La haute disponibilité

1. Introduction à la Haute Disponibilité 4
1.1 Définition et Objectif 4
1.2 La Répartition de Charge (Load Balancing) 4
1.3 La Redondance de Serveur (Failover et Heartbeat) 4
1.4 Le Principe de l'Adresse IP Flottante (VIP) 4
2. Architecture réseau à réaliser 6
3. Phase 1 : Préparation des Serveurs Web (Apache) 7

3.1 Configuration de de la machine web1 7
3.2 Installation du Service HTTP 8
3.3 Personnalisation pour Identification 9
4. Phase 2 : Mise en place du Failover (Heartbeat) 11
4.1 Installation 12
4.2 Configuration des fichiers Heartbeat 12
4.3 Résolution de noms (/etc/hosts) 13
4.4 Démarrage et Test 14
5. Phase 3 : Mise en place du Load Balancing (LVS/IPVS) 15
5.1. Configuration de la machine Load Balancer (LB) 15
5.2. Activation du routage (IP Forwarding) 16
Vérification de l'activation du routage : 17
5.3. Installation et Configuration d'IPVSADM 17
Configuration des fichiers ipvsadm : 17
5.4. Création des règles de répartition 18
5.5 Vérification du paramétrage 19

1. Introduction à la Haute Disponibilité

1.1 Définition et Objectif

La **Haute Disponibilité** (ou *High Availability* en anglais) désigne l'ensemble des dispositions et techniques mises en œuvre pour garantir qu'un service reste accessible et opérationnel 24 heures sur 24, réduisant ainsi au minimum les interruptions de service.

1.2 La Répartition de Charge (Load Balancing)

Pour assurer la disponibilité et la performance, les requêtes des clients peuvent être distribuées sur plusieurs serveurs proposant le même service. Deux méthodes principales existent :

- **Round Robin DNS :** Il s'agit d'une méthode simple qui redirige les requêtes des clients de manière équitable et circulaire sur l'ensemble des serveurs, sans se soucier de l'état de charge de ces derniers.
- **Load Balancer (Répartiteur de charge) :** C'est un dispositif plus intelligent (matériel ou logiciel) qui distribue le trafic en prenant en compte des critères dynamiques, tels que la puissance des machines ou le nombre d'utilisateurs déjà connectés, afin d'optimiser les performances globales.

1.3 La Redondance de Serveur (Failover et Heartbeat)

Ce mécanisme repose sur la création d'une **grappe de serveurs** (ou *cluster*), où chaque serveur est appelé un **nœud** (*node*).

- **Principe Actif/Passif :** Dans cette architecture, un logiciel de battements de cœur (**Heartbeat**) est installé sur le serveur maître (Actif) et sur le serveur secondaire (Passif).
- **Surveillance :** Le serveur passif surveille en permanence les signaux (battements) émis par le serveur actif.
- **Basculement (Failover) :** Si le serveur actif cesse d'émettre des battements (panne), le serveur secondaire détecte l'arrêt et prend automatiquement le relais pour devenir le serveur actif.

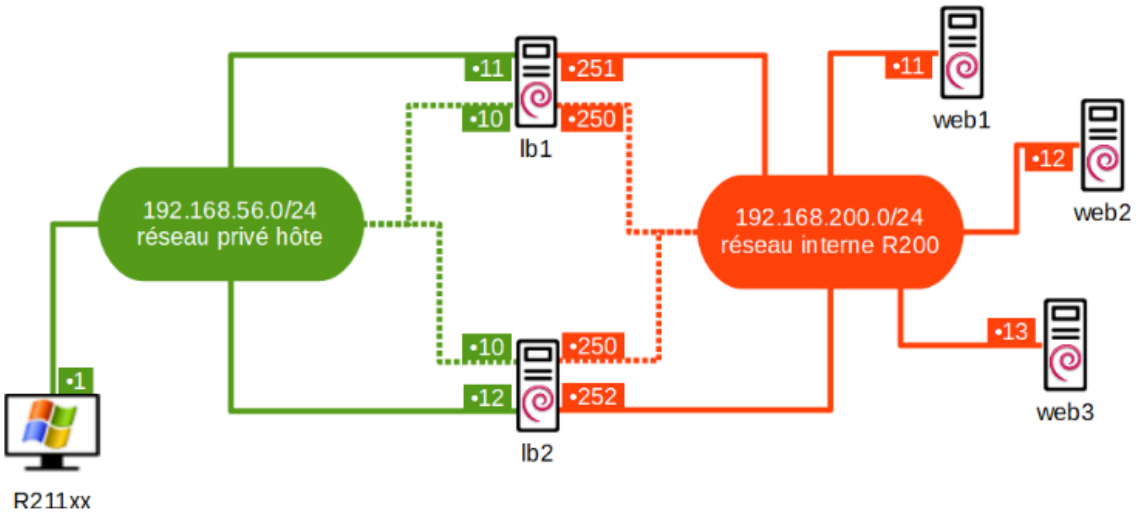
1.4 Le Principe de l'Adresse IP Flottante (VIP)

