

---

# LA HAUTE DISPONIBILITÉ

---

« On appelle « haute disponibilité » toutes les dispositions visant à garantir la disponibilité d'un service et son bon fonctionnement 24H/24 ».

## Table des matières

<b>Partie 1 : Cours</b>	<b>1</b>
Principe général	1
Éléments vitaux à surveiller	1
Fiabilité et disponibilité	2
MTBF (" Mean Time Between Failure ")	2
MTTR (" Mean Time To Repair ")	2
La méthode des 9	2
Les outils	2
Mode dégradé	4
PCA ? PRA ?	4
La sauvegarde vs l'archivage	5
L'archivage	5
La sauvegarde des données	5
Différents types des sauvegardes	5
Redondance de serveurs	5
Round Robin DNS	5
Load balancer ou répartiteur de charges	6
Heartbeat	6
Principe et vocabulaire technique	6
Principe et adresse flottante	7
<b>Partie 2 : TP</b>	<b>8</b>
<b>Objectif : Mettre en place Heartbeat entre deux serveurs web</b>	<b>8</b>
Commandes utiles	8
Schéma du réseau virtuel à effectuer	9
Mise en place de Heartbeat	9
Création des serveurs web	9
Configuration des serveurs web	10
Installation et configuration de Heartbeat	10
Modification des fichiers de configurations de Heartbeat	10
<b>Objectif : Mettre en place un Load-balancer pour faire de la répartition des charges</b>	<b>13</b>
Schéma du réseau virtuel à réaliser	13
Installation des serveurs web et du serveur load-balancer	14
Configuration des interfaces réseaux des trois serveurs	14
Modification des fichiers de configuration du service ipvsadm	15
Activation du routeur	15
Modification des fichiers de configuration	15

<b>Objectifs: Ajouter un troisième serveur web dans le réseau interne R200</b>	<b>15</b>
Schéma du réseau virtuel à effectuer	15
Modifications pour ajouter le nouveau serveur web	16
<b>Objectifs : Mise en place d'un second serveur lb et appliquer la méthode de Heartbeat entre ces deux serveurs</b>	<b>16</b>
Schéma de l'infrastructure à mettre en place	17
Mise en place d'un second serveur load-balancer	17
Installation et configuration de Heartbeat	1
Référencement des différents serveurs dans /etc/hosts pour les serveurs lb	1
Configuration des interfaces réseaux sur chacun des serveurs	1

## Partie 1 : Cours

### Principe général

- High Availability (HA) en anglais.
- Désigner le fait qu'une architecture ou un service à un taux de disponibilité convenable.
- Enjeu important car une indisponibilité entraîne des coûts très élevés.

Panne de Facebook, Instagram, Messenger : Des personnes travaillent via ces réseaux (Community Manager), elles ont donc été privées de leurs boulots. Le problème est que ces personnes se doivent d'animer constamment leurs pages pour conserver leur e-réputation.

### Éléments vitaux à surveiller

- La disponibilité des services : Outils indispensables pour le fonctionnement d'un site web marchand.
- La disponibilité des données : Intégrité des données □ perte de données impensable !
- La tolérance aux catastrophes : Probabilité faible mais risque à ne pas négliger.

### Fiabilité et disponibilité

#### *MTBF (" Mean Time Between Failure ")*

- Temps moyen entre 2 pannes.
- Permet d'avoir une indication sur la durée de vie espérée d'un composant (matériel et logiciel).
- Permet d'évaluer le temps qui s'écoule jusqu'à l'arrêt d'un service ou à la panne d'un composant ou d'un logiciel.

#### *MTTR (" Mean Time To Repair ")*

- Permet de connaître l'intervalle de temps ou un service est indisponibles c'est-à-dire jusqu'à son rétablissement.

Pour obtenir un système fiable, 2 leviers d'action :

- Soit obtenir un **MTBF fort**, c'est-à-dire que l'intervalle entre deux pannes du système est grand.
- Soit obtenir un **MTTR faible**, c'est-à-dire que le temps pour rétablir mon système est le plus court possible.

#### La méthode des 9

- Autre demande pour évaluer le niveau de disponibilité.
- Consiste à ne pas tenir compte de la fréquence des pannes mais uniquement de leur durée.
- Il s'agit d'évaluer la durée d'arrêt cumulée du service sur un an.

Disponibilité	Indisponibilité (min/an)	Commentaires
90.0%	52 560 min (36,5 jours)	Pas de service
99.0%	5 256 min (3, 65 jours)	Service fournit
99.9%	526 min (9 heures)	Bon niveau de service
99.99%	52,6 min	Tolérant aux pannes
99.999%	5,26 min	Hautement disponible
99.9999%	0,53 min (31 secondes)	Très hautement disponible
99.99999%	0,053min (3 secondes)	Ultra disponible

Disponibilité en %	Indisponibilité par année	Indisponibilité par mois <sup>3</sup>	Indisponibilité par semaine
90 % (« un neuf »)	36,5 jours	72 heures	16,8 heures
95 %	18,25 jours	36 heures	8,4 heures
98 %	7,30 jours	14,4 heures	3,36 heures
99 % (« deux neuf »)	3,65 jours	7,20 heures	1,68 heure
99,5 %	1,83 jour	3,60 heures	50,4 minutes
99,8 %	17,52 heures	86,23 minutes	20,16 minutes
99,9 % (« trois neuf »)	8,76 heures	43,2 minutes	10,1 minutes
99,95 %	4,38 heures	21,56 minutes	5,04 minutes
99,99 % (« quatre neuf »)	52,56 minutes	4,32 minutes	1,01 minute
99,999 % (« cinq neuf »)	5,26 minutes	25,9 secondes	6,05 secondes
99,9999 % (« six neuf »)	31,5 secondes	2,59 secondes	0,605 seconde

Remarque : Il est évident que l'on évitera de parler de haute disponibilité en dessous de 3 neuf...

#### Les outils

- **Onduleurs (UPS : Uninterruptible Power System)** : si la panne doit durer, il faut s'assurer que l'onduleur est capable d'arrêter proprement le serveur via un signal.
- **Alimentation redondante** : 2 ou 3 alimentations pour se protéger en cas de défaillance de l'alimentation principale.

