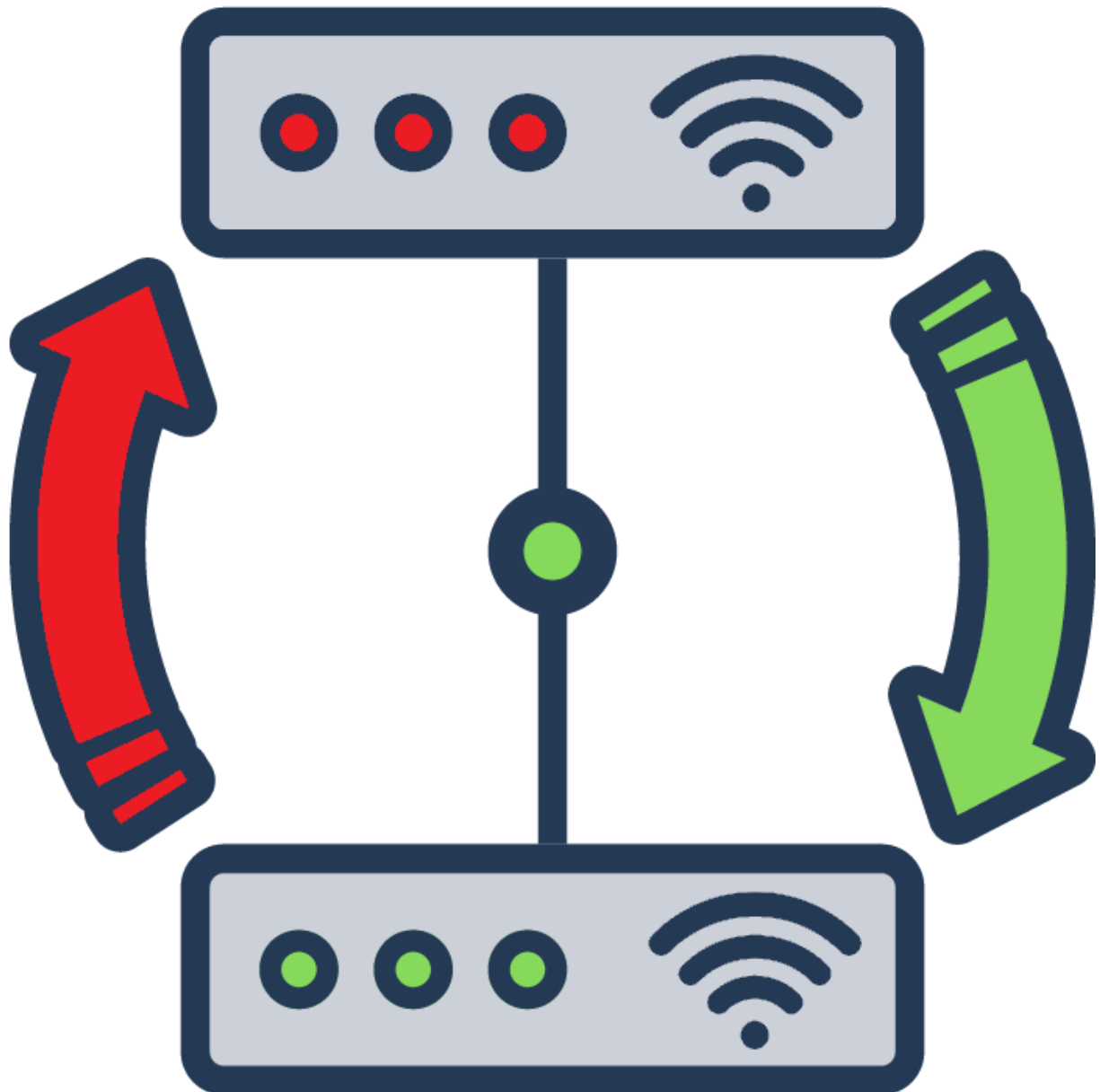


---

## LA HAUTE DISPONIBILITE

---



## Table des matières

Principe général.....	4
Eléments vitaux à surveiller .....	4
Fiabilité et disponibilité.....	4
La méthode des 9 .....	5
Exemple .....	5
Les outils.....	6
Mode dégradé .....	7
PCA ? PRA ? .....	7
Local adapté .....	8
La sauvegarde des données .....	9
L'archivage.....	9
L'organisation des sauvegardes .....	9
Différents types des sauvegardes .....	10
Redondance de serveurs.....	12
Round Robin DNS .....	12
Load balancer ou répartiteur de charges.....	12
Heartbeat .....	13
Principe et vocabulaire technique.....	13
Objectif : installation des machines Web1 et Web 2 et du service Heartbeat .....	15
Schéma .....	15
Fichier Ha.cf.....	16
Fichier authkeys.....	17
Fichier haresources .....	17
Déclaration des hôtes.....	17
Test final .....	18
Load Balancing (Répartiteur de charge).....	20
Fonctionnement du Load Balancing.....	20
Schéma du réseau à mettre en place .....	20
Machines web: Configuration IP de Web1 et Web2 .....	21
Serveur « lb ».....	21
Activation du routage pour la machine lb.....	21
Configuration du fichier ipvsadm .....	22
Configuration du fichier ipvsadm.rules .....	22
Objectif : ajout d'un répartiteur de charge et d'une machine Web3 .....	24
Schéma du réseau à mettre en place :.....	24

