

LA HAUTE DISPONIBILITE

Table des matières

LA HAUTE DISPONIBILITE.....	1
Introduction et définition.....	3
Principe général.....	3
- Qu'est-ce donc ?.....	3
Eléments vitaux à surveiller.....	3
Fiabilité et disponibilité.....	3
La méthode des 9.....	4
Les outils.....	5
Mode dégradé.....	6
PCA ? PRA ?	6
- Comment faire un PRA ?	6
Local adapté.....	7
- Travail à faire sur Power Point	7
La sauvegarde des données	7
La sauvegarde Vs l'archivage.....	7
La sauvegarde des données	7
Redondance de serveurs	9
- Schéma du fonctionnement de la redondance.....	9
- Fonctionnement de Heartbeat « Surveillance de la disponibilité des programmes. ».....	9
- Définition importante.....	10
Mise en place de heartbeat sur deux serveurs pour assurer la redondance en cas de pannes d'un des deux	11
- Objectifs.....	11
- Prérequis	11
Création des trois répertoires et des données devant s'y trouver.	12
- Premier fichier à configurer « /etc/ha.d/ha.cf ».....	12
- Second fichier à configurer « /etc/ha.d/authkeys »	13
- Troisième fichier à configurer « /etc/ha.d/haresources »	13
Fichier « hosts »	14

Le service Heartbeat est -il bien actif ?	14
Load balancing ou répartiteur de charge.	15
- Tout d'abord qu'est-ce que le load balancing ?	15
Schéma d'un système de load balancing 1.....	15
Schéma d'un système de load balancing 2.....	16
Mise en place d'un load balancing (Schéma de l'infrastructure à réaliser sur oracle)	17
Objectifs.....	17
Prérequis	17
- Premier fichier à configurer pour le Load balancing.	19
- Second fichier à configurer	20
Mise en place d'une machine Web 3 et ajout au schéma du fonctionnement du Load balancing	21
Schéma à réaliser sur Virtual box :	21
Objectifs :	21
Prérequis	21
Réalisation :	22
Mise en place d'un deuxième Load balancer sur le schéma de l'infrastructure	23
Prérequis	23
Réalisation :	23
Fichier de configuration Debian LB 1 « nano /etc/network/interfaces »	24
Fichier de configuration Debian LB 2 « nano /etc/network/interfaces »	24
- Fichier à configurer sur les deux LB de la même manière, « /etc/ha.d/ha.cf »	25
- Second fichier à configurer sur les deux LB de la même manière « /etc/ha.d/authkeys » ..	25
- Troisième fichier à configurer sur les deux LB de la même manière « /etc/ha.d/haresources »	26
- Fichier à configurer sur les deux LB de la même manière « hosts »	27
Vérification du load balancing et activation du routage sur les deux Debian LB.	27
Premier fichier à configurer sur les deux LB de la même manière.	28
- Second fichier à configurer sur les deux LB de la même manière	28
Vérification du paramétrage • « Ipvsadm -ln »	29
Correction devoir BTS BLANC	31
Mission 2 : Question A.2.1	32
Question 5) Comment limiter l'impact de ces impacts	32

Introduction et définition

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

Principe général

- Qu'est-ce donc ?
- High availability (HA) en anglais
- Designer le fait qu'une architecture ou un service a un taux de disponibilité convenable
- Enjeux importants car une indisponibilité entraîne des coûts très élevés

Aujourd'hui la haute disponibilité est un enjeu majeur pour la réputation des sites d'e-commerce, des réseaux sociaux....

Éléments vitaux à surveiller

- **La disponibilité des services** : Par exemple, outils indispensables pour le fonctionnement d'un site web marchand
- **La disponibilité des données** : Intégrité des données → perte de données impensables !
- **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 indisponible c'est-à-dire jusqu'à son rétablissement.
- Soit obtenir un **MTBF fort**, c'est-à-dire que l'intervalle entre deux pannes du système est grand.
- Soit 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
- 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

Exemple : fixons-nous comme arrêt cumulé sur un an 5 minute → 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

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

Disponibilité en %	Indisponibilité par année	Indisponibilité par mois ³	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

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** : Tolérance de panne, qui permet de stocker des informations sur plusieurs disques et qui permet selon la version du RAID utilisé une répartition des données sur tous les disques durs. Lorsque l'on perd certain D.D cela n'a pas un grand impact.
- **SPARE** : disque spare ne faisant pas partie des disques de la zone de grappe.



- **Hot swap** : changer un disque dans la zone de grappage à chaud. (Sur une baie de disque dur)
- **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 STP. Le spanning permet de déterminer une topologie réseau dans les Lan par ponts.**
- **Redondance de passerelles** : permet d'accéder à un réseau, à l'aide d'une passerelle (plus sur de la sortie), la connexion internet doit être assurée, la redondance permet donc d'accéder à internet chez des prestataires différents, il doit y avoir une connexion externe.
- **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, 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 aux Wake-on-lan (réveil par le réseau)
- **Accès Distant** : Tunnel SSH, Telnet
- **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 système RAID, channel bonding et autres.
- **Sauvegardes** : complète différentielles, incrémentielles.
- **Mode dégradé** : fournir le service jugé indispensable.
- **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...) doivent fonctionner sans leurs ressources habituelles, humaine et matériel.
- Ex : attentat, catastrophe naturelle.
- Tenter de fournir le service jugé indispensable, en manquant de ressources complètes ou fiables ou régulières en énergie.
- 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.

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 de d'incident.
- Le plan de reprise d'activité (PRA) anticipe une interruption de l'activité et prévoit les conditions de sa reprise.
- 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'éventuelle obligation légale.
- [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 sur Power Point
- 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 des données


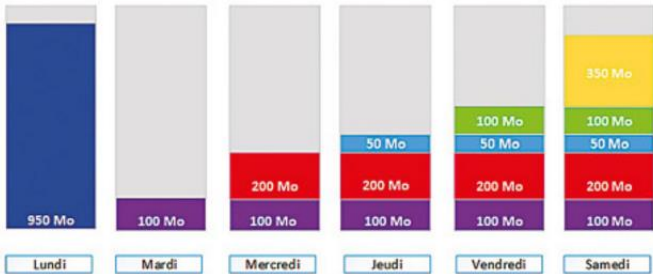
- La législation impose un archivage des données

<https://entreprendre.service-public.fr/vosdroits/F10029>

La sauvegarde Vs l'archivage

- La **sauvegarde** est une opération qui consiste à faire une copie des données, **afin de pallier leurs éventuelles destructions, totale ou partielle** (conséquence d'une catastrophe naturelle, d'un sabotage, de l'attaque d'un virus, d'une défaillance du système informatique).
- Une sauvegarde permet de restaurer les données en cas de panne.
- **L'archivage** 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.