- **RAID** : technique de tolérance de panne qui permet de stocker des informations sur plusieurs disques, elle permet une répartition des données sur plusieurs disques ce qui a pour avantages au système de supporter la perte d'un disque dur selon le RAID utilisé.
  - **Cartes réseaux additionnelles** : on est capable de créer une interface réseau virtuelle qui va regrouper plusieurs interfaces physiques grâce au Channel Bonding (Agrégation de lien).
  - **Spanning tree** : protocole STP utilisé pour éliminer les boucles de pontage dans les réseaux locaux Ethernet et désactiver les liens redondants dans un réseau Ethernet pour éviter des tempêtes de broadcast. STP peut réactiver un lien momentanément désactivé afin de le remplacer en cas de réelle panne physique.
  - **Redondance de passerelles** : l'entreprise va s'assurer qu'elle a toujours une connexion Internet valable.
  - **Réplication des bases de données** : permet de basculer facilement les données d'une base sur une autre machine.
  - **Changement à chaud des périphériques (Hotplug)** : est-on capable de brancher/débrancher un disque dur à la suite à une panne ou est-on obligé d'arrêter le système (changement à froid) ?
  - **Climatisation et hygrométrie** : on évite d'installer la salle serveur dans un sauna ou un hammam...
  - **Surveillance de l'état du système** : on va surveiller la température des différents composants ainsi que le bon fonctionnement des ventilateurs □ Monitoring.
  - **Redémarrage à distance de la machine** : notamment grâce au Wake-On-Lan (réveil par réseau).
  - **Accès Distant** : tunnel SSH.
  - **Remontée des événements** : à vous de mettre en place les bons outils (à vos scripts s'ils n'existent pas déjà) afin de surveiller vos systèmes RAID, Channel Bonding et autres
  - **Sauvegardes** : complètes, différentielles, incrémentielles.
  - **Mode dégradé**
  - **Plan de secours, plan de continuité d'activité (PCA), plan de reprise d'activité (PRA)**
1. Hot-Swapping : changement à chaud d'un disque de secours en cas de panne.
  2. Spare : disque de secours pour prendre le relais d'un disque en panne.
  3. Agrégation de lien : technique regroupant plusieurs ports réseaux et permettant de les utiliser comme s'il y en avait qu'une seule.
  4. Tempête de broadcast : saturation d'un réseau entraînant un blocage.

#### *Mode dégradé*

- Initialement un langage militaire
- Désigne les situations où tout une partie d'une entité organisée (armée, entreprise, système, gouvernement, groupe humain, hôpital, voire exceptionnellement tout un continent ou la planète...) doivent (ou devraient) fonctionner sans leurs ressources habituelles, humaines et matérielles.
- **Exemple** : guerre, grave attentat, catastrophe majeure (technologique ou naturelle), accident nucléaire, tremblement de terre, tsunami majeur, épidémie ou pandémie grave.
- Tenter de fournir le service jugé indispensable, en manquant de ressources complètes ou fiables ou régulières en énergie (dont électrique), en transport, télécommunications, etc.

- Pour réagir au mieux et retrouver au plus vite une situation normale ou « restaurée », les acteurs vitaux généralement invités à se préparer à fonctionner en « mode dégradé », par exemple et notamment dans le cadre des plans de continuité.

#### PCA ? PRA ?

- Plan de continuité d'activité (PCA)
- Plan de reprise d'activité (PRA)
- Le PRA est **complémentaire** du PCA
- Le plan de continuité d'activité organise la poursuite des activités de l'entreprise en cas d'incident.
- Le plan de reprise d'activité anticipe une interruption de l'activité et prévoit les conditions de sa reprise.

#### Pourquoi mettre en place un PRA :

- La reprise d'une activité, même partielle, garantit un niveau de Chiffre d'Affaires minimum, et participe donc de la survie de l'entreprise.
- Une entreprise capable de satisfaire ses clients, même en période de crise, fidélise autant qu'elle améliore son image.
- La bonne gestion du fonctionnement de l'entreprise en période de crise permet également de fidéliser les collaborateurs et de fluidifier l'organisation interne de l'entreprise.
- En assurant une reprise rapide de son activité, l'entreprise s'engage également à répondre à d'éventuelles obligations légales.

#### Comment faire un PRA ? Intégration de :

- Un état des lieux des enjeux et des besoins de l'entreprise.
- Le listing des activités-clés pour le bon fonctionnement de l'entreprise
- L'identification des incidents possibles.
- Les actions préalables à mener pour limiter l'impact de ces incidents sur les activités-clés.
- Les ressources-clés (notamment les ressources humaines) indispensables à la réalisation des activités-clés
- La démarche et les étapes à suivre pour remettre en route l'activité, notamment en cas de reprise progressive.

#### **Local adapté**

Travail à faire : Réalisez un diaporama où vous mettrez en exergue toute l'infrastructure matérielle et informatique nécessaire afin de sécuriser les données numériques dans un local technique.

#### La sauvegarde vs l'archivage

##### L'archivage

Archiver est une opération consistant à assurer la conservation d'un document, quel que soit son support, en vue d'une consultation ultérieure, à titre de preuve ou d'information.

	Sauvegarde	Archivage électronique
Objectif	Restauration des données	Conservation de l'information (document de référence ou archive probante)
Gestion des risques	Limite le risque de perte et de corruption des données et des systèmes.	Limite le risque de contentieux et garantit la conformité à la réglementation fiscale.
Accès	<ul style="list-style-type: none"> <li>– Fréquence imprévisible car dépend d'un éventuel incident.</li> <li>– Souvent urgent : la comptabilité doit continuer « à vivre » en temps réel.</li> <li>– Accès au niveau de la base de données ou du système</li> </ul>	<ul style="list-style-type: none"> <li>– Fréquent pour les archives récentes</li> <li>– Occasionnel pour les archives intermédiaires</li> <li>– Imprévisible pour les archives anciennes</li> <li>– Accès au niveau du dossier ou du document</li> </ul>
Sécurité d'accès	<ul style="list-style-type: none"> <li>– Au niveau du jeu de sauvegarde</li> <li>– Accès par le personnel de l'exploitation informatique relevant de la direction des systèmes d'information (DSI)</li> </ul>	<ul style="list-style-type: none"> <li>– Au niveau de la base d'archivage et au niveau des catégories d'archives</li> <li>– Accès par le personnel du service producteur ou concerné</li> </ul>
Conservation	Conçu pour une conservation à court terme	<ul style="list-style-type: none"> <li>– Conçu pour une conservation à long terme</li> <li>– Doit pouvoir être relu par l'Administration fiscale</li> <li>– Conservation pendant les durées requises par le législateur</li> </ul>

## La sauvegarde des données

- La législation impose un archivage des données
- <https://entreprendre.service-public.fr/vosdroits/F10029>
- Sauvegarder des données est une opération qui consiste à en faire une copie, afin de pallier leur éventuelle destruction, totale ou partielle (conséquence d'une catastrophe naturelle, d'un sabotage, de l'attaque d'un virus, d'une défaillance du système informatique « plantage », d'une mauvaise manipulation...)
- Une sauvegarde permet de **restaurer les données en cas de panne.**

### L'organisation des sauvegardes :

La sauvegarde doit être **effectuée régulièrement** et la **restauration** de la sauvegarde doit également être testée afin de vérifier que l'ensemble de la procédure est parfaitement opérationnel.