Pour rendre le basculement transparent pour les utilisateurs, on utilise une **adresse IP virtuelle** (ou flottante) :

- Chaque serveur dispose de sa propre adresse IP physique fixe.
- En complément, le nœud actif porte l'adresse IP virtuelle. **C'est cette adresse unique que les clients utilisent pour accéder au service.**
- Lors d'un basculement, le serveur secondaire s'attribue instantanément cette adresse flottante en devenant actif. Ainsi, les clients continuent d'accéder au service via la même adresse, sans avoir à être reconfigurés, bien que le traitement soit désormais effectué par une autre machine.

2. Architecture réseau à réaliser

Voici le schéma logique de l'infrastructure que nous allons déployer. Elle comprend des nœuds Web redondants et un répartiteur de charge.



Configuration Réseau Initiale (Phase 1 & 2) :

- Réseau Privé Hôte : `192.168.56.0/24`
- Web1 : `192.168.56.11`
- Web2 : `192.168.56.12`
- IP Virtuelle (VIP) : `192.168.56.10`

3. Phase 1 : Préparation des Serveurs Web (Apache)

Cette étape consiste à configurer deux serveurs identiques (clones) sous Debian Bullseye.

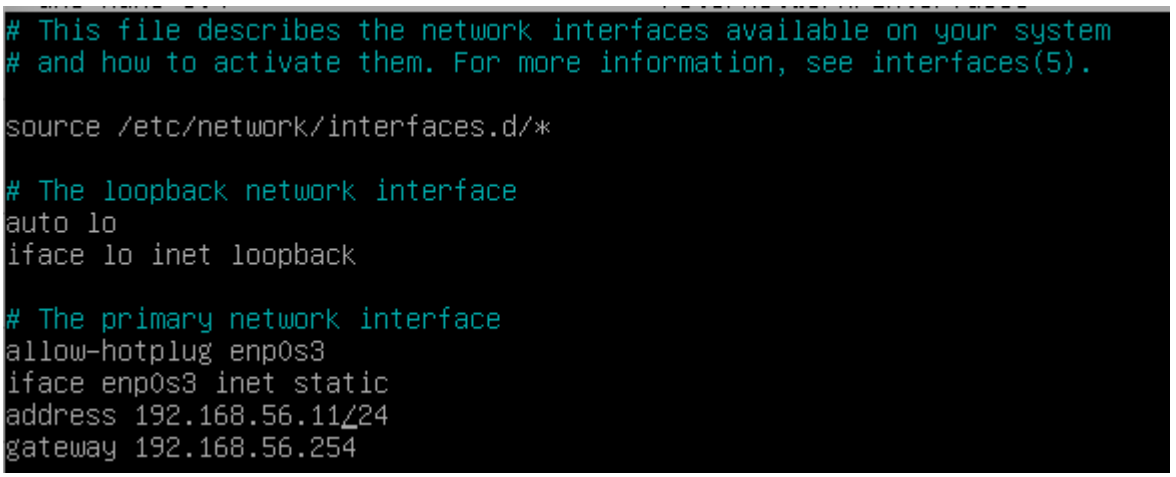
3.1 Configuration de de la machine web1

Sur chaque machine, nous devons définir une adresse IP statique.

Sur le serveur web1 : Ouvrez le fichier de configuration :

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

Modifiez la configuration comme suit :



Sur le serveur web2 : Répétez l'opération en adaptant l'IP :

```
# 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
```

puis ouvrir le fichier correspondant au nom de la machine sur les deux VM :

```
nano /etc/hostname
```

Sur web1 :

```
  GNU nano 5.4                                    /etc/hostname
web1
```

Sur web2 :

```
  GNU nano 5.4                                    /etc/hostname
web2
```

Appliquez les changements avec un `reboot`.