Configuration du fichier ipvsadm.rules .....	25
Création de lb2 et installation de Heartbeat.....	25
Schéma du réseau à mettre en place :.....	25
Configuration IP de lb1 et lb2 :.....	26
Configuration pour la machine lb1.....	26
Configuration pour la machine lb2.....	26
Configuration du répartiteur de charge sur les deux machine Load Balancer.....	27
Test final .....	28

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

## Principe général

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

## Éléments vitaux à surveiller

### La disponibilité des données

- Par exemple : 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 logiciel.

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

- Permet de connaître l'intervalle de temps où un service est indisponible 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 soit 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 méthode pour évaluer le niveau de disponibilité d'un service.
- Consiste à ne pas tenir compte de la fréquence des pannes mais uniquement de leur durée
- Il s'agit d'évaluer d'arrêt cumulée du service sur un an

**Exemple :** Fixons-nous comme arrêt cumulé sur un an 5 minutes → Le service doit être disponible 99,999 % du temps.

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

Principe : un simple produit en croix

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...

100% de disponibilité → sur un an 365 jours de 24h soit 8 760h ou encore 525 600 minutes (8 760\*60)

### Exemple de l'arrêt cumulé de 5 minutes

- 99,999% de 525 600 minutes soit 525 594,74 minutes de fonctionnement cumulé.
- $525\,600 - 525\,594,74 = 5256$  soit environ 5 minutes

## 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 même 3 alimentations pour se protéger en cas de défaillance de l'alimentation principale.
- Hot-swapping : changement à chaud d'un disque de secours au cas où celui-ci est en panne
- Spare : disque de secours (comme une roue de secours)
- RAID : technique de tolérance de panne qui permet de stocker différentes informations sur un disque, elle permet une répartition des données sur tous les disques dur, l'avantage est que le système peut supporter la perte d'un ou deux disques dur (cela dépend de la version du RAID).
- Cartes réseaux additionnelles : on est capable de créer une interface réseau virtuelle qui va regrouper plusieurs interfaces physiques grâce au Channel Bonding (Agrégation de liens)
- Spanning tree : protocole réseau (STP) qui permet de détecter les boucles dans un réseau et de les désactiver afin d'éviter une tempête de broadcast (surcharge du réseau), il désactive donc des ports. Il va réactiver les ports si un câble est défectueux afin que tout puisse communiquer.
- Redondance de passerelles : c'est le principe sur laquelle l'entreprise va s'assurer qu'elle a toujours une connexion internet valable (2 routeurs de 2 FAI différents)
- 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 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 le réseau).
- Accès Distant : Tunnel SSH
- Remontée des événements : à vous de mettre en place les bons outils (à vos script s'il n'existent pas déjà) afin de surveiller vos système AID, Chanel Bonding et autres !
- Sauvegardes complètes, différentielles, incrémentielles
- Le stockage des sauvegardes à un emplacement géographique différent
- Mode dégradé
- Plan de secours, plan de continuité d'activité (PCA), plan de reprise d'activité (PRA)

## Mode dégradé

- Initialement un langage militaire
- Désigne les situations où tout ou 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 (bioterrorisme), catastrophe majeure (technologie 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, télécommunication, etc.
- Pour réagir au mieux et retrouver au plus vite une situation normale ou « restaurée », les acteurs vitaux sont 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 (PCA) organise la poursuite des activités de l'entreprise en cas d'incident.
- Le plan de Reprise d'Activité (PRA) anticipe une interruption de l'activité et prévoit les conditions de sa reprise.

### Pourquoi mettre en place un Plan de Reprise d'Activité ?

- La reprise d'une activité, même partielle, garantit un niveau de CA 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 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.



### Au niveau du local technique :

- Norme TIA 569 : chaque étage doit contenir un local technique tous les 1000 m2
- Alarme / vidéo surveillance
- Norme NFE 35 – 400

### Au niveau de l'infrastructure

- Cluster
- Redondance
- Snapshot
- Réplication
- Fréquence de sauvegarde automatique (BackupPC)
- Redondance
- Onduleur & alimentation redondante
- Sauvegarde externalisé
- Mise en miroir des serveurs

## 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'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.

## 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 locaux ;
- La **rotation** des supports éventuellement retenue ;
- Le(s) salarié(s) **responsable(s)** de cette mission.