### Dans la plupart des cas, la procédure est automatisée et prévoit :

- La **périodicité** des sauvegardes (tous les jours, toutes les heures...) ;
- Le **type** de sauvegarde ;
- Le **nombre d'exemplaires** conservés des supports de sauvegarde ;
- Le **lieu** de stockage des supports de sauvegarde, dans l'idéal hors des locaux ;
- La **rotation** des supports éventuellement retenue ;
- Le(s) salarié(s) **responsable(s)** de cette mission.

## Différents types des sauvegardes

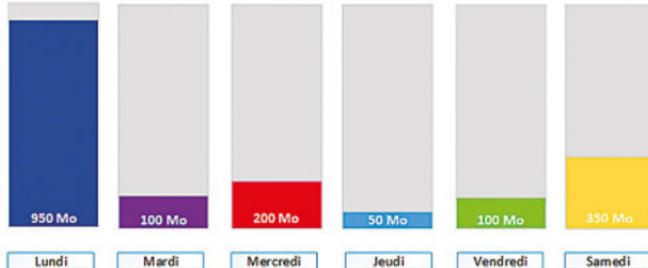
Sauvegarde complète	Il s'agit du premier type de sauvegarde forcément mis en œuvre dans une organisation. Toutes les données du périmètre prévu sont dupliquées lors d'une sauvegarde complète. Si cette méthode est la plus simple, elle peut être très longue selon le volume de données à sauvegarder. C'est pour remédier à cette difficulté que d'autres types de sauvegarde peuvent être utilisés de façon complémentaire.
---------------------	--

Sauvegarde incrémentale

Seules les données modifiées depuis la **dernière sauvegarde**, quel que soit son type (complète, différentielle ou incrémentale) sont sauvegardées. Cela permet un gain de temps significatif car le volume à sauvegarder reste limité, mais la restauration des données nécessitera de restaurer dans un premier temps la dernière sauvegarde complète, puis chaque sauvegarde incrémentale postérieure à la sauvegarde complète.

**Exemple**

En cas d'incident le vendredi, il faudra restaurer la sauvegarde complète du lundi, puis la sauvegarde incrémentale du mardi, celle du mercredi, et celle du jeudi.



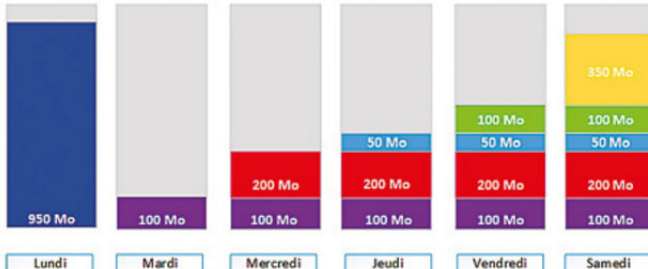
Jour	Type de sauvegarde	Valeur
Lundi	Sauvegarde complète	950 Mo
Mardi	Sauvegarde incrémentale	100 Mo
Mercredi	Sauvegarde incrémentale	200 Mo
Jeudi	Sauvegarde incrémentale	50 Mo
Vendredi	Sauvegarde incrémentale	100 Mo
Samedi	Sauvegarde incrémentale	350 Mo

Sauvegarde différentielle

Seules les données modifiées depuis la **dernière sauvegarde complète** sont sauvegardées. Le volume à sauvegarder augmente donc progressivement, ce qui nécessite jour après jour de plus en plus de temps et d'espace de stockage. La restauration nécessitera de restaurer dans un premier temps la dernière sauvegarde complète, puis seulement la dernière sauvegarde différentielle, ce qui est moins fastidieux qu'avec une sauvegarde incrémentale.

**Exemple**

En cas d'incident le vendredi, il faudra restaurer la sauvegarde complète du lundi, puis la sauvegarde différentielle du jeudi.



Jour	Type de sauvegarde	Valeur
Lundi	Sauvegarde complète	950 Mo
Mardi	Sauvegarde différentielle	100 Mo
Mercredi	Sauvegarde différentielle	200 Mo
Jeudi	Sauvegarde différentielle	50 Mo
Vendredi	Sauvegarde différentielle	100 Mo
Samedi	Sauvegarde différentielle	350 Mo

Sauvegarde mixte

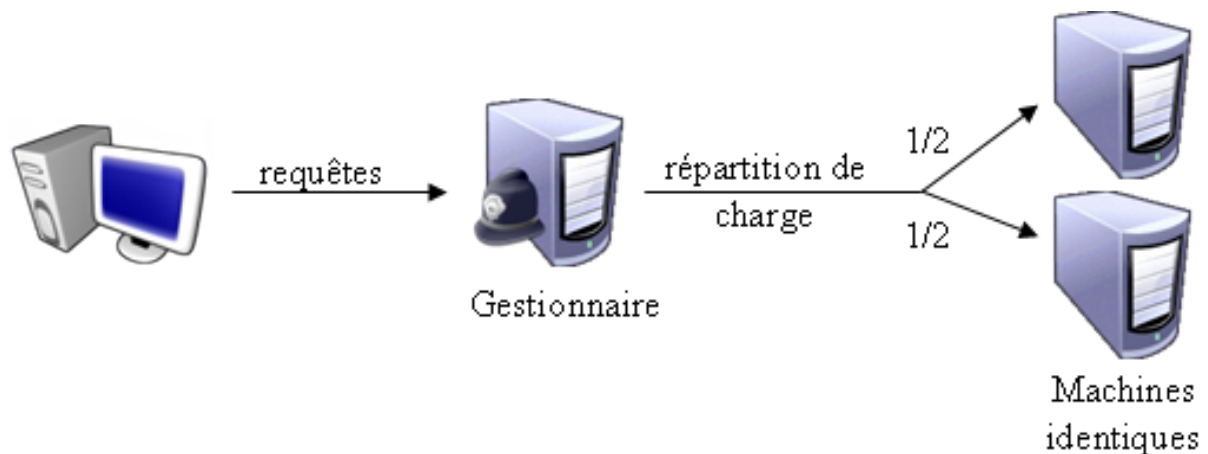
Il s'agit d'une combinaison des types de sauvegarde précédents. C'est le cas le plus fréquent pour maximiser la sécurité tout en tenant compte de contraintes temporelles et matérielles.

