ce que le Load Balancing ?

Le "**load balancing**" (équilibrage de charge) est une technique utilisée dans les systèmes informatiques pour **distribuer** efficacement la **charge de travail** entre plusieurs composants, tels que serveurs, disques ou autres ressources réseau. L'objectif principal du "**load balancing**" est d'optimiser l'utilisation des ressources disponibles, d'**améliorer les performances**, de **garantir la haute disponibilité du service**, et de **fournir une résilience en cas de défaillance d'un des composants**.

4. Prérequis

Nous allons tout d'abord configurer l'**@IP** de **web1** et **web2**:

```
# This file describes the network in
# and how to activate them. For more

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
    gateway 192.168.200.254
```

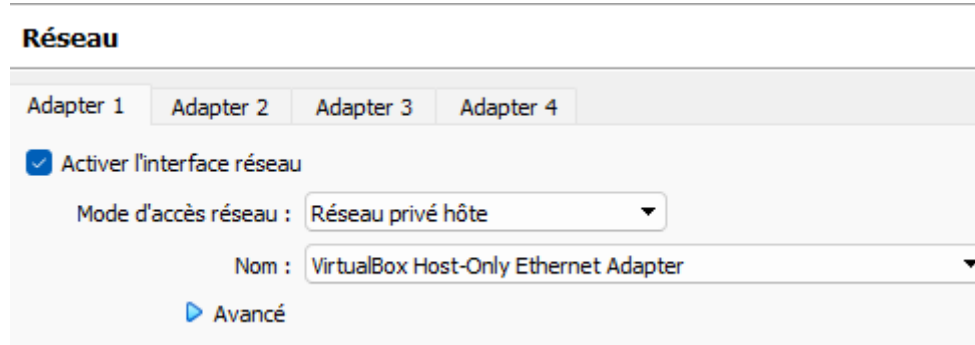
Web1

Web2

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

5. Installation et Configuration du serveur “lb”

Nous allons mettre en place **2 adaptateurs réseaux**, le premier concerne le réseau **192.168.56.0**.



Réseau

Adapter 1 Adapter 2 Adapter 3 Adapter 4

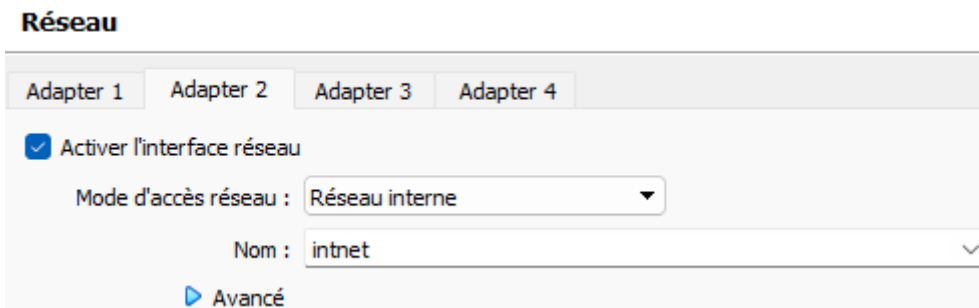
☒ Activer l'interface réseau

Mode d'accès réseau : Réseau privé hôte

Nom : VirtualBox Host-Only Ethernet Adapter

▶ Avancé

Le deuxième adaptateur réseau concerne le réseau **192.168.200.0**.



Réseau

Adapter 1 Adapter 2 Adapter 3 Adapter 4

☒ Activer l'interface réseau

Mode d'accès réseau : Réseau interne

Nom : intnet

▶ Avancé

a. Configurer le serveur Load Balancing “lb”

Configurer les interfaces sur la machine Serveur lb.

Réseau privé hôte (.10)

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

Réseau interne (.254)

```
#2eme
allow-hotplug enp0s8
iface enp0s8 inet static
    address 192.168.200.254/24_
```