### Différents types de 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.

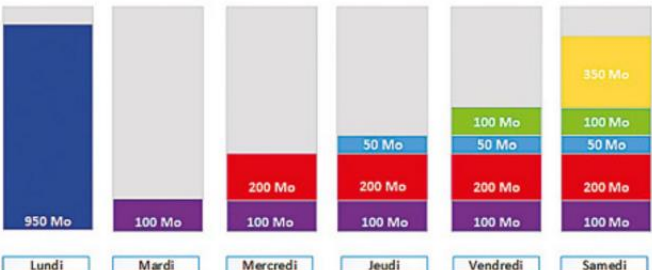
## Sauvegarde incrémentale

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	Volume (Mo)
Lundi	Complète	950
Mardi	Incrémentale	100
Mercredi	Incrémentale	200
Jeudi	Incrémentale	50
Vendredi	Incrémentale	100
Samedi	Incrémentale	350

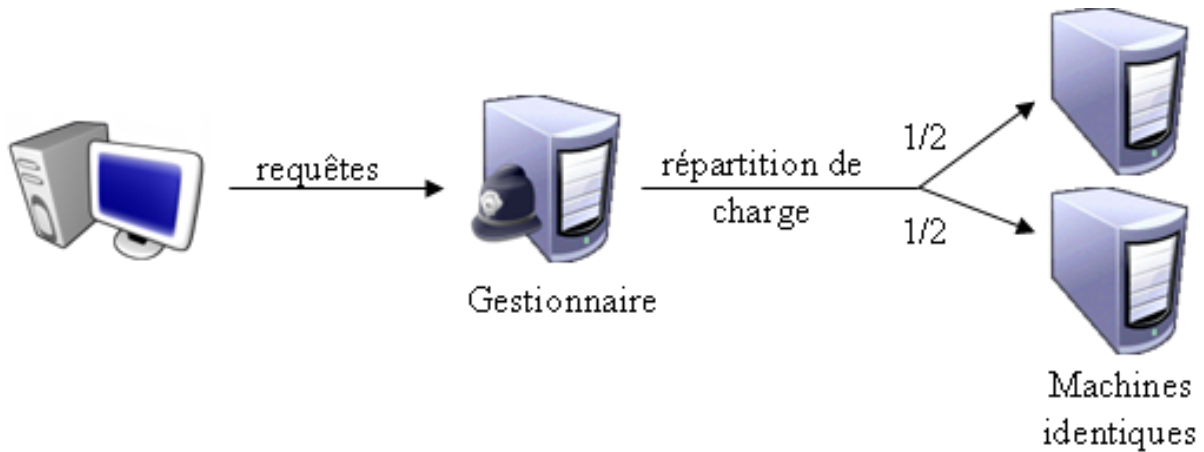
<b>Sauvegarde différentielle</b>	<p>Seules les données modifiées depuis la dernière <b>sauvegarde complète</b> sont sauvegardées. Le volume à sauvegarder augmente donc progressivement, ce qui nécessite jour après jour de plus en plus de temps et d'espace de stockage. La restauration nécessitera de restaurer dans un premier temps la dernière sauvegarde complète, puis seulement la dernière sauvegarde différentielle, ce qui est moins fastidieux qu'avec une sauvegarde incrémentale.</p> <p><b>Exemple</b> En cas d'incident le vendredi, il faudra restaurer la sauvegarde complète du lundi, puis la sauvegarde différentielle du jeudi.</p> 
<b>Sauvegarde mixte</b>	<p>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.</p>

	Sauvegarde	Archivage électronique
<b>Objectif</b>	Restauration des données	Conservation de l'information (document de référence ou archive probante)
<b>Gestion des risques</b>	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.
<b>Accès</b>	<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>
<b>Sécurité d'accès</b>	<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>
<b>Conservation</b>	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>

## Redondance de serveurs

### Round Robin DNS

- Plusieurs serveurs proposant le même service
- Pouvoir rediriger 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, etc.