## Redondance de serveurs

### Round Robin DNS

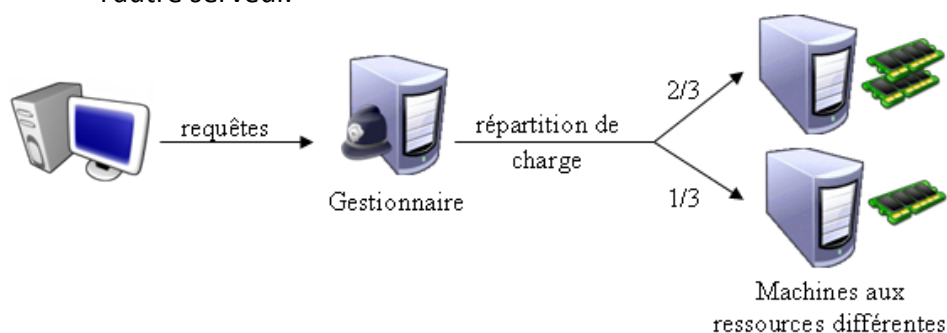
- Plusieurs serveurs proposant le même service
- Pouvoir répartir les requêtes des clients de manière équitable sur tous les serveurs





### Load balancer ou répartiteur de charges

- Prendre en compte de la puissance des machines, le nombre d'utilisateurs déjà connectés.
- Les clients envoient leurs requêtes au « **load-balancer** » qui se charge de les transmettre au cluster de serveurs.
- La charge de travail est donc répartie entre les différents serveurs car ils sont tous actifs simultanément.
- En cas de panne de l'un deux, le travail se portera sur le serveur restant.
- Le « load balancer », en isolant les serveurs du reste du réseau, augmente la sécurité des serveurs en les cachant à la vue des clients qui ne connaissent que l'adresse du « load-balancer », comme c'est le cas dans les DMZ.
- L'algorithme de répartition de charge dans l'exemple est le « Round-Robin » (tourniquet) qui attribue chaque nouvelle requête au serveur suivant disponible du cluster.
- Avec 2 serveurs, les requêtes seront donc attribuées alternativement à l'un et à l'autre serveur.



### Heartbeat

#### Principe et vocabulaire technique

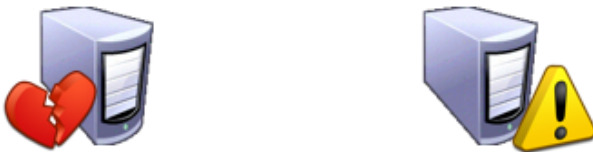
- Basculement (actif/passif) OU Failover
- Un logiciel « battement de cœur » (**Heartbeat**) est installé sur le serveur maître et sur chaque serveur secondaire.



- L'ensemble forme une **grappe** de serveurs ou **cluster**
- Chaque serveur du « cluster » est un « nœud » (node en anglais)
- Le serveur secondaire (**passif**) va surveiller en permanence les battements de cœur du serveur principal (**actif**)

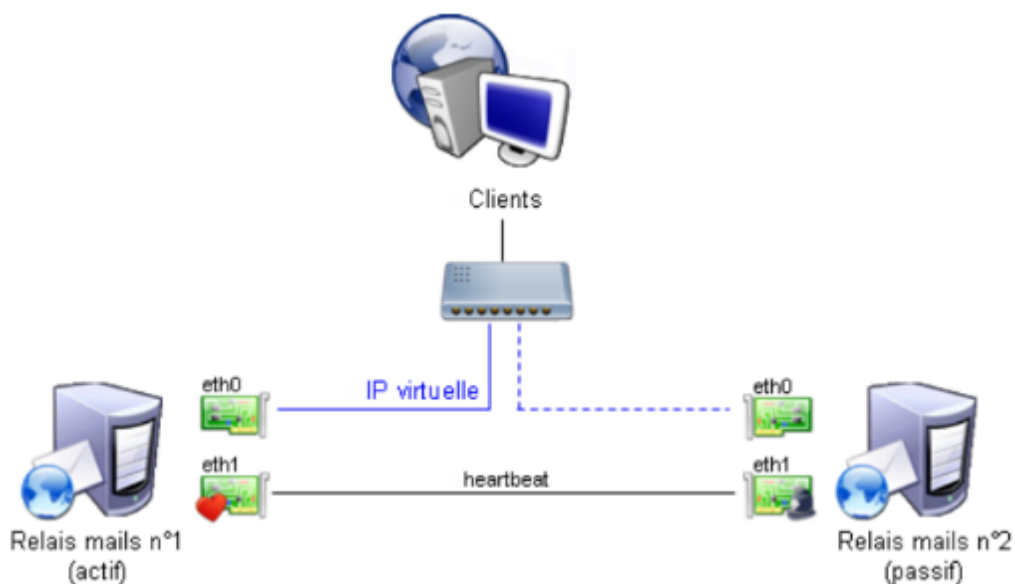


- ... et prendre le relais automatiquement en cas **d'arrêt** des **battements**. Il devient alors **actif**.



#### *Principe et adresse flottante*

- Chaque serveur possède sa propre adresse IP, mais seul le « **nœud** » actif possède, en plus, une adresse virtuelle : c'est par cette adresse « flottante » que les clients accèdent au service.



- **Le remplacement d'un nœud par l'autre s'est donc fait de façon transparente pour les clients.**
- Lorsque le serveur secondaire prend la place du maître, il s'attribue cette adresse flottante en devenant actif.
- Les clients continuent les accès sur cette adresse qui cette fois correspond donc au serveur secondaire.