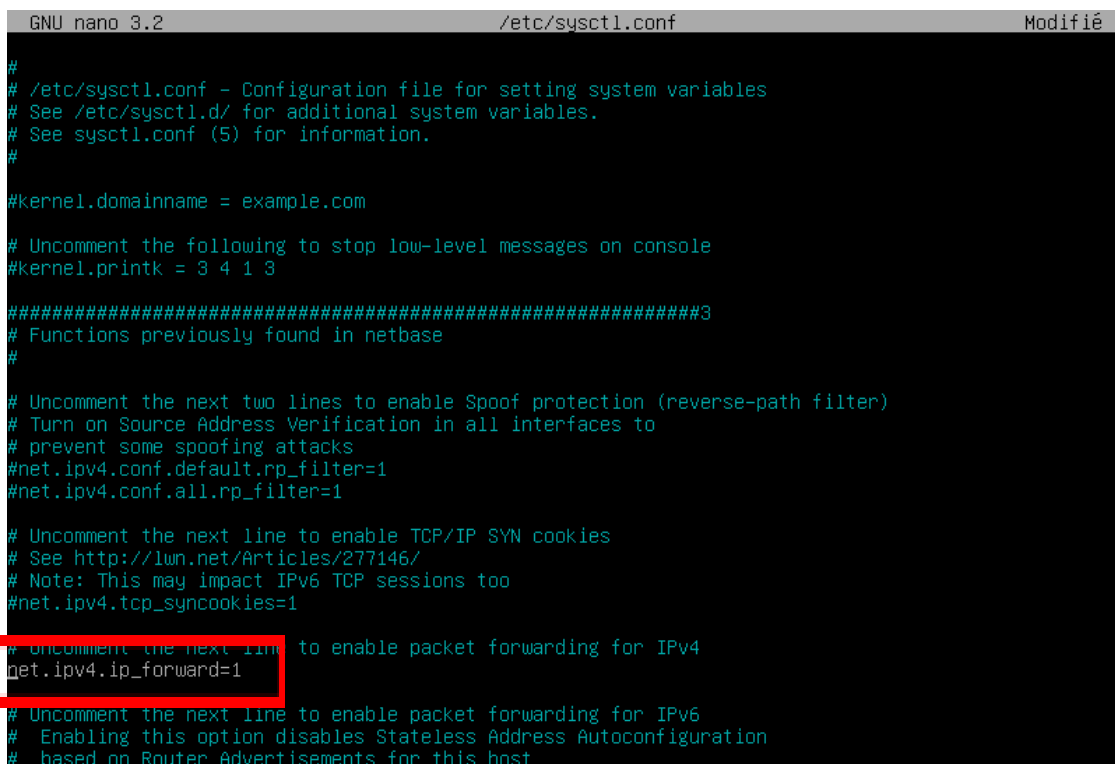
b. Activation du routage

Vous devez activer le routage IP sur la machine. Pour cela, ouvrez le fichier de configuration **sysctl.conf** :

nano /etc/sysctl.conf

Recherchez la ligne qui contient **net.ipv4.ip_forward** et assurez-vous qu'elle est **décommentée** (c'est-à-dire que le symbole # n'est pas présent au début de la ligne). Si la ligne n'existe pas, ajoutez-la à la fin du fichier :

Le “1” active le routage “net.ipv4.ip_forward=1



```
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
#####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
# Uncomment the next line to enable packet forwarding for IPv6
# Enabling this option disables Stateless Address Autoconfiguration
# based on Router Advertisements for this host
```

On peut bien sûr le vérifier avec :

i. Vérification de l'activation routage

Pour vérifier si le routage IP est **activé** sur votre système, vous pouvez examiner la valeur du paramètre **ip_forward** dans le répertoire **/proc/sys/net/ipv4/**. Vous pouvez le faire en utilisant la commande cat.

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

Pensez à redémarrer la machine pour que le fichier s'actualise

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

6. Installation Ipv6adm

Pour installer le paquet **ipvsadm** sur une distribution basée sur Debian. Effectuer les commandes suivantes :

```
apt update
apt install ipvsadm
```

```
Fichier de configuration « /etc/ipvsadm.rules »
==> Fichier du système créé par vous ou par un script.
==> Fichier également présent dans le paquet fourni par le responsable du paquet.
Que voulez-vous faire ? Vos options sont les suivantes :
  Y ou I : installer la version du responsable du paquet
  N ou O : garder votre version actuellement installée
  D      : afficher les différences entre les versions
  Z      : suspendre ce processus pour examiner la situation
L'action par défaut garde votre version actuelle.
*** ipvsadm.rules (Y/I/N/O/D/Z) [défaut=N] ? O
Progression : [ 67%] [#####.....]
```

7. Configuration des fichiers ipvsadm

Nous allons maintenant configurer les 2 fichiers suivants :