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

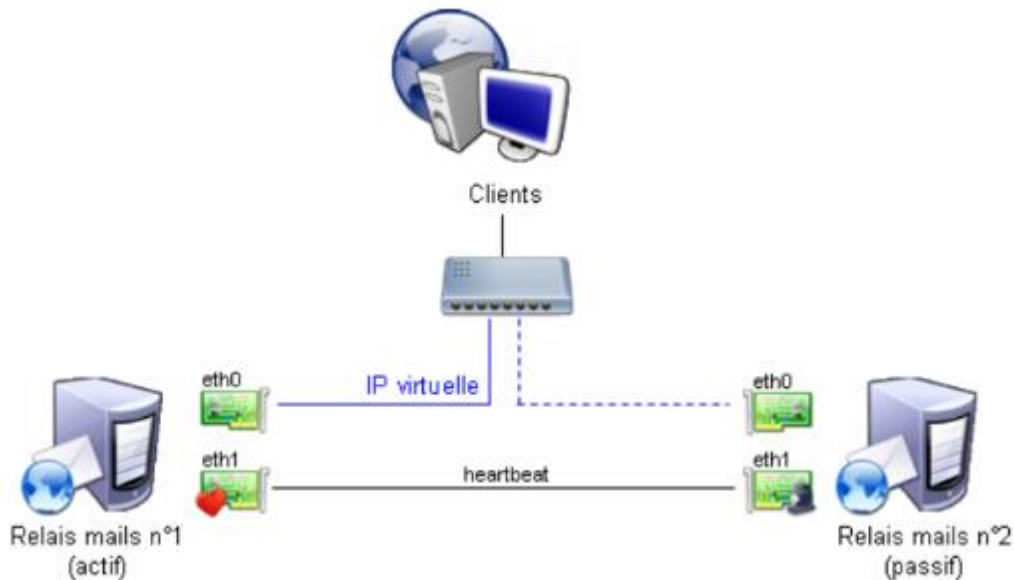


- ...et prendre le relais automatiquement en cas d'**arrêt** des **battements**. Il devient alors **actif**.

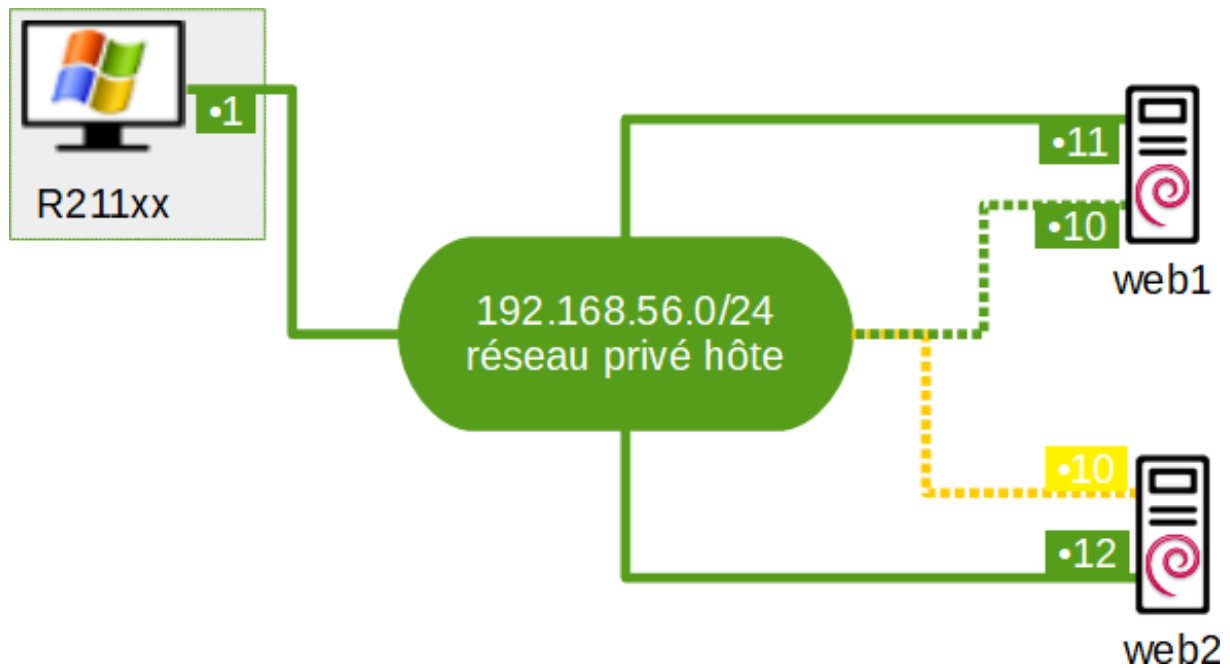


- Chaque serveur possède sa propre adresse IP, mais seul **le « nœud »** actif possède, en plus une adresse virtuelle : c'est par cette adresse « flottante » que les clients accèdent au service

- 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 faite 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.