## Partie 2 : TP

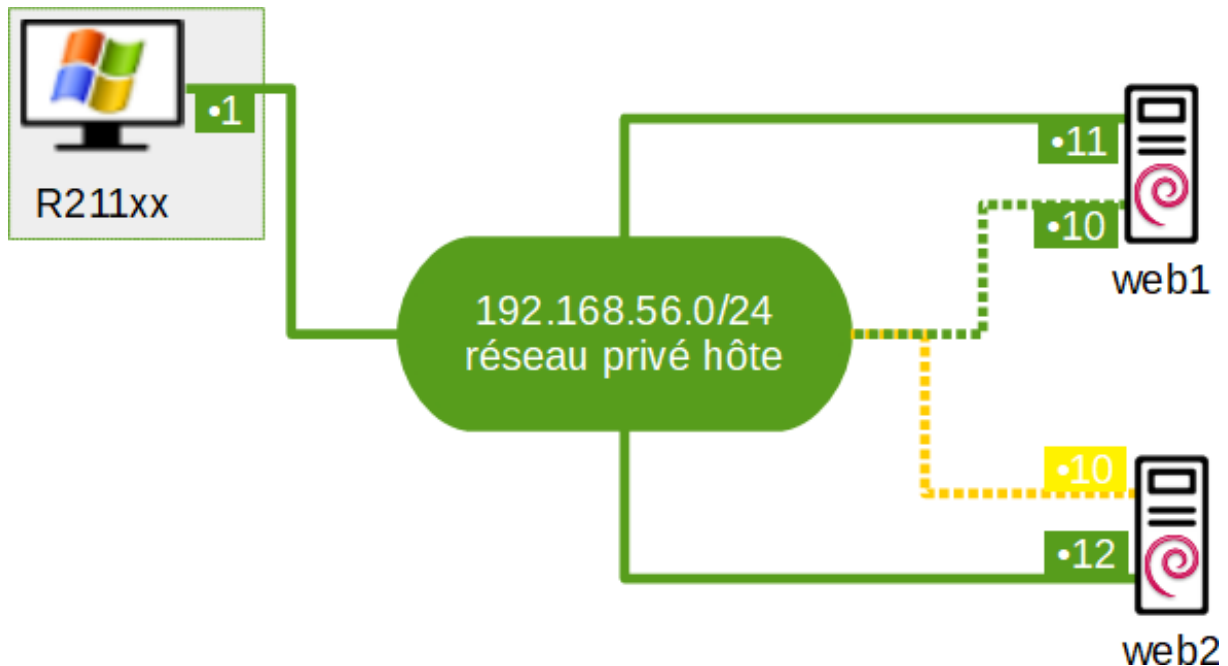
Objectif : Mettre en place Heartbeat entre deux serveurs web

- Créer deux serveurs web
- Configurer les deux serveurs
- Installer le service Heartbeat sur les deux serveurs
- Créer et configurer les différents fichiers de Heartbeat
- Modifier la page Apache
- Tester sur Internet le fonctionnement de Heartbeat

### Commandes utiles

1. Modifier les interfaces réseaux : `nano /etc/network/interfaces`
2. Modifier le DNS : `nano /etc/resolv.conf`
3. Modifier le Proxy : `nano /etc/apt/apt.conf`
4. Modifier un fichier : `nano`
5. Se déplacer entre les répertoires : `cd`
6. Mettre à jour les paquets : `apt update`
7. Installer un paquet : `apt install`
8. Redémarrer le paquet : `systemctl restart`
9. Modifier les autorisations d'un fichier : `chmod`
10. Afficher les interfaces réseaux : `ip a`
11. Modifier le nom de la VM : `nano /etc/hostname`
12. Modifier le fichier contenant les hôtes : `nano /etc/hosts`

### Schéma du réseau virtuel à effectuer



Nous allons ici tester le service Heartbeat, l'environnement mis en place se situera dans un Réseau Privé Hôte et contiendra 2 serveurs web.

## Mise en place de Heartbeat

### Création des serveurs web

On va créer deux serveurs web Debian Buster 10 et mettre ces deux machines en Réseau Privé Hôte. De plus, il faudra démarrer la VM tc-router afin de pouvoir installer des paquets.

☒ Activer l'interface réseau

Mode d'accès réseau : Réseau privé hôte

Nom : VirtualBox Host-Only Ethernet Adapter

[Avancé](#)

### Configuration des serveurs web

Modifier les interfaces réseaux de ces deux machines afin de leur attribuer des adresses IP et une passerelle afin d'être reliée à la VM tc-router

```
# The primary network interface
allow-hotplug enp0s3
iface enp0s3 inet static
address 192.168.56.11/24
gateway 192.168.56.254
```

Pour le serveur web 1

```
# The primary network interface
allow-hotplug enp0s3
iface enp0s3 inet static
address 192.168.56.12/24
gateway 192.168.56.254
```

Pour le serveur web 2

## Installation et configuration de Heartbeat

Tout d'abord il faudra installer Heartbeat qui permettra de garantir la disponibilité en cas de panne. Heartbeat utilise un processus de Fail-over, ce principe est le fait qu'un serveur passif prenne le relais d'un serveur actif. On testera plus tard cette disponibilité en arrêtant le fonctionnement de web 1.

```
root@web1:~# apt install heartbeat
```

### *Modification des fichiers de configurations de Heartbeat*

**ATTENTION :** Après l'installation de Heartbeat, ces 3 fichiers ne sont pas présents sur la machine ! Il faudra les créer.

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

```
bcast enp0s3
deadtime 5
keepalive 1
node web1 web2
```

- Bcast : Défini que la méthode de communication entre les nœuds est le broadcast sur l'interface enp0s3.
- Deadtime : Temps nécessaire avant de considérer qu'un nœud est mort. Le temps est en secondes par défaut. On ajoute 'ms' derrière pour l'avoir en millisecondes.
- Keepalive : Intervalle entre 2 battements de cœur. La valeur est en secondes par défaut. Pour la spécifier en millisecondes, on ajoute 'ms' derrière.

**Attention avec cette valeur :** si elle est trop courte, le système risque de s'auto-déclarer mort. Si elle est trop grande, l'autre machine mettra un temps conséquent avant de s'en apercevoir et de reprendre la main.

- Node : Liste des machines utilisées pour la haute disponibilité, séparées par des espaces.

 `/etc/ha.d/authkeys`

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

```
auth 1
1 md5 mot de passe
```

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

```
root@web1:~# chmod 600 /etc/ha.d/authkeys
```

## /etc/ha.d/haresources

Liste des ressources (adresses virtuelles et services concernés) fournies par le cluster. La configuration sur chacune des machines est la même. Ce nom doit être le même sur les 2 machines. C'est le nom de la machine qui sera activée par défaut au démarrage de Heartbeat.

```
web1      IPaddr::192.168.56.10  apache2_
```

Le serveur web1 est le serveur maître.