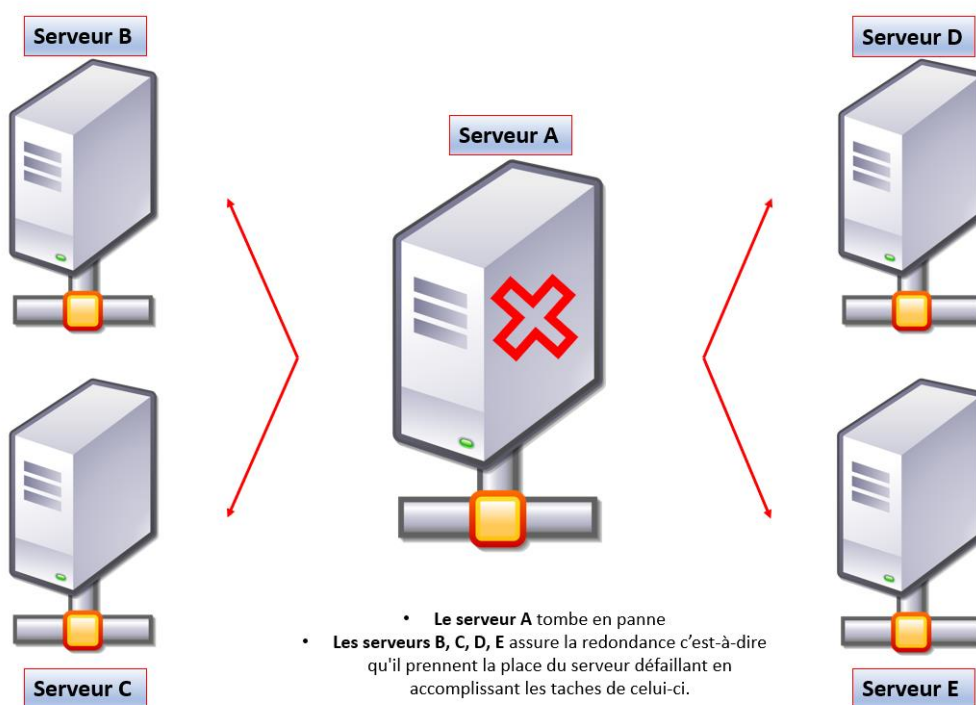
La sauvegarde des données

Sauvegarde complète	<p>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.</p>																					
Sauvegarde incrémentale	<p>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.</p> <p>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.</p>  <table><thead><tr><th>Jour</th><th>Type de sauvegarde</th><th>Volume (Mo)</th></tr></thead><tbody><tr><td>Lundi</td><td>Sauvegarde complète</td><td>950</td></tr><tr><td>Mardi</td><td>Sauvegarde incrémentale</td><td>100</td></tr><tr><td>Mercredi</td><td>Sauvegarde incrémentale</td><td>200</td></tr><tr><td>Jeudi</td><td>Sauvegarde incrémentale</td><td>50</td></tr><tr><td>Vendredi</td><td>Sauvegarde incrémentale</td><td>100</td></tr><tr><td>Samedi</td><td>Sauvegarde incrémentale</td><td>350</td></tr></tbody></table>	Jour	Type de sauvegarde	Volume (Mo)	Lundi	Sauvegarde complète	950	Mardi	Sauvegarde incrémentale	100	Mercredi	Sauvegarde incrémentale	200	Jeudi	Sauvegarde incrémentale	50	Vendredi	Sauvegarde incrémentale	100	Samedi	Sauvegarde incrémentale	350
Jour	Type de sauvegarde	Volume (Mo)																				
Lundi	Sauvegarde complète	950																				
Mardi	Sauvegarde incrémentale	100																				
Mercredi	Sauvegarde incrémentale	200																				
Jeudi	Sauvegarde incrémentale	50																				
Vendredi	Sauvegarde incrémentale	100																				
Samedi	Sauvegarde incrémentale	350																				
Sauvegarde différentielle	<p>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.</p> <p>Exemple En cas d'incident le vendredi, il faudra restaurer la sauvegarde complète du lundi, puis la sauvegarde différentielle du jeudi.</p>  <table><thead><tr><th>Jour</th><th>Type de sauvegarde</th><th>Volume (Mo)</th></tr></thead><tbody><tr><td>Lundi</td><td>Sauvegarde complète</td><td>950</td></tr><tr><td>Mardi</td><td>Sauvegarde différentielle</td><td>100</td></tr><tr><td>Mercredi</td><td>Sauvegarde différentielle</td><td>200 + 100</td></tr><tr><td>Jeudi</td><td>Sauvegarde différentielle</td><td>50 + 200 + 100</td></tr><tr><td>Vendredi</td><td>Sauvegarde différentielle</td><td>100 + 50 + 200 + 100</td></tr><tr><td>Samedi</td><td>Sauvegarde différentielle</td><td>350 + 100 + 50 + 200 + 100</td></tr></tbody></table>	Jour	Type de sauvegarde	Volume (Mo)	Lundi	Sauvegarde complète	950	Mardi	Sauvegarde différentielle	100	Mercredi	Sauvegarde différentielle	200 + 100	Jeudi	Sauvegarde différentielle	50 + 200 + 100	Vendredi	Sauvegarde différentielle	100 + 50 + 200 + 100	Samedi	Sauvegarde différentielle	350 + 100 + 50 + 200 + 100
Jour	Type de sauvegarde	Volume (Mo)																				
Lundi	Sauvegarde complète	950																				
Mardi	Sauvegarde différentielle	100																				
Mercredi	Sauvegarde différentielle	200 + 100																				
Jeudi	Sauvegarde différentielle	50 + 200 + 100																				
Vendredi	Sauvegarde différentielle	100 + 50 + 200 + 100																				
Samedi	Sauvegarde différentielle	350 + 100 + 50 + 200 + 100																				
Sauvegarde mixte	<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>																					

Redondance de serveurs

- **La redondance**, est utiliser lorsqu'un serveur tombe en panne, il permet à un serveur que l'on appelle serveur maitre de prendre la fonction du serveur défaillant pour éviter les disfonctionnements sur le réseau.
 - On peut avoir recours à la redondance à l'aide de **Heartbeat**.
- Schéma du fonctionnement de la redondance.

<https://www.it-connect.fr/wp-content-itc/uploads/2013/11/dhcpred1.jpg>



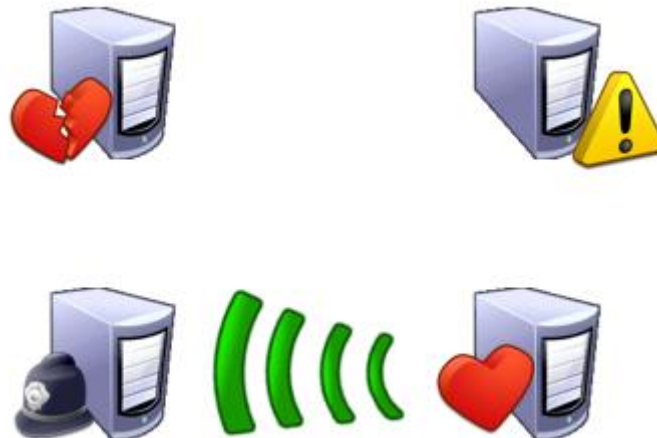
- Fonctionnement de Heartbeat « Surveillance de la disponibilité des programmes. »

Heartbeat est un logiciel de surveillance de la disponibilité des programmes.

- On parle de Heartbeat pour « **battement de cœur** ».
- Heartbeat doit être installer sur tous les serveurs. Lorsqu'un Serveur (le serveur maitre) tombe en panne, un serveur secondaire disposant de heartbeat réussi à comprendre aussitôt que celui-ci est « inactif ».



Le serveur maître dis « actif » va ensuite devenir « passif » et laisser au serveur secondaire sa place, il devient ensuite actif.



Le serveur de droite comprend que le serveur de gauche ne fonctionne plus il prend donc le relais s'il est capable de gérer les ressources du serveur en panne.

- Définition importante

- **Keepalive** : intervalle entre 2 battements de cœur. La valeur est en seconde par défaut. Pour la spécifier en millisecondes, on rajoutera 'ms' derrière. (Par exemple, 200 ms).
- **Deadtime** : Temps nécessaire avant de considérer qu'un nœud est mort. Le temps est en seconde par défaut. On rajoutera 'ms' derrière pour l'avoir en millisecondes.