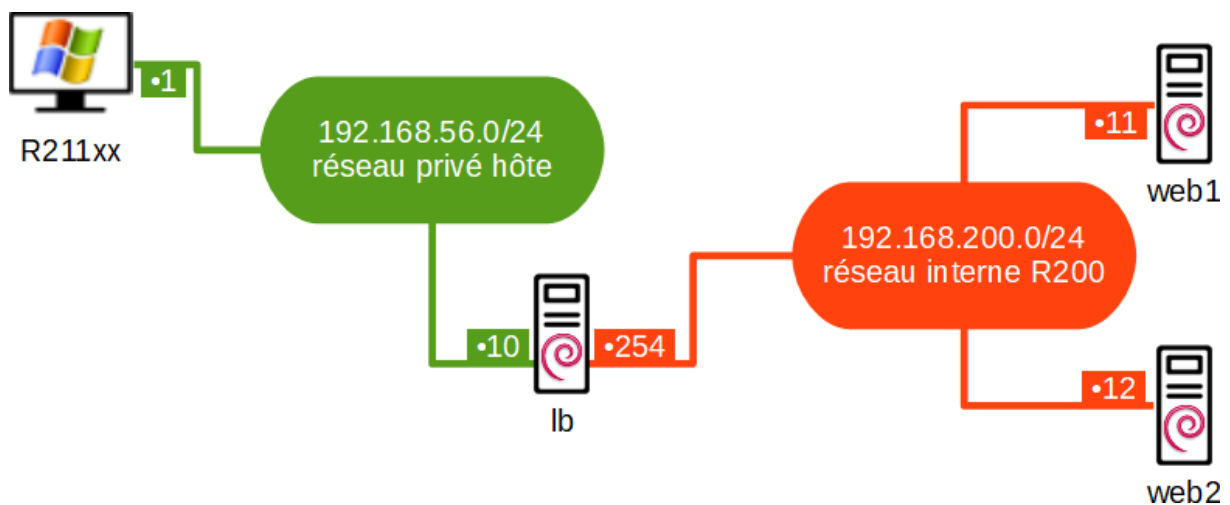
### Mise en œuvre

La configuration de **heartbeat** doit être identique sur tous les serveurs de la grappe.

- Les 3 fichiers demandés doivent se trouver dans les répertoires suivant :

Objectif : installation des machines Web1 et Web 2 et du service Heartbeat

### Schéma



- Il faut tout d'abord renommer les différentes machines pour que Heartbeat puisse se baser dessus
- On peut maintenant installer apache2.service sur les 2 machine Web avec la commande « apt install apache2 »
- Ensuite on peut installer le packet « Heartbeat » avec la commande « apt install Heartbeat »

**Attention : Après l'installation de heartbeat, ces 3 fichiers ne sont pas présents sur la machine, à noter que la configuration de ces fichier est identique sur tous les serveurs.**

Avantage de heartbeat : Simplicité de mise en œuvre

Inconvénient de heartbeat : - Un serveur est monopolisé à ne rien 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.

[Fichier Ha.cf](#)

 **/etc/ha.d/ha.cf**

- La configuration sur chacune des machines est la même « /etc/ha.d/ha.cf »

```
bcast emp0s3
deadtime 5
keepalive 1
node web1 web2
```

Interface réseau

Nœud web1 et web2

**Keepalive** : Intervalle entre 2 battement de cœur. La valeur est en secondes par défaut. Pour la spécifier en millisecondes, on rajoutera 'ms' derrière. (Par exemple, 200ms).

**Deadtime** : Temps nécessaire avant de considérer qu'un nœud est mort. Le temps est en secondes par défaut. On rajoutera 'ms' derrière pour l'avoir en millisecondes.

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.

Fichier authkeys

## /etc/ha.d/authkeys

- 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 choix : sha1 ou SHA256*)

```
auth 1
1 md5 motdepasse
```

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

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

Fichier haresources

## /etc/ha.d/haresources

- Lise des ressources (adresses virtuelles et services concernées) fournies par la grappe. La configuration sur chacune des machines est la même. Ce nom doit être le même pour 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 web reste le serveur  
« maître »

Déclaration des hôtes

- **Web1** et **web2** doivent être déclarés dans **/etc/hosts** ! (excepté si un service DNS est installé)

```
12127.0.0.1 localhost
12127.0.1.1 web2
19192.168.56.11 web1
```

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

- Commençons par stopper le service apache2 sur les 2 machines

```
systemctl stop apache2.service
```

- On vérifie que le service est bien inactif

```
inactive
```

Afin de reconnaître les pages debian des machines web1 et web2 on peut modifier la page par défaut d'apache dans le dossier /var/www/html/index.html :

```
Apache2 Debian Default Page : 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>
-->
root@buster:/var/www/html#
```

- Désactiver le service apache2 sur les 2 machines → Heartbeat le démarrera lui-même.

```
systemctl disable apache2.service
```