 `/etc/default/ipvsadm`

 `/etc/ipvsadm.rules`

a. 1er fichier à configurer "ipvsadm"

```
GNU nano 3.2 /etc/default/ipvsadm
# ipvsadm
# if you want to start ipvsadm on boot set this to true
AUTO="true"
# daemon method (none|master|backup)
DAEMON="master"
# use interface (eth0,eth1...)
IFACE="enp0s3"
# syncid to use
# (0 means no filtering of syncids happen, that is the default)
# SYNCID="0"
```

- **AUTO = true** : Chargement de l'application et des règles au démarrage.
- « **Maître** » par défaut puisqu'il est le seul load balancer.
- **IFACE="enp0s3"** : C'est par cette interface qu'arrivent les requêtes vers le cluster de serveurs Web.

b. 2eme fichier à configurer "ipvsadm.rules"

Configurer le fichier comme affiché ci-dessous :

```
GNU nano 3.2 /etc/ipvsadm.rules

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

#les 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
```

- **A**: Indique l'ajout d'un service.
- **t** [VIP:Port]: Spécifie l'adresse IP virtuelle (VIP) et le port du service.
- **s** [Protocole]: Spécifie le protocole du service.
- **rr** : Algorithme de répartition Round Robin

c. Vérification du paramétrage

La commande `ipvsadm -ln` est utilisée pour afficher les informations sur les services configurés sur le Load Balancer IPVS. Voici ce que signifient les options de cette commande:

ipvsadm -ln

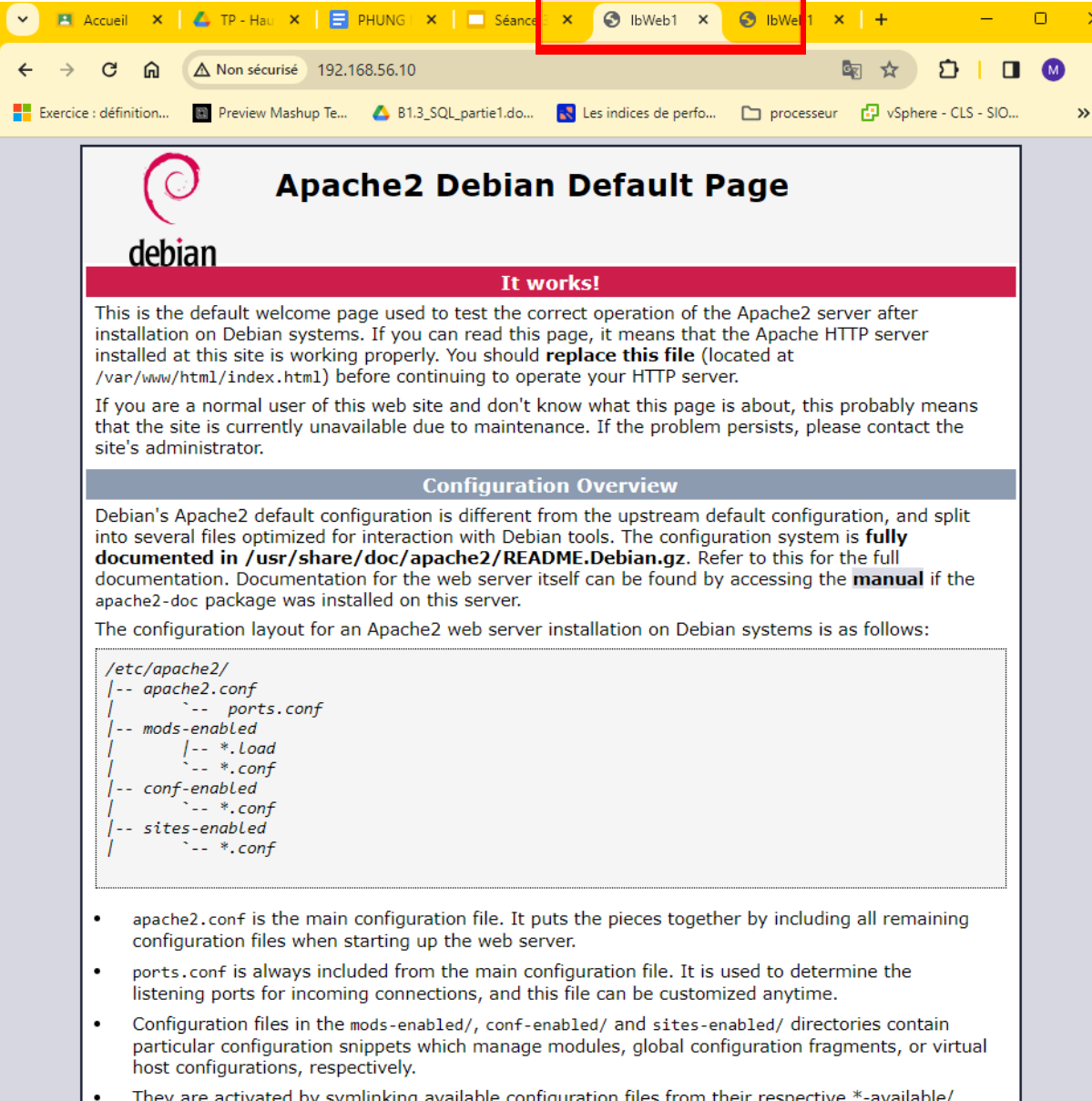
- **'l'**: Affiche les informations sur les services configurés.
- **'n'**: Affiche les adresses IP et les numéros de port au format numérique plutôt qu'au format symbolique.

```
root@lb:~# 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
root@lb:~#
```