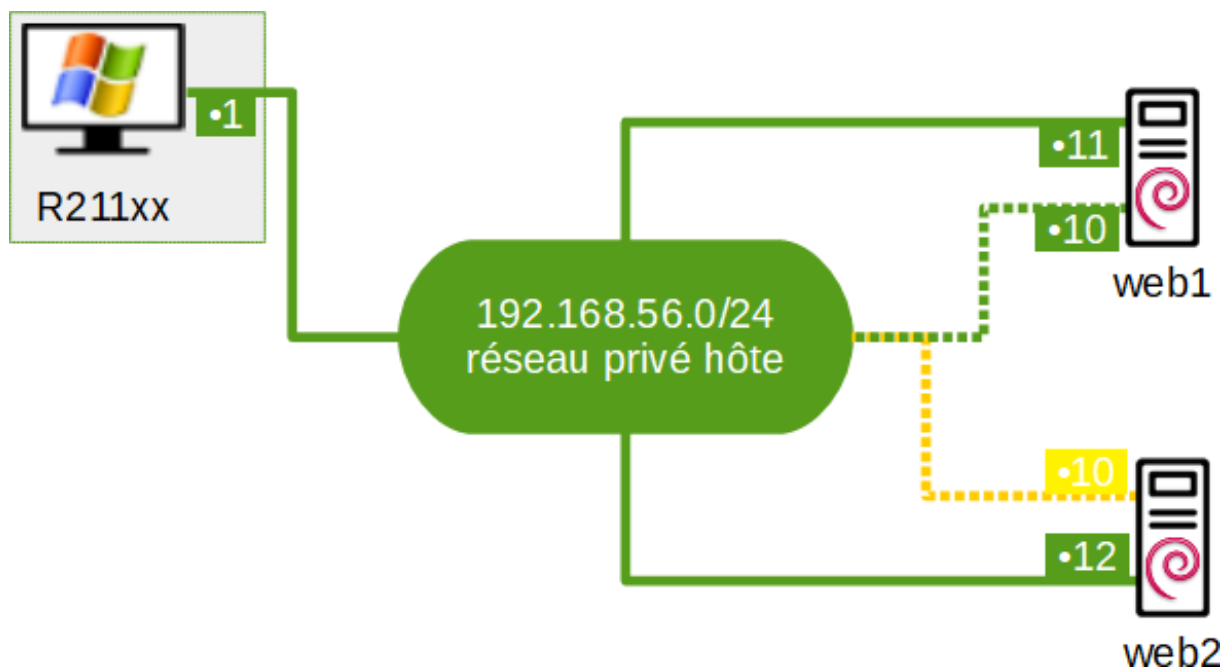
Attention avec cette valeur : si elle est trop courte, le système risque de s'auto déclarer mort. Si elle est trop grande, l'autre machine mettra un temps conséquent avant de s'en apercevoir et de reprendre la main.

- **Node** : Liste des machines utilisées pour la haute disponibilité, séparer par des espaces.
- **Le battement de cœur** sur Heartbeat est présent sur le « serveur maître » et sur tous les autres serveurs disponibles.

Mise en place de heartbeat sur deux serveurs pour assurer la redondance en cas de pannes d'un des deux

- Objectifs

- Heartbeat va servir à assurer la redondance entre les deux serveurs si le serveur A lâche le serveur B prend le relai.
- Réaliser le schéma ci-dessous pour cela il faut disposer de deux **srv-web 1 et 2** (2 machines Debian)
- Ainsi qu'une machine hôte sur le schéma ci-présent qui est la **Windows 10**.



- Prérequis

- Créer deux machine **Debian** (Srv web1 et Srv web2)
- Installer **Heartbeat** qui est le load balancer sur chacune des 2 machines **web 1** et **web 2** (apt install heartbeat)
- Installer apache2 (le serveur apache sert à exécuter un site web.)
- Les trois fichiers demandés doivent se trouver dans les répertoires.
- **!! Attention** c'est 3 fichiers ne sont pas présent, il faut les créer à la main.

Aide : pour créer un fichier on peut utiliser **leaf** ou **Nano**

Création des trois répertoires et des données devant s'y trouver.

Note : La configuration sur chacune des machines est la même aussi bien sur la srv-web 1 que la srv-web 2.

- Premier répertoire à configurer « /etc/ha.d/ha.cf »

A quoi sert ce fichier ?

Le fichier ici présent va permettre de repérer qu'il y a un problème sur l'un des nœuds.

Une fois sur les machines srv web 1 et web 2, entrer la commande ci-dessous.

« **Nano /etc/ha.d/ha.cf** »

« **Leaf /etc/ha.d/ha.cf** »

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

- Le fichier doit regrouper les informations suivantes :

Bcast enp0s3 (l'interface réseau)
Deadtime 5
Keepalive 1
Node web1 web2

Rappel:

- **Keepalive** : intervalle entre 2 battements de cœur. La valeur est en seconde par défaut. Pour la spécifier en millisecondes, on rajoutera 'ms' derrière. (Par exemple, 200 ms).
- **Deadtime** : Temps nécessaire avant de considérer qu'un nœud est mort. Le temps est en seconde 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, séparer par des espaces.

- Second répertoire à configurer « /etc/ha.d/authkeys »

Le second fichier à modifier est celui-ci

 **/etc/ha.d/authkeys**

A quoi sert ce fichier ?

Ce fichier contient une **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.

Auth 1
1md5 motdepasse

```
auth 1
1md5 motdepasse
```

Attention ! le service **heartbeat** exige une protection supplémentaire de ce fichier sinon il ne démarrera pas et sera visible par n'importe qui.

Chmod 600 /etc/ha.d/authkeys

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

Cette commande ne donne aucune autorisation à d'autres utilisateurs.

- Troisième fichier à configurer « /etc/ha.d/haresources »

 **/etc/ha.d/haresources**

A quoi sert ce fichier ?

Liste des ressources (Adresse virtuelles et services concernés) fournie par la grappe. La configuration sur chacune des machines est la même. Ce nom doit être le même pour les deux machines. C'est le nom de la machine qui sera activée par défaut au démarrage de heartbeat.

« **Web1** **IPaddr ::192.168.56.10 apache2** »

- Fichier « **hosts** »

Web1 et **web2** doivent être déclaré dans **nano /etc/hosts** (excepté si un service **DNS** est installé)

```
127.0.0.1      localhost
127.0.1.1      web1_
192.168.56.12  web2
```

- Sur la machine web 1

```
127.0.0.1      localhost
127.0.1.1      web2_
192.168.56.11  web1
```

- Sur la machine Web 2

- Le service Heartbeat est -il bien actif ?

Sur les deux machines :

```
root@srv-web1:~# systemctl stop apache2.services
```

« *Systemctl stop apache2.services* »

Il faut stopper le service apache2 sur les deux serveurs (machines)

Puis :

« *Systemctl status apache2.service* »

```
root@srv-web2:~# systemctl status apache2.service
```

Pour vérifier que le service apache2 est bien inactif.

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

```
root@srv-web2:~# systemctl disable apache2.service
```

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

```
root@srv-web1:~# systemctl restart heartbeat.service_
```

Puis vérifier qu'elle est active.

« *Sytemclt status hearbeat.service* »

Load balancing ou répartiteur de charge.