3.2 Installation du Service HTTP

Installez le serveur web Apache2 sur les deux machines :

```
apt update
apt install apache2
```

Sur web1 :

```
root@web1:~# apt update
Atteint :1 http://deb.debian.org/debian bullseye InRelease
Atteint :2 http://deb.debian.org/debian bullseye-updates InRelease
Réception de :3 http://security.debian.org/debian-security bullseye-security InRelease
Réception de :4 http://security.debian.org/debian-security bullseye-security/main Sour
Réception de :5 http://security.debian.org/debian-security bullseye-security/main amd6
1 kB]
Réception de :6 http://security.debian.org/debian-security bullseye-security/main Trans
5 kB]
973 ko réceptionnés en 1s (939 ko/s)
Lecture des listes de paquets... Fait
Construction de l'arbre des dépendances... Fait
Lecture des informations d'état... Fait
15 paquets peuvent être mis à jour. Exécutez « apt list --upgradable » pour les voir.
```

```
root@web1:~# apt install apache2
Lecture des listes de paquets... Fait
Construction de l'arbre des dépendances... Fait
Lecture des informations d'état... Fait
apache2 est déjà la version la plus récente (2.4.65-1~deb11u1).
Le paquet suivant a été installé automatiquement et n'est plus nécessaire :
  linux-image-4.19.0-18-amd64
Veuillez utiliser « apt autoremove » pour le supprimer.
0 mis à jour, 0 nouvellement installés, 0 à enlever et 15 non mis à jour.
root@web1:~# _
```

Sur web2 :

```
root@web2:~# apt update
Atteint :1 http://security.debian.org/debian-security bullseye-security InRelease
Atteint :2 http://deb.debian.org/debian bullseye InRelease
Atteint :3 http://deb.debian.org/debian bullseye-updates InRelease
Lecture des listes de paquets... Fait
Construction de l'arbre des dépendances... Fait
Lecture des informations d'état... Fait
15 paquets peuvent être mis à jour. Exécutez « apt list --upgradable » pour les voir.
```

3.3 Personnalisation pour Identification

Pour tester la haute disponibilité, il est crucial de savoir quel serveur répond. Modifiez la page par défaut sur chaque serveur.

Sur Web1 :

```
nano /var/www/html/index.html
```

Ajoutez ou modifiez le titre pour afficher : `Apache2 Page de Web1 - IP 192.168.56.11`.

```
      Apache2 Page de Web1
    </span>
  </div>
-     <div class="table_of_contents floating_element">
    <div class="section_header section_header_grey">
      TABLE OF CONTENTS
    </div>
    <div class="table_of_contents_item floating_element">
      <a href="#about">About</a>
    </div>
    <div class="table_of_contents_item floating_element">
      <a href="#changes">Changes</a>
    </div>
    <div class="table_of_contents_item floating_element">
      <a href="#scope">Scope</a>
    </div>
    <div class="table_of_contents_item floating_element">
      <a href="#files">Config files</a>
    </div>
  </div>

  <div class="content_section floating_element">


    <div class="section_header section_header_red">
      <div id="about"></div>
      It works! l'adresse IP est 192.168.56.11
```

Sur Web2 :

`nano /var/www/html/index.html`

Ajoutez ou modifiez le titre pour afficher : `Apache2 Page de Web2 - IP 192.168.56.12`.

```
    <span class="floating_element">
      Apache2 Page de Web2
    </span>
  </div>
      <div class="table_of_contents floating_element">
    <div class="section_header section_header_grey">
      TABLE OF CONTENTS
    </div>
    <div class="table_of_contents_item floating_element">
      <a href="#about">About</a>
    </div>
    <div class="table_of_contents_item floating_element">
      <a href="#changes">Changes</a>
    </div>
    <div class="table_of_contents_item floating_element">
      <a href="#scope">Scope</a>
    </div>
    <div class="table_of_contents_item floating_element">
      <a href="#files">Config files</a>
    </div>
  </div>

  <div class="content_section floating_element">


    <div class="section_header section_header_red">
      <div id="about"></div>
      It works! l'adreese IP 192.168.56.12
    </div>
    <div class="content_section_text">
```

4. Phase 2 : Mise en place du Failover (Heartbeat)

Heartbeat permet de gérer le basculement automatique. Si le nœud actif tombe, le passif récupère l'IP virtuelle.