Nous allons déclarer les hôtes web1 et web2 pour qu'ils puissent se reconnaître dans /etc/hosts (excepté si un serveur DNS est installé).

```
127.0.0.1      localhost
127.0.1.1      web2
192.168.56.11  web1
```

```
127.0.0.1      localhost
127.0.1.1      web1
192.168.56.12  web2
```

Ensuite, nous allons pouvoir stopper le service apache2 sur les deux machines puis vérifier qu'il est bien arrêté.

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

```
• apache2.service - The Apache HTTP Server
  Loaded: loaded (/lib/systemd/system/apache2.service; enabled; vendor preset: enabled)
  Active: inactive (dead) since Thu 2022-11-24 15:08:21 CET; 49s ago
    Docs: https://httpd.apache.org/docs/2.4/
  Process: 298 ExecStart=/usr/sbin/apachectl start (code=exited, status=0/SUCCESS)
  Process: 717 ExecStop=/usr/sbin/apachectl stop (code=exited, status=0/SUCCESS)
 Main PID: 351 (code=exited, status=0/SUCCESS)

nov. 24 15:05:37 web1 systemd[1]: Starting The Apache HTTP Server...
nov. 24 15:05:37 web1 apachectl[298]: AH00558: apache2: Could not reliably determine the server
nov. 24 15:05:37 web1 systemd[1]: Started The Apache HTTP Server.
nov. 24 15:08:21 web1 systemd[1]: Stopping The Apache HTTP Server...
nov. 24 15:08:21 web1 apachectl[717]: AH00558: apache2: Could not reliably determine the server
nov. 24 15:08:21 web1 systemd[1]: apache2.service: Succeeded.
nov. 24 15:08:21 web1 systemd[1]: Stopped The Apache HTTP Server.
lines 1-15/15 (END)
```

Puis, désactiver le service apache2 ☞ Heartbeat le démarrera lui-même

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

On va désormais redémarrer le service Heartbeat car c'est lui qui va démarrer apache2 et grâce à lui on pourra retrouver le site via l'IP flottante.

```

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 Thu 2022-11-24 15:13:47 CET; 14s ago
     Docs: man:heartbeat(8)
           http://www.linux-ha.org/wiki/Documentation
  Main PID: 872 (heartbeat)
    Tasks: 4 (limit: 1149)
   Memory: 10.4M
   CGroup: /system.slice/heartbeat.service
           └─872 heartbeat: master control process
             └─875 heartbeat: FIFO reader
               └─876 heartbeat: write: bcast enp0s3
                 └─877 heartbeat: read: bcast enp0s3

```

On peut vérifier le réseau de web1, on pourra remarquer l'IP flottante.

```

root@web1:~# 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:fa:3a:bb 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:fefa:3abb/64 scope link
       valid_lft forever preferred_lft forever

```

L'IP flottante est importante car elle est sur le serveur web1 et sur le serveur web2, c'est avec elle qu'on fait la liaison entre les deux serveurs.

Avant de tester le service Heartbeat, on va modifier les pages Apache sur les deux serveurs web afin que l'on puisse reconnaître sur quel serveur on affiche la page. Il suffit de modifier l'index.html dans /etc/var/www/html/index.html et dans le grand titre de du site on va mettre Web 1 ou Web 2 pour les différencier.

Pour finir, on va entrer l'IP flottante sur Internet, on pourra remarquer que la page affichée est celle du serveur web1. Si on éteint ce serveur et qu'on actualise la page on pourra voir que c'est la page du serveur web2 qui apparaîtra.

### **Conclusion du TP :**

#### ***Avantage :***

- Simplicité de mise en œuvre

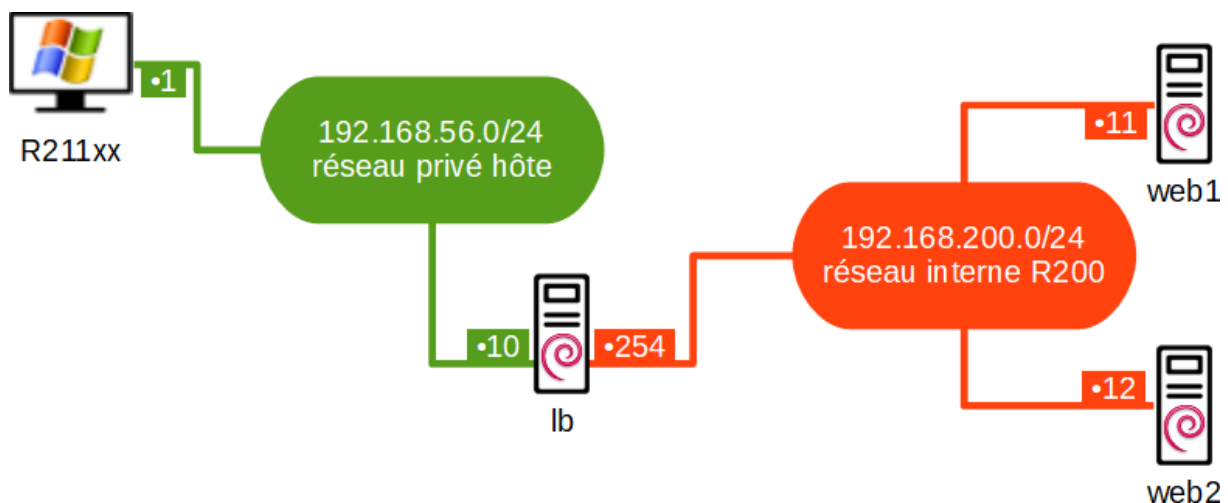
#### ***Inconvénients :***

- Un serveur est monopolisé à ne rien à faire
- En cas d'attaque par déni de service, la même adresse est pointée pour les 2 serveurs
- Le site est indisponible pendant quelques secondes

## Objectif : Mettre en place un Load-balancer pour faire de la répartition des charges

- Création des serveurs web et du serveur lb
- Configuration des interfaces réseaux de ces 3 serveurs
- Mise en place du load-balancer
- Modification des fichiers de configuration du service ipvsadm
- Modifier les pages Apache sur les deux serveurs web
- Tester sur Internet le fonctionnement du load-balancer

### Schéma du réseau virtuel à réaliser



Dans ce réseau, on peut remarquer qu'il y a deux interfaces réseau : Réseau Privé Hôte et Réseau Interne. Le load-balancer est celui qui fait le lien entre les deux il va communiquer avec la machine hôte et avec le réseau interne comprenant les serveurs web.

- Dans ce mode, les serveurs ignorent qu'ils sont en cluster.
- Aucune configuration particulière sur les serveurs, toute la configuration se fait sur le load balancer.
- La seule contrainte est que la passerelle par défaut des serveurs du cluster doit-être le « load-balancer » !