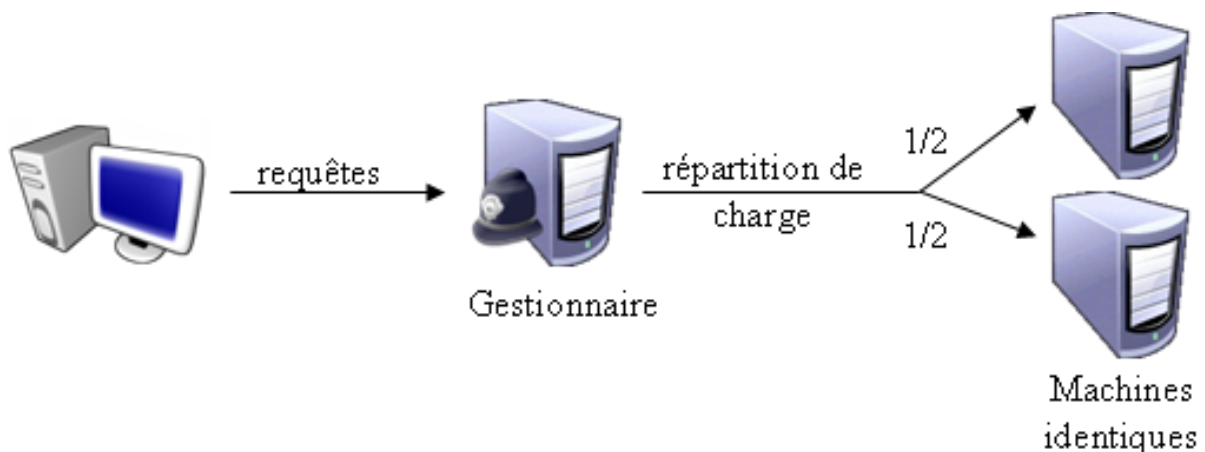
- Tout d'abord qu'est-ce que le load balancing ?

Le load balancing ou **répartiteur de tâches** (répartition de tâches) en utilisant une définition globale, permet de répartir différentes tâches sur un ensemble de ressources.

Pour faire plus simple une répartition de charges a pour but de répartir les différentes demande, charge ou tâches entre les différents clusters ou grappe de serveurs présent sur un réseau.

Exemple : plusieurs poste client envoie des requêtes/demande, c'est ensuite le gestionnaire qui va répartir les requêtes en fonction des ressources (capacités) des serveurs présents dans la grappe.

Schéma d'un système de load balancing 1.

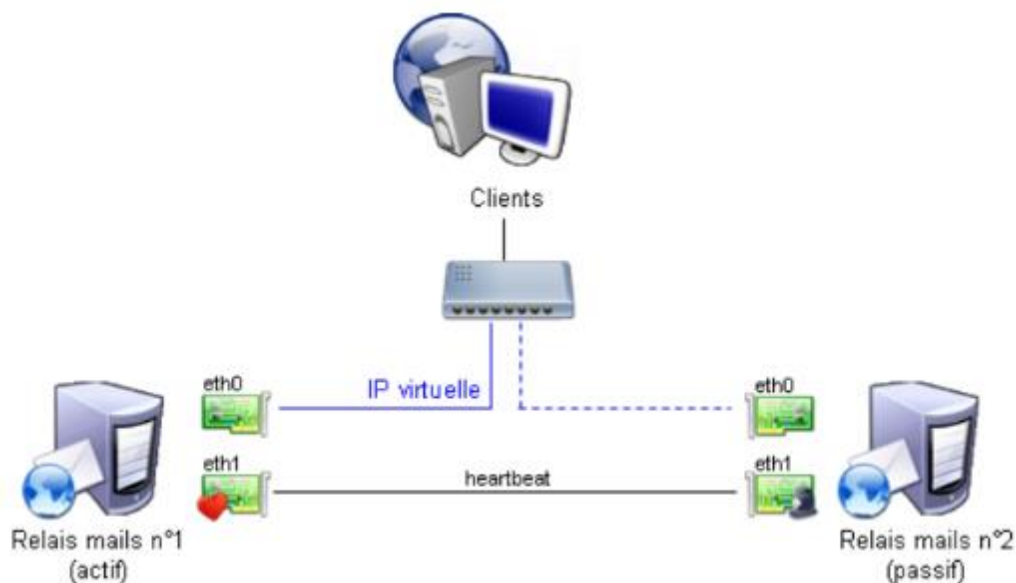


- Sur le schéma ci-dessus, le gestionnaire reparti les charges de façon équitable avec deux machines disposant de mêmes ressources.
- On peut parler ici de **Round Robin** ou **tourniquet**.



- A l'inverse sur ce schéma, des machines avec des ressources différentes n'auront pas la même répartition de charges, si une des deux machines (Serveurs) disposent de plus de ressources (en termes de capacité ou autre...) elle aura plus de charges puisqu'elle peut traiter et s'occuper d'une quantité plus importante de charges.

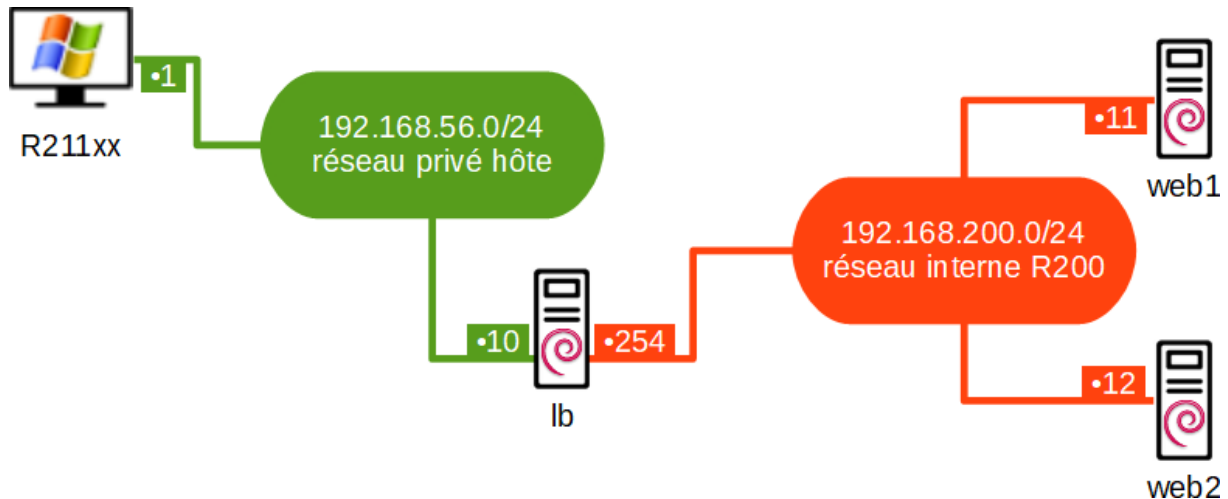
Schéma d'un système de load balancing 2.



- 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.

Mise en place d'un load balancing (Schéma de l'infrastructure à réaliser sur oracle)



Objectifs

- Mise en place d'une Debian LB (load balancing) pour la répartition de charges.
- Durant la réalisation il n'y a aucune manipulation à réaliser sur les autres serveurs mise à part sur le load balancer.

Prérequis

- Créer les machines Srv web 1 et srv web 2 (elles ont déjà été créées au préalable lors du TP sur la redondance).
- Connexion à internet.
- Réaliser un « ***Apt update*** » pour les mises à jour.
- Installer le paquet « ipvsadm » (Apt install ipvsadm)

Il faut que la Debian LB dispose de deux interfaces réseaux

☒ Activer l'interface réseau

Mode d'accès réseau : Réseau privé hôte ▼

Nom : VirtualBox Host-Only Ethernet Adapter ▼

Puis

☒ Activer l'interface réseau

Mode d'accès réseau : Réseau interne ▼

Nom : R200 ▼

- Une fois les deux interfaces rajouter sur la configuration de la machine directement sur VirtualBox il faut rajouter cette deuxième interface sur la Debian LB (pour load balancer) dans la ligne de commande. Pour cela accéder au fichier « **Nano /etc/network/interfaces** ».

```
#The 2nd network interfaces enp0s8
allow-hotplug enp0s8
iface enp0s8 inet static
address 192.168.200.254/24
```

- Il faut rajouter ces lignes sur le fichier pour que la deuxième interfaces réseau soit prise en charge par la machine virtuelle.

Par la suite il faut se rendre dans le fichier :

```
root@lb:~# nano /etc/network/sysctl.conf_
```

Et enlever le # sur la ligne.