7. Tester le fonctionnement

Dans le navigateur : **192.168.56.10**

fonctionne grâce à web1



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

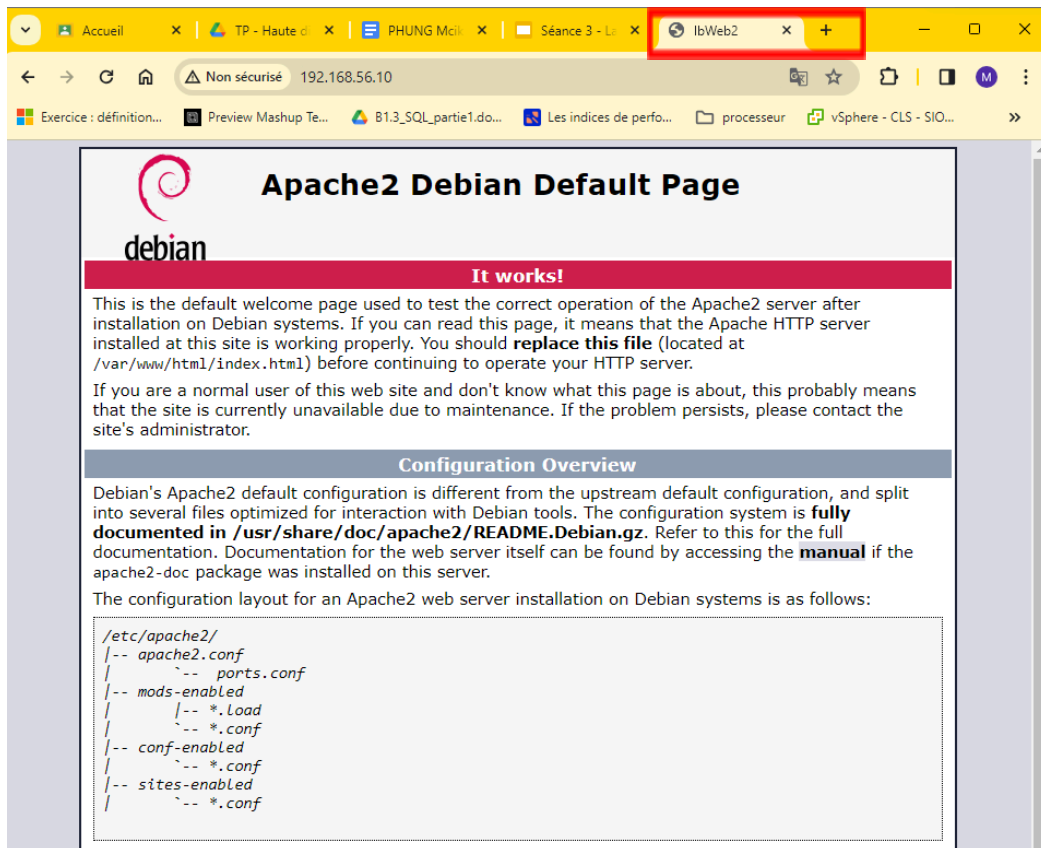
Debian's Apache2 default configuration is different from the upstream default configuration, and split into several files optimized for interaction with Debian tools. The configuration system is **fully documented in `/usr/share/doc/apache2/README.Debian.gz`**. Refer to this for the full documentation. 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/
|-- apache2.conf
/
|-- ports.conf
/
|-- mods-enabled
/   |-- *.load
/   |-- *.conf
/
|-- conf-enabled
/   |-- *.conf
/
|-- sites-enabled
/   |-- *.conf
```

- `apache2.conf` is the main configuration file. It puts the pieces together by including all remaining configuration files when starting up the web server.
- `ports.conf` is always included from the main configuration file. It is used to determine the listening ports for incoming connections, and this file can be customized anytime.
- Configuration files in the `mods-enabled/`, `conf-enabled/` and `sites-enabled/` directories contain particular configuration snippets which manage modules, global configuration fragments, or virtual host configurations, respectively.