4.1 Installation

Installez le paquet sur **Web1** et **Web2** :

```
apt update
apt install heartbeat
```

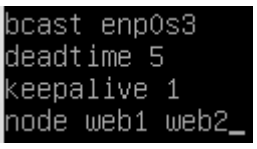
4.2 Configuration des fichiers Heartbeat

Trois fichiers clés doivent être configurés dans `/etc/ha.d/` sur les deux machines.

A. Fichier de configuration principale (`ha.cf`) Ce fichier définit les paramètres de communication du cluster.

```
nano /etc/ha.d/ha.cf
```

```
bcast enp0s3          # Interface de diffusion des heartbeats
deadtime 5            # Délai avant de considérer un nœud mort
keepalive 1           # Fréquence des battements (secondes)
node web1 web2        # Noms des nœuds du cluster
```



B. Fichier d'authentification (`authkeys`) Ce fichier sécurise les échanges entre les nœuds.

```
nano /etc/ha.d/authkeys
```

Ajoutez :

```
auth 1
1 sha1 motdepasse
```

auth 1
1 sha1 motdepasse # Remplacez 'motdepasse' par une clé forte

Important : Sécurisez ce fichier immédiatement :

```
root@web2:/etc/ha.d# chmod 600 /etc/ha.d/authkeys
```

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

C. Fichier de ressources (`haresources`) Ce fichier définit qui est le maître par défaut et quelle est l'IP virtuelle.

`nano /etc/ha.d/haresources`

Ajoutez la ligne suivante (identique sur les deux serveurs) :

```
web1    IPaddr::192.168.56.10   apache2
```

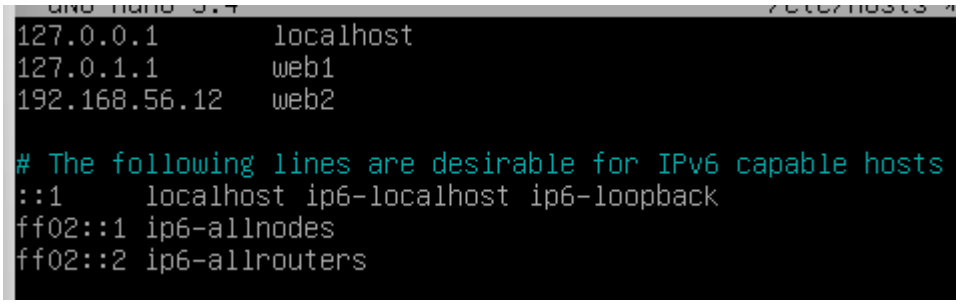
web1 IPaddr::192.168.56.10 apache2

Cela signifie que `web1` gère l'IP `192.168.56.10` et le service `apache2` par défaut.

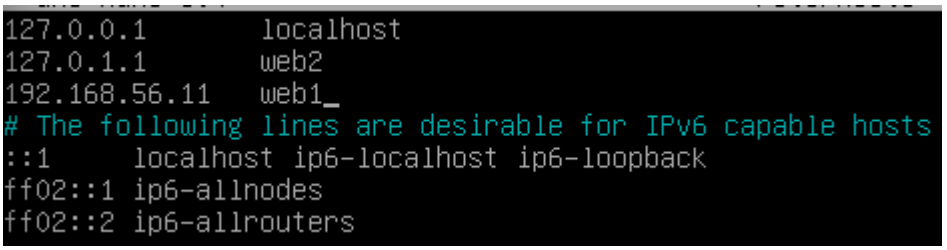
4.3 Résolution de noms (`/etc/hosts`)

Assurez-vous que les serveurs se connaissent par nom. Éditez `/etc/hosts` sur les deux machines pour inclure :

192.168.56.11 web1



192.168.56.12 web2



4.4 Démarrage et Test

Démarrez le service Heartbeat sur les deux machines :

`systemctl` restart heartbeat.service

Sur web1 :

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

Sur web2 :

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

Pour tester, arrêtez Apache ou éteignez le serveur Web1 et vérifiez que Web2 prend le relais sur l'adresse virtuelle `192.168.56.10`.