- Fichier avant la manipulation :

```
#net.ipv4.ip_forward=1
```

- Fichier après la manipulation.

```
net.ipv4.ip_forward=1
```

- Le **1** va permettre d'activer le routage.

- **Vérification du load balancing**, pour cela il faut rentrer la commande suivante :

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

- Si le **1** est affiché l'activation du routage à fonctionner

```
1
```

- Premier fichier à configurer pour le Load balancing.

 `/etc/default/ipvsadm`

Le fichier est à configurer de la sorte.

```
# ipvsadm

# if you want to start ipvsadm on boot set this to true
AUTO="true"

# daemon method (none|master|backup)
DAEMON="master"

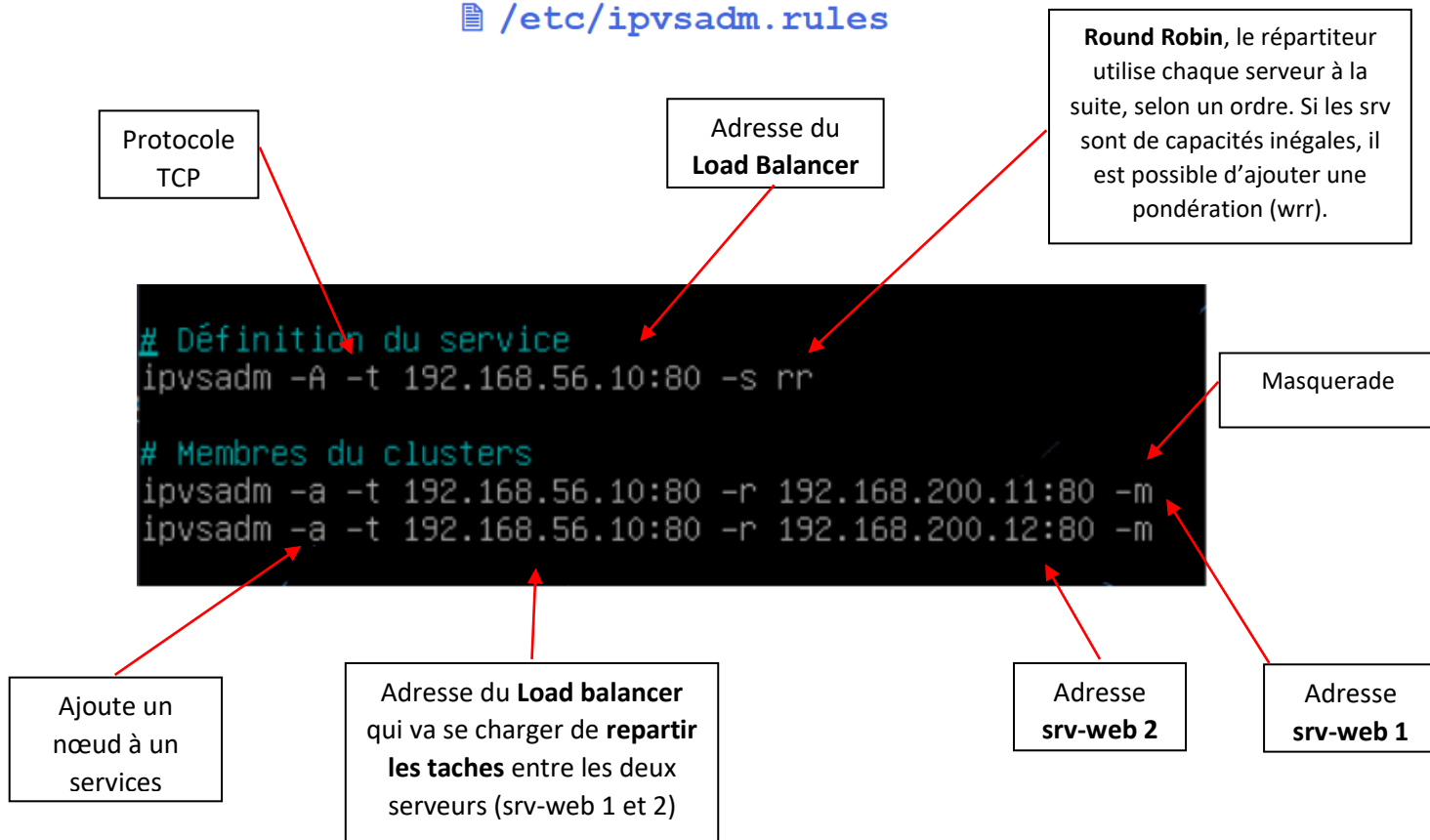
# use interface (eth0,eth1...)
IFACE="enp0s3"

# syncid to use
# (0 means no filtering of syncids happen, that is the default)
# SYNCID="0"
```

- « **True** » = Chargement de l'application et des règles aux démarrages.
- « **Master** » = On met le load balancer en « maitre »
- **IFACE** = « enp0s3 » = c'est l'interface qui va englober les requêtes envoyer par le serveur WEB.

- Second fichier à configurer pour le Load balancing.

/etc/ipvsadm.rules



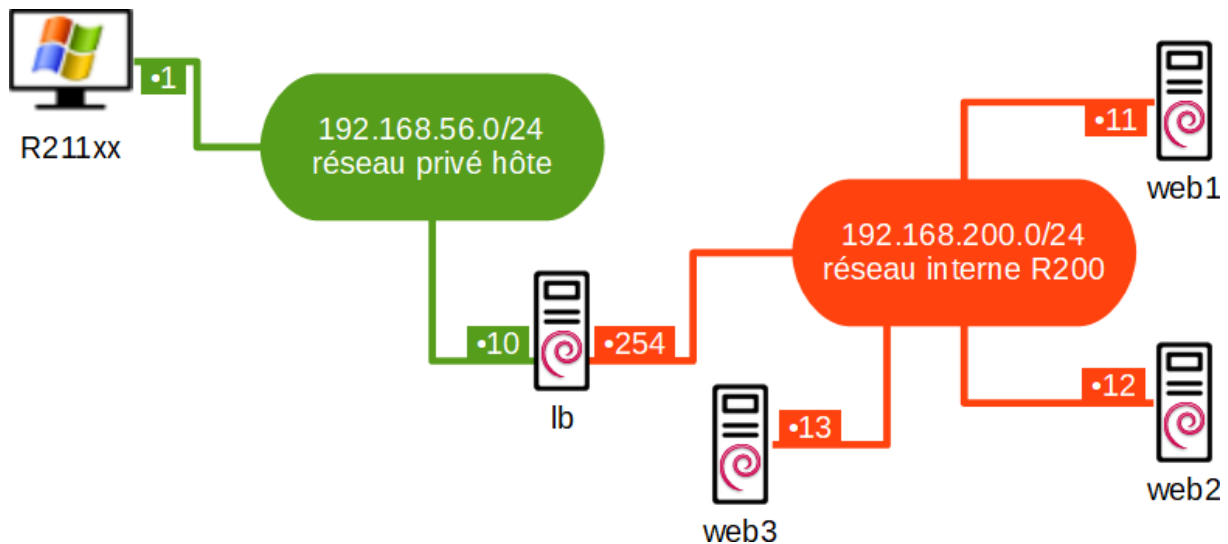
- Vérification du paramétrage • « `ipvsadm -ln` »

```
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
```

Si vos adresses de machine sont affichées avec un 1 sur la ligne Weight c'est que la configuration est correcte.