- They are activated by symlinking available configuration files from their respective `*-available/`

Sur web 2

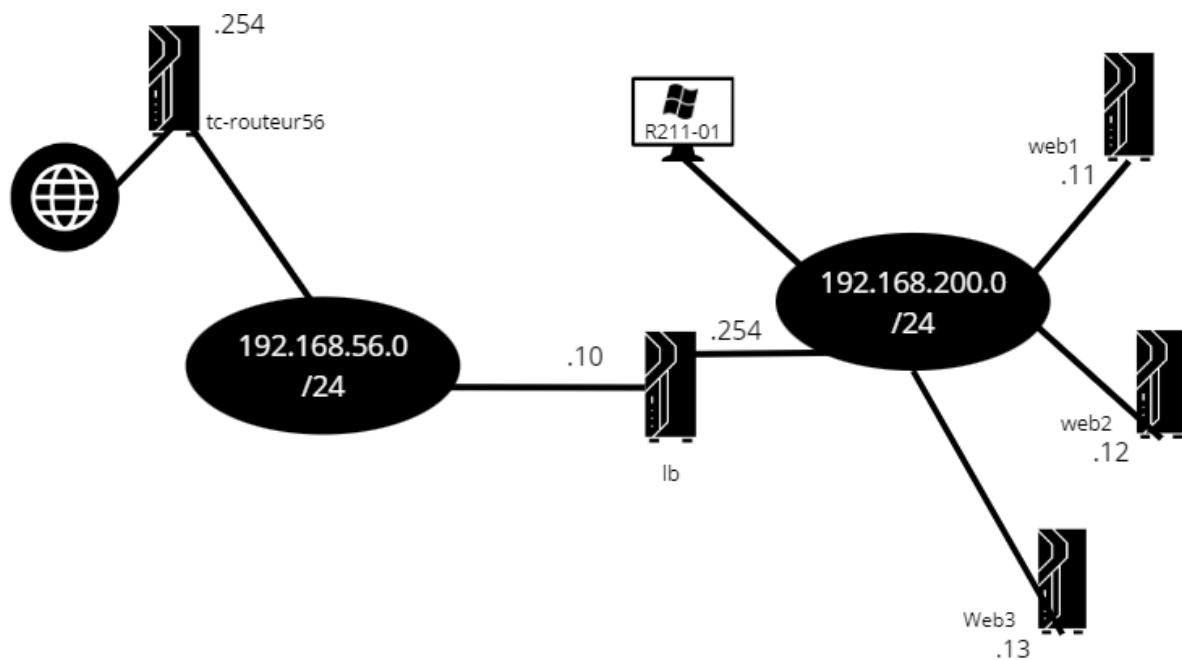


A. En cas d'erreur

- Confirmer le bon acheminement des requêtes HTTP en utilisant un navigateur ou un outil tel que curl pour interagir avec l'adresse IP virtuelle associée au Load Balancer.
- S'assurer que les serveurs web (web1 et web2) sont opérationnels et que leurs journaux ne signalent pas d'erreurs majeures.
- Tester la connectivité depuis le load balancer (lb) vers les serveurs web (web1 et web2) en utilisant la commande ping.
- Examiner attentivement la configuration réseau de chaque machine virtuelle (web1 et web2) dans VirtualBox pour garantir que les paramètres IP et les configurations réseau sont corrects.

III. Ajout d'une machine Web3

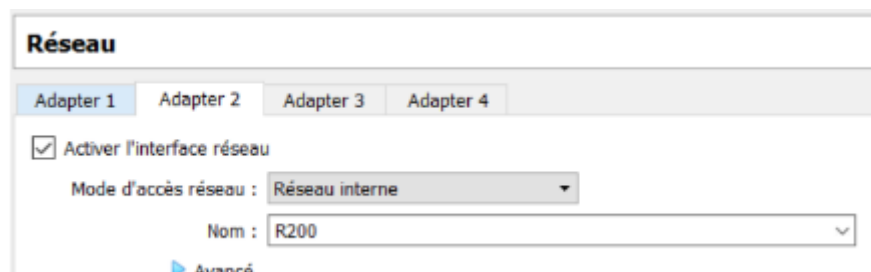
1. Schéma réseau



2. Mise en place

Commençons par créer une nouvelle machine virtuelle, **Web3**. Une fois celle-ci configurée, nous pourrons l'intégrer au load balancer afin de répartir la charge des requêtes entre nos trois machines du réseau.

Pensez à modifier l'**adaptateur réseau** en **réseau interne** pour **Web3**.



Ensuite, nous avons la possibilité de mettre à jour la configuration réseau selon les directives suivantes :