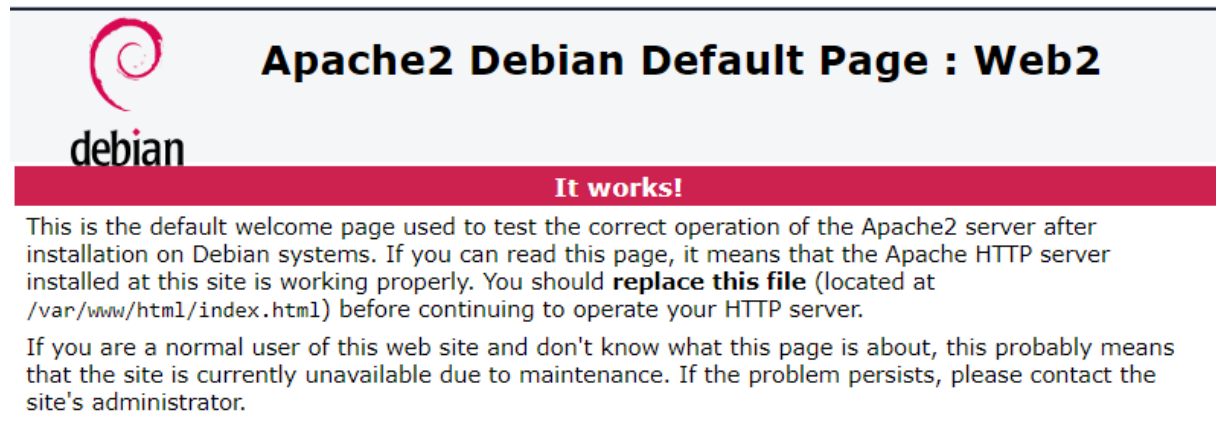
- Tester que le service Heartbeat est bien actif. Le redémarrer avec la commande restart.

```
systemctl restart heartbeat.service
```

## Test final

- Connectez-vous au site « par défaut » avec la machine « hôte » à l'adresse
- Eteignez web1
- Tentez de joindre à nouveau la grappe à partir du navigateur du poste « hôte ».
- Le serveur web2 a pris le relais

⚠ Non sécurisé | 192.168.56.10

**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

**Load Balancing (Répartiteur de charge)**

- 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.

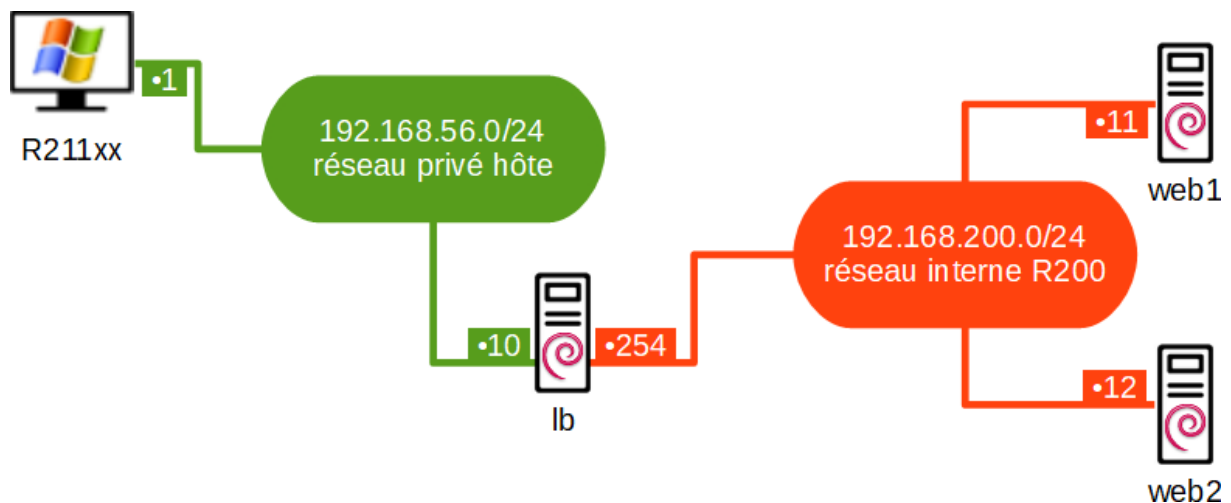
## Load Balancing (Répartiteur de charge)

- Les clients envoient leurs requêtes au « **load-balancer** » qui se charge de les transmettre à la grappe 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 d'eux, 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

### Fonctionnement du Load Balancing

- L'algorithme de répartition de charge utilisé dans l'exemple est le « Round-Robin » (tourniquet) qui attribue chaque nouvelle requête au serveur suivant disponible.

### Schéma du réseau à mettre en place



- 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 » !