Mise en place d'une machine Web 3 et ajout au schéma du fonctionnement du Load balancing

Schéma à réaliser sur Virtual box :



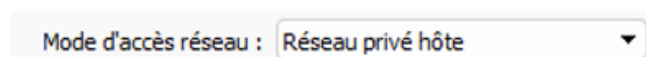
- La Debian **Srv-Web 1** prendra l'adresse IP **192.168.56.11**
- La Debian **Srv-Web 2** prendra l'adresse IP **192.168.56.12**
- La Debian **Srv-Web 3** prendra l'adresse IP **192.168.56.13**

Objectifs :

- Ici la **Debian LB** va répartir les tâches avec un nouveau serveur dans le fonctionnement du réseau, ici la **Debian srv-web 3**.

Prérequis

- L'interface réseau 1 (de la **Debian LB**) sera en **réseau privé hôte** et permettra de communiquer avec le PC qui demande des requêtes, le **PC R211** en conséquence sur le schéma.



- La seconde interface réseau sera en **réseau interne R200** qui permettra au load balancer (donc à la **Debian LB**) de répartir les tâches sur les différents serveurs (**Web 1, Web 2, Web 3**)

Mode d'accès réseau : Réseau interne ▼

Réalisation :

- Créer une machine **Debian Srv-web 3** (Pour cela prendre une image d'une **Debian Buster 10**)
- Modifier son nom en srv web 3 dans le fichier « **nano /etc/Hostname** »
- Modifier son adresse réseau dans le fichier **nano /etc/network/interfaces** (adresse IP de la srv-web 3 : « **192.168.200.13** »)

Une fois les changements réseau effectuer installer apache2. A l'aide de la commande « **apt install apache2.** »

- Pour finir il faut se rendre sur la machine Debian LB pour ajouter l'adresse IP de la machine Debian srv web 3 de la même manière que les deux autres srv web 1 et 2. (le fichier dans lequel se trouve les adresses IP et le fichier :

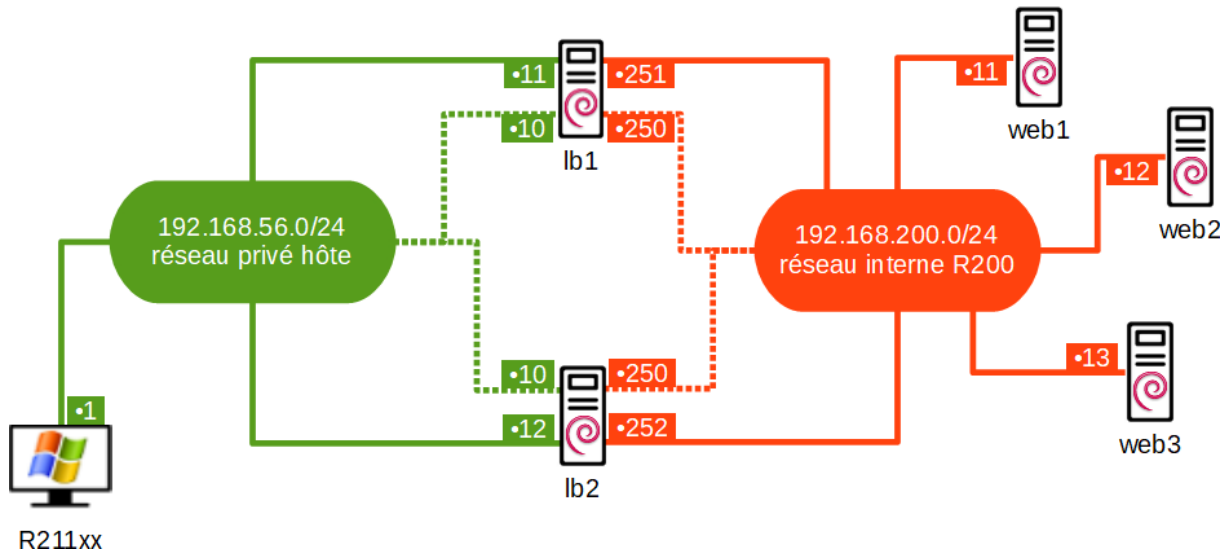
 **/etc/ipvsadm.rules**

```
# Définition du service
ipvsadm -A -t 192.168.56.10:80 -s rr
# Membres du clusters
ipvsadm -a -t 192.168.56.10:80 -r 192.168.200.11:80 -m
ipvsadm -a -t 192.168.56.10:80 -r 192.168.200.12:80 -m
ipvsadm -a -t 192.168.56.10:80 -r 192.168.200.13:80 -m
```

Adresse
srv-web 3

Une fois l'adresse IP de la Debian srv-web 3 entrer dans le fichier **ipvsadm.rules**, enregistrer les modifications du fichier puis **reboot** la machine de tel sorte à ce que la nouvelle configuration soit prise en compte.

Mise en place d'un deuxième Load balancer sur le schéma de l'infrastructure



Prérequis

Dans se schéma là il faudra disposer :

- D'une machine **srv web 1** avec tous les configurations effectués (Adresse ip **192.168.200.11**, changements des fichiers...)
- D'une machine **srv web 2** avec tous les configurations effectués (Adresse ip **192.168.200.12**, changements des fichiers...)
- D'une machine **srv web 3** avec tous les configurations effectués (Adresse ip **192.168.200.13**, changements des fichiers...)
- D'une **Debian LB 1** avec les interfaces réseau et les fichiers configurer comme dans les réalisations ci-dessus
- Et pour finir une nouvelle machine **Debian LB 2** qui va servir de deuxième load balancer.

Réalisation :

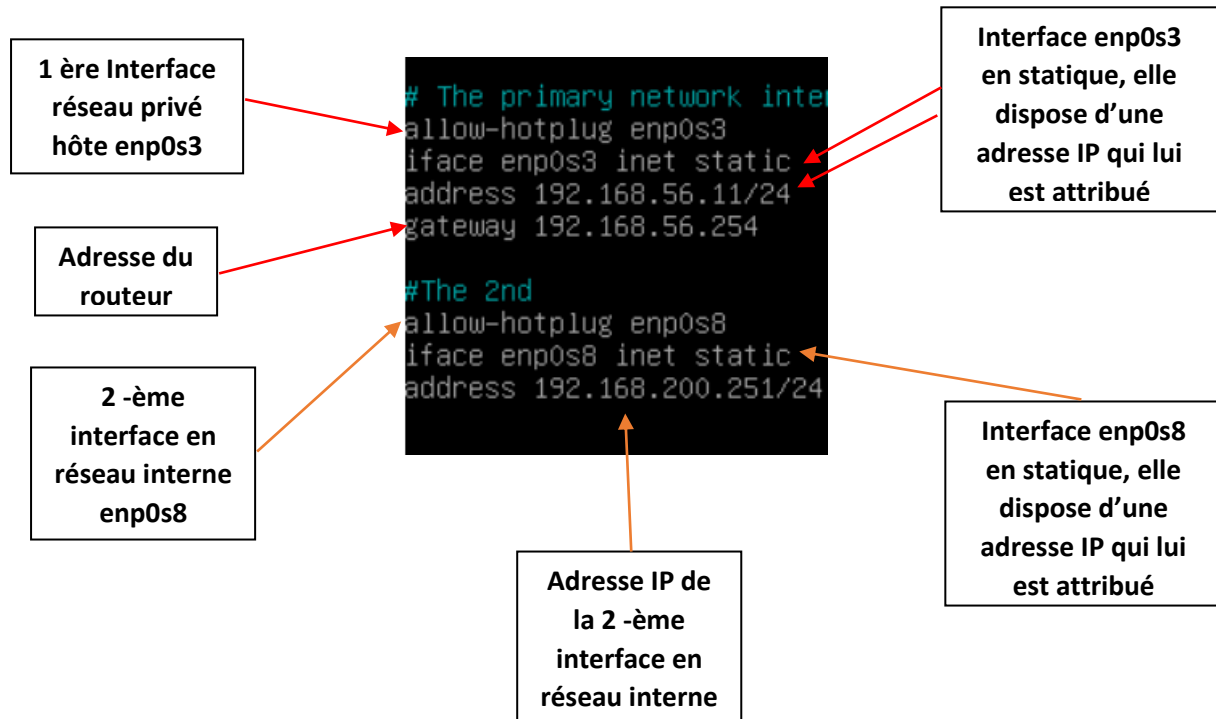
- Créer une nouvelle machine **Debian LB 2** à l'aide d'une iso d'une **Debian Buster 10**.
- Comme la **Debian LB 1** mettre la première interface en réseau privé hôte.