```
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 192.168.200.13
    gateway 192.168.200.254
```

Il faudra également installer **apache2** et modifier la page par défaut d'apache afin de distinguer **Web3**.

```
<body>
  <div class="main_page">
    <div class="page_header floating_element">
      
      <span class="floating_element">
        Apache2 Debian Default Page : Web3
      </span>
    </div>
```

3. Configuration du fichier ipvsadm.rules

Ensuite, procédez à la modification du fichier ipvsadm.rules sur le load balancer (lb) selon les instructions suivantes :

/etc/ipvsadm.rules

```

GNU nano 3.2 /etc/ipvsadm.rules

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

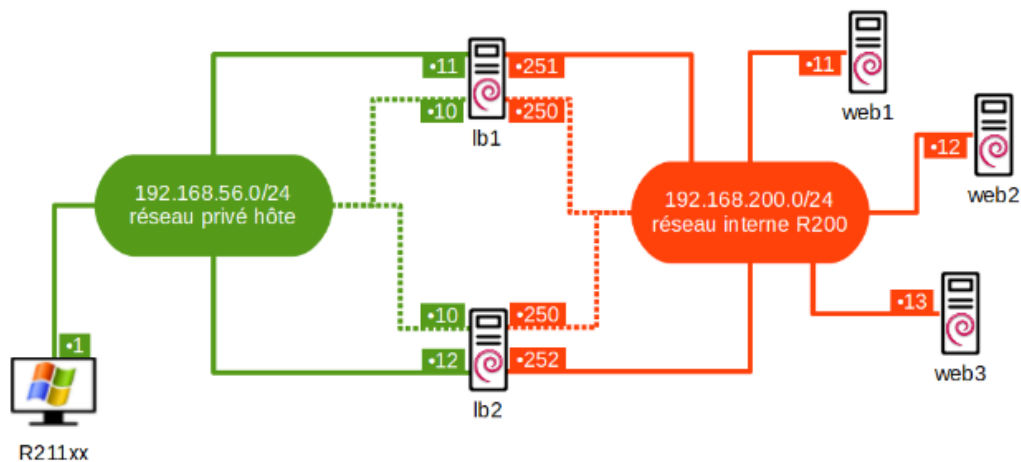
#Les membres du clusters
ipvsadm -a -t 192.168.56.10:80 -r 192.168.200.11:80 -m
ipvsadm -a -t 192.168.56.10:80 -r 192.168.200.12:80 -m
ipvsadm -a -t 192.168.56.10:80 -r 192.168.200.13:80 -m_

```

On va venir ajouter le **Web3** : **192.168.200.13**

IV. Heartbeat et Load Balancing

1. Schéma réseau



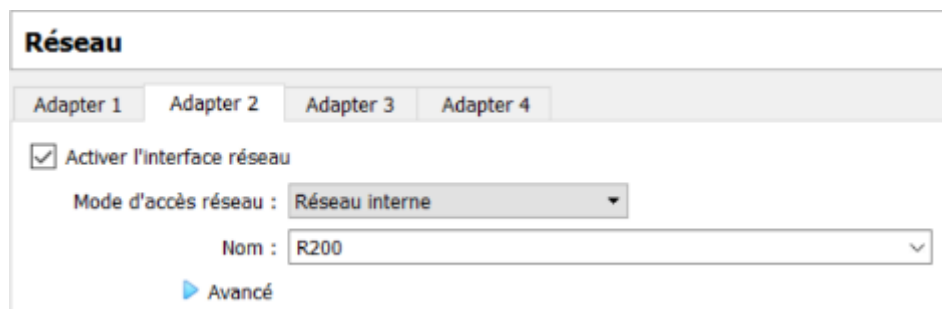
2. Ajouter le serveur LB2

On possède déjà les machines suivantes :

- lb1
- Web1
- Web2
- Web3

Il suffit de créer le **serveur lb2**

On va venir ajouter une deuxième interface en réseau interne pour lb2 :



- Changer également son IP