5. Phase 3 : Mise en place du Load Balancing (LVS/IPVS)

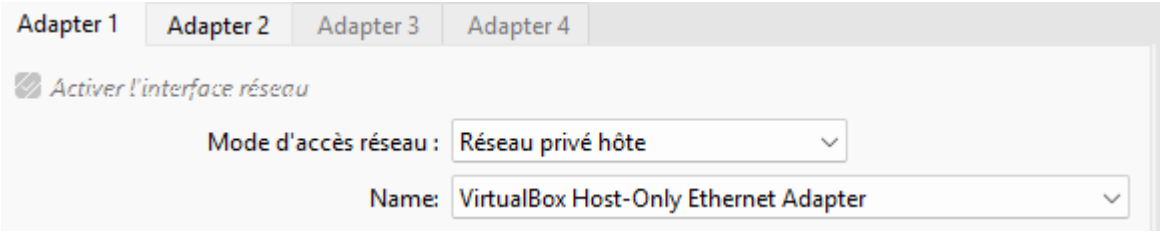
Dans cette configuration, nous utilisons un répartiteur de charge (Load Balancer) pour distribuer le trafic vers des serveurs réels.

5.1. Configuration de la machine Load Balancer (LB)

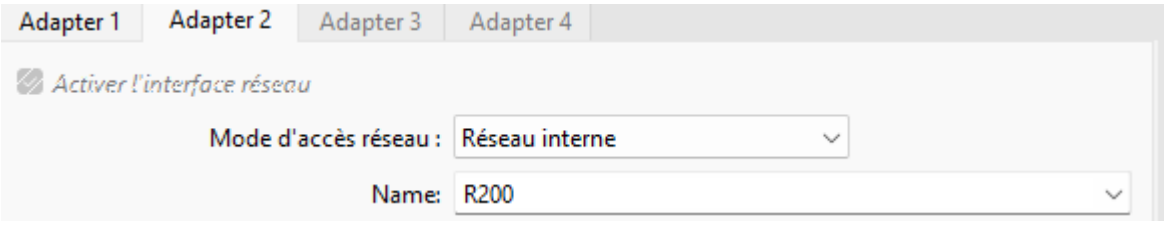
Cette machine nécessite deux interfaces réseau:

1. **Interface externe (VIP) :** `192.168.56.10` (Réseau privé hôte)
2. **Interface interne (DIP) :** `192.168.200.254` (Réseau interne)

Dans l'interface 1 :



Dans l'interface 2 :



Configuration réseau :

```
  GNU nano 5.4                                /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 anp0s8 inet static
address 192.168.200.254/24
```

5.2. Activation du routage (IP Forwarding)

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**. Si la ligne n'existe pas, ajoutez-la à la fin du fichier :

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

```
# Uncomment the next line to enable packet forwarding for IPv4
net.ipv4.ip_forward=1
```

Vérification de l'activation du 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, avec la commande "**`reboot`**"

```
root@debian0:~# cat /proc/sys/net/ipv4/ip_forward
1
```

Le **1** signifie que le **routage** est bien **activé**.

5.3. Installation et Configuration d'IPVSADM

Installez l'outil d'administration LVS :

apt install ipvsadm

```
root@lb:~# apt install ipvsadm_
```

Configuration des fichiers ipvsadm :

Configurez le démarrage automatique dans `/etc/default/ipvsadm` :

AUTO="true"
DAEMON="master"
IFACE="enp0s3"

```
# 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"
```

5.4. Création des règles de répartition

Nous allons configurer le service pour utiliser l'algorithme **Round Robin (rr)** et le mode **Masquerade (NAT)**.
Éditez ou créez le fichier de règles :

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

Les commandes suivantes définissent le cluster :

1. **Ajout du service virtuel (VIP) :**
   - `-A` : Ajouter un service
   - `-t` : Protocole TC
   - `-s rr` : Algorithme Round Robin

```
# emtpy rules file for ipvsadm
# 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 192.168.200.11:80 -m
```

2. **Ajouts des noeuds (Serveurs Réels) :**
   - `-a` : Ajouter un serveur réel

- `-r` : Adresse du serveur réel (ex: 192.168.200.11)
- `-m` : Masquerading (NAT)

5.5 Vérification du paramétrage

On vérifie le paramétrage avec la commande **`ipvsadm -ln`** :

```
root@debian0:~# 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
```