Mode d'accès réseau : Réseau privé hôte

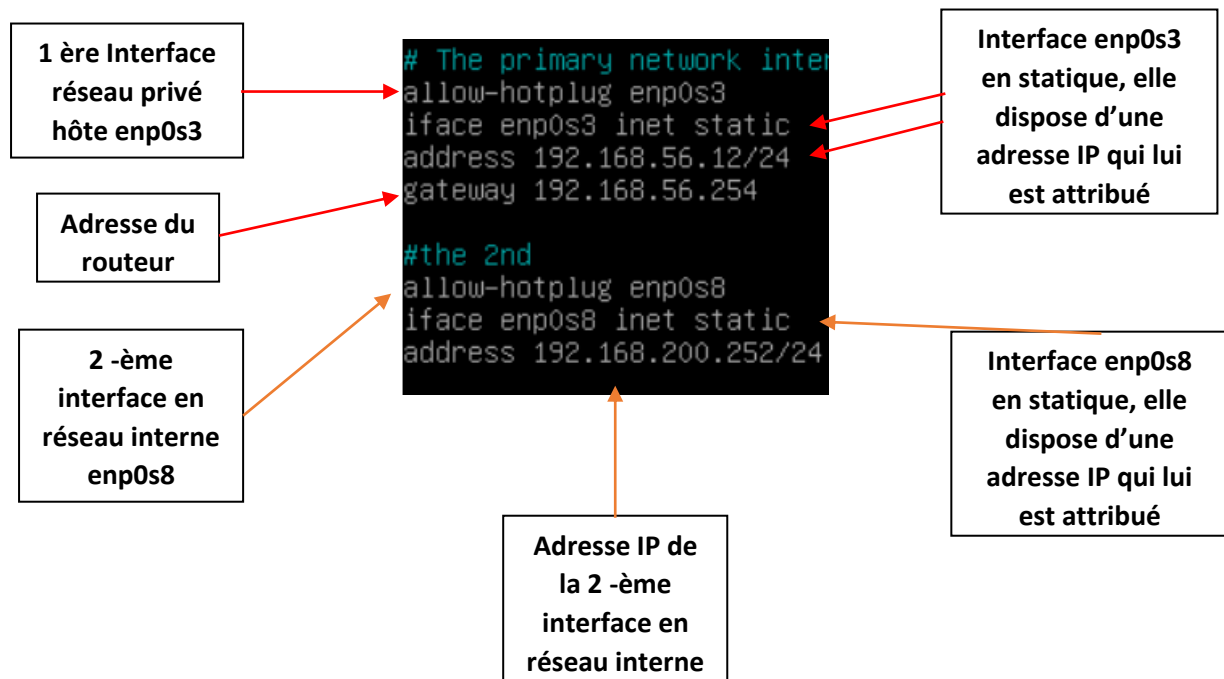
- Et la seconde en réseau interne « R200 »

Mode d'accès réseau : Réseau interne

Fichier de configuration Debian LB 1 « nano /etc/network/interfaces »



Fichier de configuration Debian LB 2 « nano /etc/network/interfaces »



- Par la suite installer « **heartbeat** » sur les deux load balancer (Debian LB 1 et Debian LB 2) à l'aide de la commande **apt install heartbeat** ainsi que « **ipvsadm** »
- Fichier à configurer sur les deux LB de la même manière, « **/etc/ha.d/ha.cf** »

A quoi sert se fichier ?

Le fichier ici présent va permettre de repérer qu'il y a un problème sur l'un des nœuds.

Une fois sur les machines srv web 1et web 2, entrer la commande ci-dessous.

« **Nano /etc/ha.d/ha.cf** »

« **Leaf /etc/ha.d/ha.cf** »

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

```
bcast enp0s3
deadtime 5
keepalive 1
node LB1 LB2
```

- Second fichier à configurer sur les deux LB de la même manière
« **/etc/ha.d/authkeys** »

Le second fichier à modifier est celui-ci

 **/etc/ha.d/authkeys**

A quoi sert ce fichier ?

Ce fichier contient une **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.

Auth 1 1md5 motdepasse

```
auth 1
1md5 motdepasse
```

Attention ! le service **heartbeat** exige une protection supplémentaire de ce fichier sinon il ne démarrera pas et sera visible par n'importe qui.

Chmod 600 /etc/ha.d/authkeys

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

Cette commande ne donne aucune autorisation à d'autres utilisateurs.

- Troisième fichier à configurer sur les deux LB de la même manière « /etc/ha.d/haresources »

 **/etc/ha.d/haresources**

A quoi sert ce fichier ?

Liste des ressources (Adresses virtuelles et services concernés) fournie par la grappe. La configuration sur chacune des machines est la même. Ce nom doit être le même pour les deux machines. C'est le nom de la machine qui sera activée par défaut au démarrage de heartbeat.

- Il faut faire la même chose mais changer le nom de la machine en LB 2 pour le deuxième load balancer.

Adresse **IP flottante** de
la première interface
(du LB 1 et LB 2)

```
LB1 IPaddr::192.168.56.10 apache2  
LB1 IPaddr::192.168.200.250 enp0s8
```

Adresse **IP flottante** de la
deuxième interface
(du LB 1 et LB 2)

- Fichier à configurer sur les deux LB de la même manière « hosts »

Web1 et **web2** doivent être déclaré dans **nano /etc/hosts** (excepté si un service **DNS** est installé)

- Sur la machine web 1

```
127.0.0.1      localhost
127.0.1.1      LB1
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
```

- Sur la machine Web 2

```
127.0.0.1      localhost
127.0.1.1      LB2
192.168.56.11  LB1
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
```

Vérification du load balancing et activation du routage sur les deux Debian LB.

Par la suite il faut se rendre dans le fichier :

```
root@lb:~# nano /etc/network/sysctl.conf_
```

Et enlever le # sur la ligne.

- Fichier avant la manipulation :

```
#net.ipv4.ip_forward=1
```

- Fichier après la manipulation.

```
net.ipv4.ip_forward=1
```

- Le 1 va permettre d'activer le routage.

- **Vérification du load balancing**, pour cela il faut rentrer la commande suivante :

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

- Si le **1** est affiché l'activation du routage à fonctionner

```
1
```

Premier fichier à configurer sur les deux LB de la même manière.