```
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
    gateway 192.168.56.254

#2eme
allow-hotplug enp0s8
iface enp0s8 inet static
    address 192.168.200.252/24
```

3. Mise en place des systèmes

- Faites de même pour **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 192.168.56.11
    gateway 192.168.56.254

#2eme
allow-hotplug enp0s8
iface enp0s8 inet static
    address 192.168.200.251/24
```

a. Modifier les interfaces réseaux des serveurs web

- Web1 : 192.168.200.11 / gateway : 192.168.200.250
- Web2 : 192.168.200.12 / gateway : 192.168.200.250
- Web3 : 192.168.200.13 / gateway : 192.168.200.250

b. Installer heartbeat et load balancing sur lb2

Après avoir configuré le serveur **load balancer 2** (lb2), vous pouvez installer **Heartbeat** et configurer le **load balancing**. Voici les étapes générales pour effectuer cette opération :

```
apt update
apt install heartbeat
```

Puis pour load balancing :

```
apt update
apt install ipvsadm
```

c. Configuration de heartbeat sur les serveurs lb1 et lb2

Crées les fichiers :

- ha.cf

```
GNU nano 3.2 /etc/ha.d/ha.cf
bcast enp0s3 enp0s8
deadtime 5
keepalive 1
node lb1 lb2
```

- authkeys

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

Le service heartbeat exige une protection de ce fichier sinon il ne démarrera pas et sera visible par n'importe qui.

chmod 600 /etc/ha.d/authkeys

- haresources

```
GNU nano 3.2 /etc/ha.d/haresources
lb1 IPaddr::192.168.56.10      ipvsadm
lb1 IPaddr::192.168.200.250   enp0s8
```

Créer l'IP flottante (virtuelle) du réseaux interne (on l'affilie à l'interface enp0s8) dans le répertoire **/etc/ha.d/haresources** de lb1 et lb2

d. Activer le routing

Les **deux machines Load balancer** doivent prendre en compte le routage, on peut donc les activer exactement de la même manière que précédemment c'est-à-dire dans le fichier :

nano /etc/sysctl.conf

Vérifier avec :

cat /proc/sys/net/ipv4/ip_forward

```
root@lb2:~# cat /proc/sys/net/ipv4/ip_forward
1
```

e. Configuration de heartbeat sur les serveurs lb1 et lb2

f. Répartition des charges 'Round Robin'

- Mettre en place une **répartition des charges avec 3 charges** pour web1 et une pour web2 et web3.

Sur les 2 serveurs **lb1** et **lb2** configurer le fichier `/etc/ipvsadm.rules`.

```
GNU nano 3.2 /etc/ipvsadm.rules

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

#Les membres du clusters
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
ipvsadm -a -t 192.168.56.10:80 -r 192.168.200.13:80 -m -w 1
```

- **'Wrr'** : signifie **Weight round Robin**, il indique la charge qui sera mise sur les 3 serveurs

Si nous exécutons la commande **ipvsadm -ln** à ce stade, nous devrions constater que les trois serveurs ont effectivement traité des requêtes conformément à la configuration préalablement établie.

```
root@lb1:~# 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 wrr
  -> 192.168.200.11:80           Masq    3      0        0
  -> 192.168.200.12:80           Masq    1      0        0
  -> 192.168.200.13:80           Masq    1      0        0
root@lb1:~#
```

4. Vérification du bon fonctionnement

Systemctl stop ipvsadm

systemctl restart heartbeat.service

- Vérifier les interfaces réseaux

```
root@lb1:~# 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 08:00:27:32:dc:a7 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:fe32:dca7/64 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 08:00:27:1f:d6:2f 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:fe1f:d62f/64 scope link
        valid_lft forever preferred_lft forever
root@lb1:~#
```

Lancer un navigateur et vérifier : **192.168.56.10**