### Installation des serveurs web et du serveur load-balancer

Tout d'abord on va procéder de la même façon que pour Heartbeat, les différences ici est qu'on va ajouter un load-balancer avec 2 interfaces réseau.. Pour l'interface du Réseau Internet on va la renommée R200. L'autre différence est que l'on va mettre les serveurs web en Réseau interne et on choisit le nom que l'on a donné pour le serveur lb, c'est-à-dire R200.



## Configuration des interfaces réseaux des trois serveurs

### Configuration pour les serveurs web

Avant de mettre en Réseau Interne et de modifier l'interface réseau, on va installer apache2.

```
# The primary network interface
allow-hotplug enp0s3
iface enp0s3 inet static
address 192.168.200.11/24
gateway 192.168.200.254
```

Web 1

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

Web 2

### Configuration pour le serveur lb

Comme pour les serveurs web, on va installer le service "ipvsadm" pour le load-balancing. Ce service met en œuvre le répartiteur de charges LVS.

```
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
```

## Modification des fichiers de configuration du service ipvsadm

### Activation du routeur

Pour commencer, on va modifier la configuration du fichier sysctl.conf dans /etc/sysctl.conf afin d'activer le routeur. Il suffit juste d'enlever le commentaire sur la ligne suivant :

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

La valeur 1 confirme qu'il est bien activé.

On va reboot le serveur et vérifier que le routeur est bien activé.

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

Modification des fichiers de configuration

## /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"
```

- Chargement de l'application et des règles au démarrage
- « Maître » par défaut puisqu'il est le seul load balancer
- C'est par cette interface qu'arrivent les requêtes vers le cluster de serveurs Web

## /etc/ipvsadm.rules

On va maintenant configurer ce fichier en commençant par définir le service, puis on va mettre les membres du clusters.

```
# 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
```

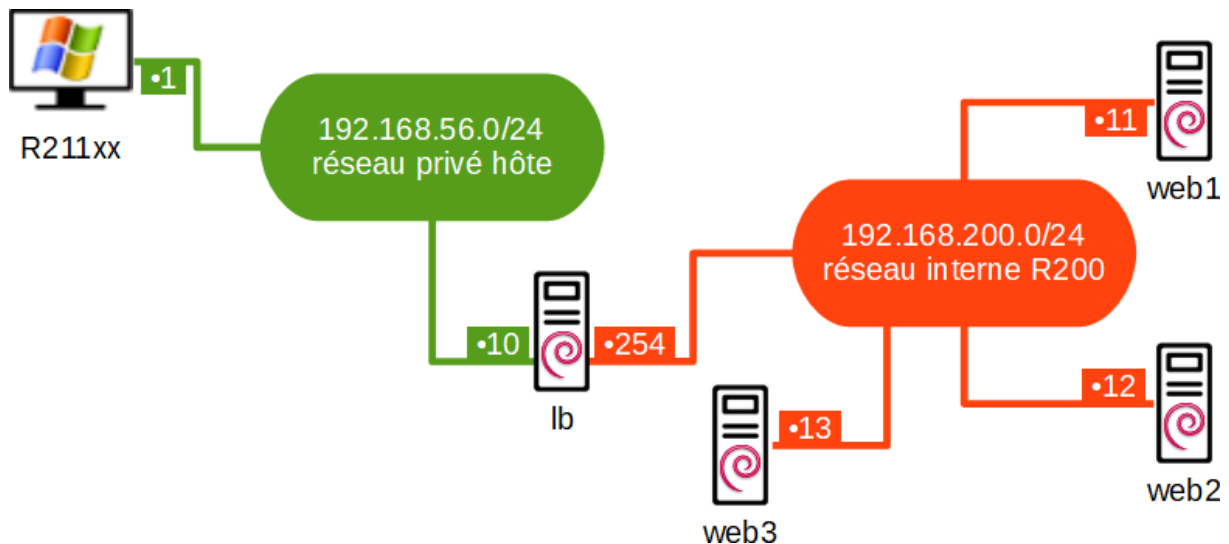
Avec la commande `ipvsadm -ln` on va pouvoir vérifier la configuration du fichier.

```
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:~# _
```

Objectifs: Ajouter un troisième serveur web dans le réseau interne R200

- Création et configuration d'un autre serveur web
- Modification de la page Apache du serveur web3
- Intégration du nouveau serveur dans les fichiers de configurations ipvsadm

Schéma du réseau virtuel à effectuer



Cette étape est similaire à l'étape précédente. Pour gagner du temps, on peut cloner le serveur web2 et modifier l'adresse IP ainsi que la page Apache afin de savoir que c'est la page qui provient du serveur web3.

### Modifications pour ajouter le nouveau serveur web

On va modifier l'interface réseau pour lui attribuer une nouvelle adresse IP.

```
# The primary network interface
allow-hotplug enp0s3
iface enp0s3 inet static
address 192.168.200.13/24
gateway 192.168.200.254
```

Ne pas oublier également de changer le nom de la machine dans /etc/hostname.

Ensuite on modifie la page d'Apache pour afficher le nom du serveur sur la page afin de reconnaître que c'est lui qui est affiché sur Internet.

```
</head>
<body>
  <div class="main_page">
    <div class="page_header floating_element">
      