 **/etc/default/ipvsadm**

Le fichier est à configurer de la sorte.

```
# 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"

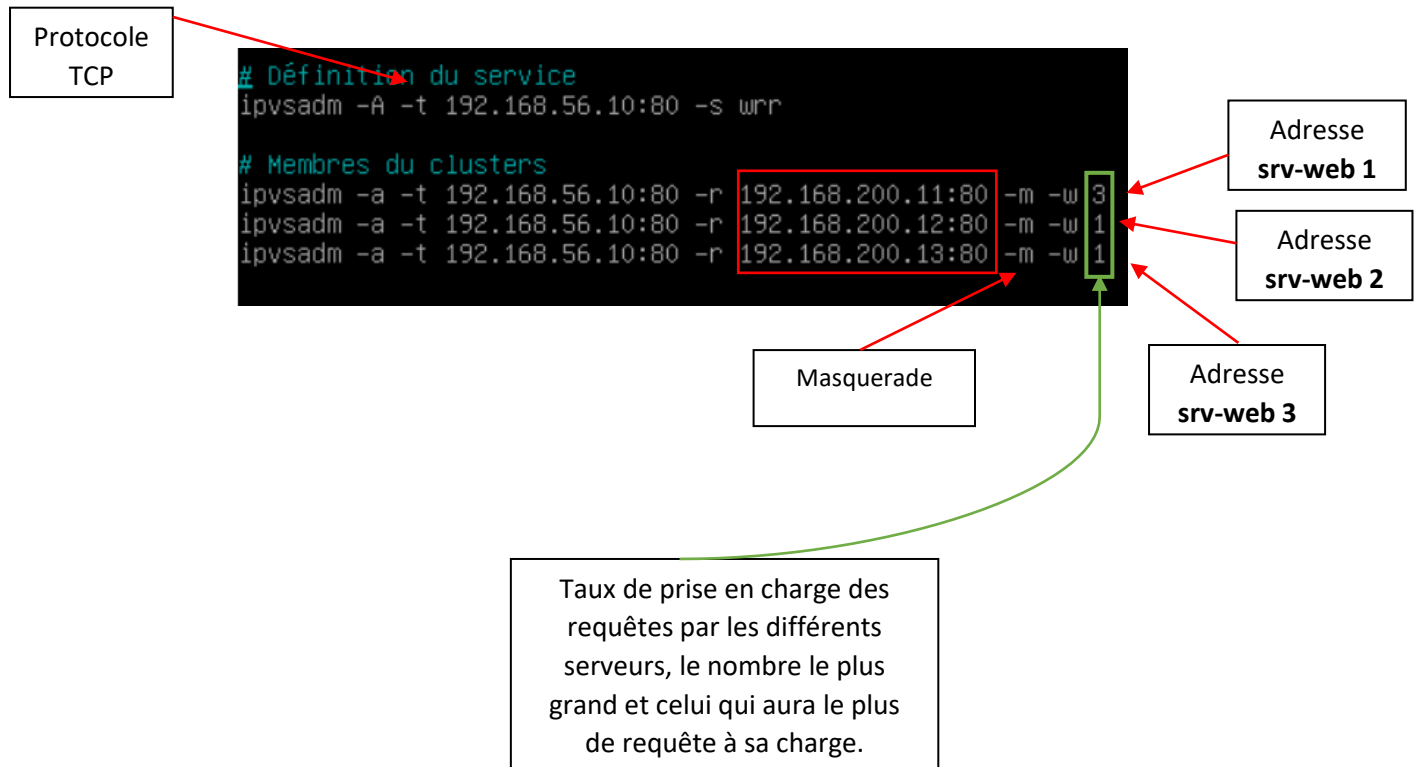
# syncid to use
# (0 means no filtering of syncids happen, that is the default)
# SYNCID="0"
```

- « **True** » = Chargement de l'application et des règles aux démarrages.
- « **Master** » = On met le load balancer en « maître »
- **IFACE** = « enp0s3 » = c'est l'interface qui va englober les requêtes envoyées par le serveur WEB.

- Second fichier à configurer sur les deux LB de la même manière

 **/etc/ipvsadm.rules**

Round Robin, le répartiteur utilise chaque serveur à la suite, selon un ordre. Si les srv sont de capacités inégales, il est possible d'ajouter une pondération (wrr).



Vérification du paramétrage • « Ipvsadm -ln »

```
root@LB2:~# ipvsadm -ln
IP Virtual Server version 1.2.1 (size=4096)
Prot LocalAddress:Port Scheduler Flags
-> RemoteAddress:Port      Forward Weight ActiveConn InActConn
TCP  192.168.56.10:80 wrr
-> 192.168.200.11:80      Masq    3      0      0
-> 192.168.200.12:80      Masq    1      0      0
-> 192.168.200.13:80      Masq    1      0      0
```

Pour que les changements aient lieu il faut reboot les machines (Debian LB1 et LB2)

Puis pour finir lorsque l'on souhaite vérifier le fonctionnement du load balancing taper l'adresse suivante 192.168.56.10 sur internet, si un serveur répond c'est que le load balancing fonctionne.

Correction devoir BTS BLANC

MPLS est aujourd'hui obsolète, il utilise différent protocole qui permet d'acheminer les paquets sur le réseau. Le FAI (fournisseur d'accès internet) dote les paquets d'un étiquetage qui permet de déterminer à l'avance le chemin à utiliser.

Un fichier de zone regroupe la configuration du DNS

DNS = Annuaire/ traduit le nom de domaine en adresse IP

Un fichier de zone est un annuaire qu'on fournit à l'utilisateur.

ARFC, est une norme qui préconise la mise en place d'au moins deux serveur DNS faisant autorité afin de garantir une tolérance de panne accrue. C'est

Il convient de répartir géographiquement les mêmes serveurs pour qu'il ne soit pas dépendant des mêmes structures ou de même prestataire.

Il serait pertinent d'avoir des serveurs DNS répartis dans des Datacenter différents, ne passant pas par les mêmes FAI.

Pour finir cela permettra de limiter les impacts d'incident physique et environnementaux.

Question 1.2 : ces enregistrements du fichier de zone ont été réalisés dans le but de mettre en œuvre une répartition de charges appelée « round robin » ou tourniquet afin d'assurer une répartition égale entre les deux serveurs web de l'entreprise. Ainsi lorsqu'un utilisateur rentrera le nom du site dans sa barre url, alors il sera dirigé via le serveur DNS vers l'un ou l'autre serveur, mais qui disposera du même contenu.

HAProxy

Open source qui fournit un équilibreur de charge haute disponibilité et un proxy inverse pour les applications TCP et HTTP qui répartissent les requêtes sur plusieurs serveurs.

La situation actuelle n'empêchera pas une indisponibilité simultanée, des serveurs web si un élément d'infrastructure du prestataire bdn est touché puisqu'ils sont dans le même Datacenter.

Un des deux serveurs web doit être installé dans un autre Datacenter, on peut également imaginer la création d'un troisième serveur web dans un autre site distant.

Mission 2 : Question A.2.1

Le compte root est un compte qui dispose de tous les droits (administrateur) de ce fait SSH n'autorise pas les connexions à l'aide de ce compte. En référence aux bonnes pratiques, la connexion distante en root par l'intermédiaire du protocole SSH est interdite, en effet si un attaquant arrive à s'introduire sur le système par le biais de ce compte il possèdera tous les droits sur le système en question.

L'apparition du message d'avertissement lors d'une première connexion depuis un client SSH. Message normal pour la première connexion.

Le pc client veut se connecter en SSH vers le serveur.

Lors de cette première connexion le serveur envoie l'empreinte contenant la clé publique, cette clé publique une fois envoyée est stockée sur le client dans le fichier **.ssh/known_hosts** et on enregistrera la connexion entre, le client et le nom de domaine ou clé. Si jamais en renouvelant l'opération l'empreinte n'est pas identique il y a de forte chance d'avoir affaire à une attaque man on the middle.

Le client SSH prévient que l'empreinte de la clé publique du serveur a changé par rapport à celle qui avait été acceptée lors de la première connexion.

Le serveur (certainement de remplacement) possède visiblement une paire de clés différentes du serveur original.

Nous ne sommes donc pas en présence d'une attaque man on the middle, mais simplement d'un changement de machine entraînant la présence d'une paire de clés différentes sur la machine différente.

Question 5) Comment limiter l'impact de ces impacts

- Mettre en place une authentification par clé plutôt que par mot de passe
- Changer le port d'écoute par défaut du serveur SSH.
- Intégrer une temporisation croissante entre deux tentatives erronées.
- Tout autre système de surveillance active des journaux.