### Machines web: Configuration IP de Web1 et Web2

```
# 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

### Serveur « lb »

- Mise en œuvre de toutes les fonctionnalités nécessaires (en profitant de l'accès à Internet) ou activation du routeur.
- Apt update
- Installer le paquet « ipvsadm » qui mettra en œuvre le répartiteur de charge « LVS ».
- Apt install ipvsadm
- Ajouter deux interfaces : Réseau privé hôte et Réseau interne
- Faire la configuration réseau :

```
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
```

### Activation du routage pour la machine lb

- Faire la commande « nano /etc/sysctl.conf »
- Enlever la commande sur la ligne :

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

On peut maintenant vérifier si le routage est bien activée.

- Pour vérifier il faut taper la commande :

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

Configuration du fichier ipvsadm

## /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
- C'est par cette interface qu'arrivent les requêtes vers le cluster de serveurs Web

Configuration du fichier ipvsadm.rules

## /etc/ipvsadm.rules

- Configurer le fichier comme afficher ci-dessous :

```
# 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
```

- -A ajoute un service et d'autres éléments juste derrière comme le protocole, l'adresse IP et le port et l'algorithme de répartition Round Robin (« rr »)

Le « -m » signifie masquerade pour que les réponses viennent du répartiteur de charge lb

```
# 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
```

-a ajoute un nœud

Adresse IP du service

Adresse IP et port du nœud du cluster

⇒ On peut tester si le répartiteur de charge fonctionne en faisant « F5 » sur la page afin de voir les adresses IP qui change (on peut modifier la page index dans </var/www/html/index.html> pour distinguer les pages)

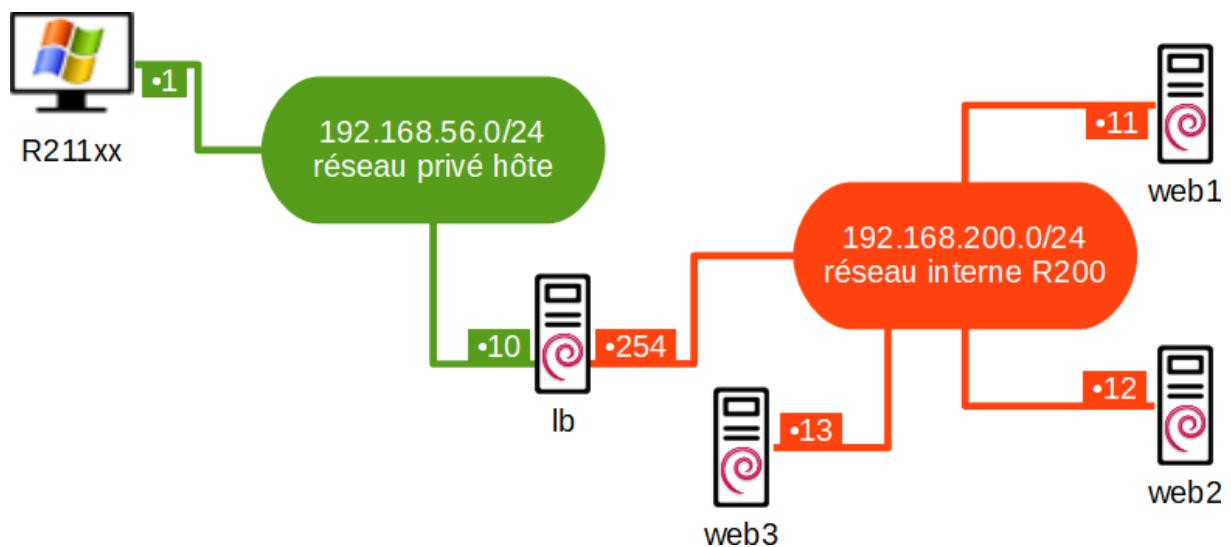
Utiliser la commande `ipvsadm -ln` pour vérifier la configuration

→ Il faut avoir un « 1 » sur la ligne « weight » sinon cela signifie que la configuration est pas bonne.

```
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:~# _
```

## Objectif : ajout d'un répartiteur de charge et d'une machine Web3

Schéma du réseau à mettre en place :



- On va tout d'abord créer une machine Web3 avec et ensuite on pourra l'ajouter au load balancer pour répartir les charges des requêtes entre les 3 machine de notre réseau

### Réseau

Adapter 1

Adapter 2

Adapter 3

Adapter 4

☒ Activer l'interface réseau

Mode d'accès réseau : Réseau interne

Nom : R200

Avancé

- Ensuite on peut modifier la configuration réseau comme ci-dessous :

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



- On installe également apache2 et on modifie 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>
```

### Configuration du fichier ipvsadm.rules

- Modifier ensuite dans lb le fichier ipvsadm.rules comme ci-dessous :

```
# 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
```