      <span class="floating_element">
        Apache2 Debian Default Page Web 3
      </span>
    </div>
```

Pour finir, on va ajouter la machine sur le fichier de configuration ipvsadm.rules pour tous les serveurs web.

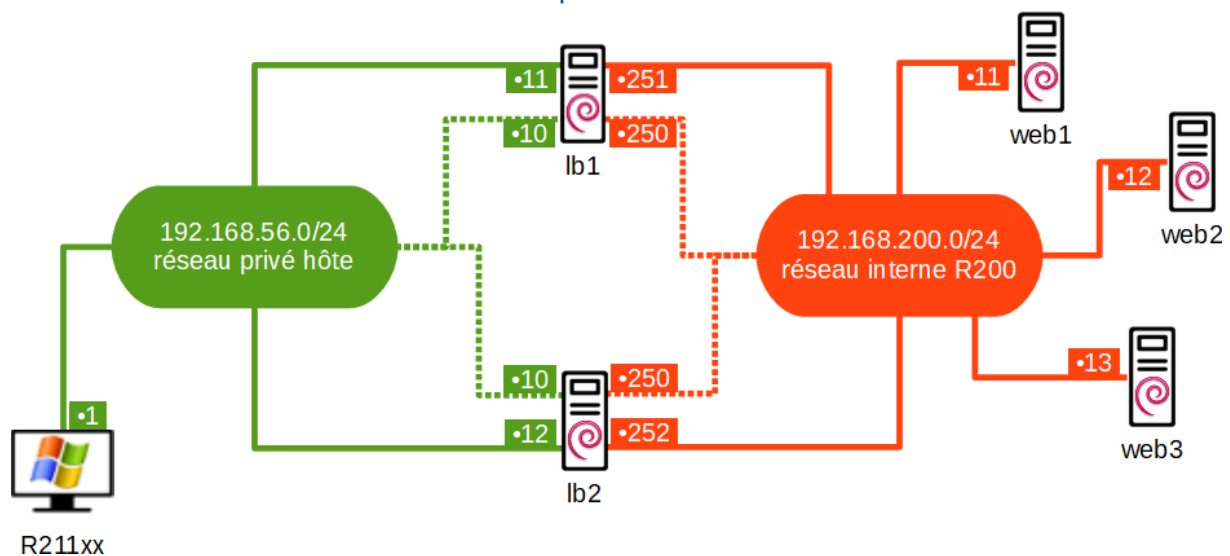
```
# empty rules file for ipvsadm
ipvsadm -A -t 192.168.56.10:80 -s rr

# membres du cluster
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
```

Objectifs : Mise en place d'un second serveur lb et appliquer la méthode de Heartbeat entre ces deux serveurs

- Création d'un second serveur lb et configuration de celui-ci
- Modification des interfaces réseau
- Mise en place du service Heartbeat
- Modification des fichiers de configuration de Heartbeat
- Ajouter les hôtes dans le fichier /etc/hosts
- Tester le fonctionnement de l'infrastructure selon divers contextes

Schéma de l'infrastructure à mettre en place



Dans cette partie, nous allons ajouter un nouveau serveur lb et mettre en place Heartbeat entre les deux serveurs afin qu'en cas de panne le deuxième puisse prendre le relais. On peut voir qu'il y a 2 IP flottantes : une pour le Réseau Privé Hôte et une pour le Réseau Interne.

Mise en place d'un second serveur load-balancer

Pour commencer, on va reproduire à l'identique ce que l'on a fait sur le serveur lb1 car ils auront les mêmes fonctions. Une fois que le service ipvsadm est installé et configuré on va pouvoir installer Heartbeat sur les 2 serveurs lb et effectuer les mêmes configurations dans les fichiers de Heartbeat.

## Installation et configuration de Heartbeat

On commence par créer et modifier le fichier `/etc/ha.d/ha.cf` ainsi que `/etc/ha.d/authkeys`. Sur la première partie du TP nous avons déjà vu et expliqué la modification des fichiers de configuration.

```
bcast enp0s3
deadtime 5
keepalive 1
node lb lb2

auth 1
1 md5 mot de passe
```

Warning : A la fin du TP pour les tests je n'arrivais pas à avoir les adresses IP flottantes sur les serveurs lb. La solution était que j'avais oublié de faire un `chmod 600` sur le fichier `authkeys`.

Les premiers changements viennent au niveau du fichier `/etc/ha.d/haresources`. Nous avons deux IP flottantes et c'est dans ce fichier où nous les stipulons. Le serveur lb est primaire c'est pour cela qu'on le garde au début des deux lignes. Ensuite on donne l'IP flottante avec le service et/ou l'interface.

```
lb IPaddr::192.168.56.10 apache2
lb IPaddr::192.168.200.250 enp0s8
```

On va désormais modifier le fichier `ipvsadm.rules` pour les deux serveurs de façon à ce que ce soit plus du Round Robin mais du Weighted Round Robin (`wrr`), cela va nous permettre de définir les charges qui seront affectées par les différents serveurs web.

```
# empty rules file for ipvsadm
ipvsadm -A -t 192.168.56.10:80 -s wrr

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

Options `ipvsadm` : <https://linuxstar.info/ipvsadm/>

Enfin nous allons référencer les serveurs dans le fichier `/etc/hosts` mais attention car les deux serveurs lb n'auront pas la même.

### Référencement des différents serveurs dans `/etc/hosts` pour les serveurs lb

Pour le serveur lb :

```

127.0.0.1      localhost
127.0.1.1      lb
192.168.56.12  lb2
192.168.200.11 web1
192.168.200.12 web2
192.168.200.13 web3

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

```

Pour le serveur lb2 :

```

127.0.0.1      localhost
127.0.1.1      lb2
192.168.56.11_ lb
192.168.200.11 web1
192.168.200.12 web2
192.168.200.13 web3

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

```

Par la même occasion, nous allons modifier les interfaces réseau des deux serveurs ainsi que les 3 serveurs web.

Configuration des interfaces réseaux sur chacun des serveurs

Pour le serveur lb1 :

```

# The primary network interface
allow-hotplug enp0s3
iface enp0s3 inet static
address 192.168.56.11/24
gateway 192.168.56.254

# The secondary network interface
allow-hotplug enp0s8
iface enp0s8 inet static
address 192.168.200.251/24

```

Pour le serveur lb2 :

```

# 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

```

Pour les serveurs web (attribuer leur bonnes adresses IP) :


```
# The primary network interface
allow-hotplug enp0s3
iface enp0s3 inet static
address 192.168.200.11/24
gateway 192.168.200.250_
```

Enfin, nous allons redémarrer le service Heartbeat et en affichant les interfaces réseau avec la commande `ip addr` on peut voir les adresses IP flottantes, cela veut dire que tout est bon.

```
2: enp0s3: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state UP group default
000
    link/ether 08:00:27:a6:ae:51 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:fea6:ae51/64 scope link
        valid_lft forever preferred_lft forever
3: enp0s8: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state UP group default
000
    link/ether 08:00:27:dd:b6:ff 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:fedd:b6ff/64 scope link
        valid_lft forever preferred_lft forever
```

Pour les tests on a juste à taper l'adresse IP flottante sur Internet afin de voir si une page Apache apparaît. Si on rafraîchit la page on aura les pages des 3 serveurs web.

Si un serveur web est down, les deux continueront de fonctionner et même chose pour le serveur lb car il est primaire mais grâce à Heartbeat le second serveur lb2 prendra le relais en cas de panne sur le premier (on peut shutdown le serveur lb pour essayer).



## Apache2 Debian Default Page Web 1

**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





## Apache2 Debian Default Page Web 3

**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



## Apache2 Debian Default Page Web 2

**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.