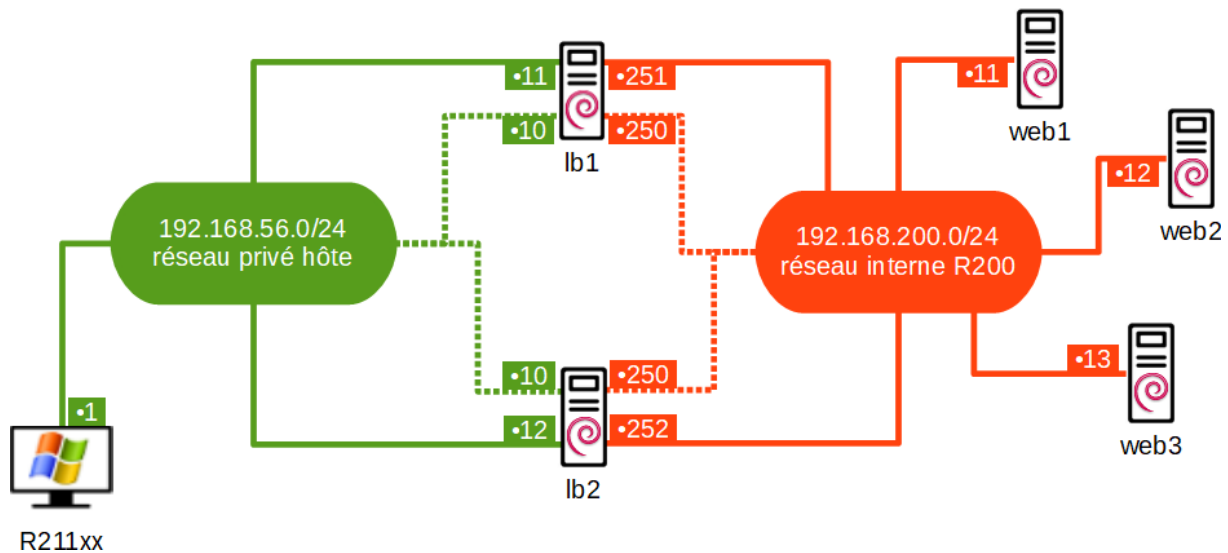
Ici, on ajoute tout simplement Web3 au répartiteur de charge

- On n'oublie pas également de renommer la machine dans le fichier /etc/hostname

```
GNU nano 3.2 /etc/hostname
web3_
```

### Création de lb2 et installation de Heartbeat

Schéma du réseau à mettre en place :



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

**Réseau**

Adapter 1   Adapter 2   Adapter 3   Adapter 4

☒ Activer l'interface réseau

Mode d'accès réseau : Réseau interne

Nom : R200

▶ Avancé

Configuration IP de lb1 et lb2 :

- Il faut à présent configurer les adresses IP des deux serveur load balancer :

Configuration pour la machine lb1

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

#The 2nd network interface enp0s8
allow-hotplug enp0s8
iface enp0s8 inet static
address 192.168.200.251/24_
```

Configuration pour la machine lb2

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

#The 2nd network interface enp0s8
allow-hotplug enp0s8
iface enp0s8 inet static
address 192.168.200.252/24
```

- Après avoir terminé la configuration on peut installer Heartbeat avec la commande « apt install Heartbeat », si cela ne fonctionne pas il faut revoir la configuration IP des serveur load balancer.

```
apt install heartbeat_
```

- On installe également ipvsadm avec la commande « apt install ipvsadm »

```
apt install ipvsadm
```

### Configuration du répartiteur de charge sur les deux machine Load Balancer

- 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 /etc/sysctl.conf

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

- On peut maintenant configurer les fichiers ipvsadm.rules et ipvsadm :

### /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"
```

### /etc/ipvsadm.rules

```
#service
ipvsadm -A -t 192.168.56.10:80 -s wrr
#membre du cluster
ipvsadm -a -t 192.168.56.10:80 -r 192.168.200.11:80 -m -w 3
ipvsadm -a -t 192.168.56.10:80 -r 192.168.200.12:80 -m -w 1
ipvsadm -a -t 192.168.56.10:80 -r 192.168.200.13:80 -m -w 1
```

Le serveur Web1 reçoit ici 3 requêtes alors que Web1 et Web2 en reçoivent qu'une seule.

« Wrr » signifie « weight round robin », il indique la charge qui sera mise sur les 3 serveurs.

- Si maintenant on exécute la commande `ipvsadm -ln` on peut voir que les trois serveurs ont bien reçu des requêtes comme indiqué et configuré précédemment.

## Test final

- On vérifie si les adresses si la machine lb1 possède les adresses IP flottantes avec la commande « `ip addr` » ou « `ip a` ». Si on se rend sur un moteur de recherche et que l'on tape l'adresse flottante 192.168.56.10 et que l'on rafraichit la page avec la touche F5 chaque page Web1, Web2 et Web3 devraient être chargée.

