

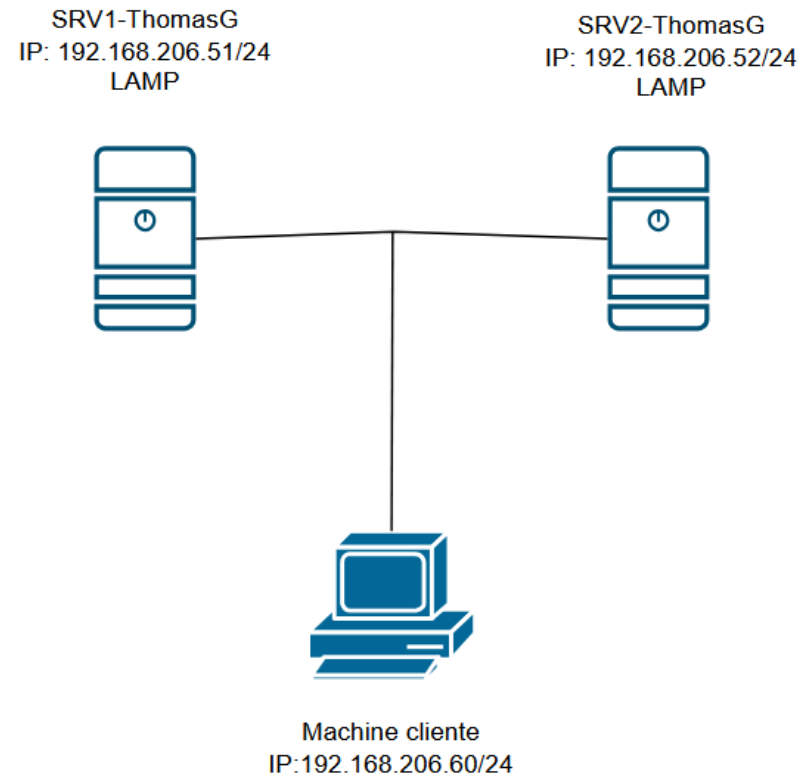
# TP-Haute disponibilité

---

THOMAS GRZESINSKI

# Schéma réseau

---



# La haute disponibilité c'est quoi ?

---

La haute disponibilité désigne une architecture système ou un service conçu pour assurer un taux de disponibilité optimal. Par exemple, dans une entreprise de commerce en ligne, les serveurs doivent être redondants et équipés de solutions de basculement automatique afin de garantir un accès continu aux clients, même en cas de panne d'un serveur.

Nous allons donc utilisé l'outil heartbeat qui est un signal périodique utilisé en haute disponibilité pour vérifier l'état des serveurs. S'il ne répond plus, un basculement (failover) est déclenché pour assurer la continuité du service.

# Configuration de l'infrastructure WEB

Nous allons donc maintenant configuré deux serveurs web sur laquelle nous allons mettre de la haute disponibilité en place.

## Configuration :

- Vous devez tout d'abord avoir deux machines DEBIAN 12 qui nous serviront de serveurs
- Sur ces deux machines mettez à jour les paquets a l'aide la commande **apt update** et ensuite installé un LAMP avec la commande **apt install apache2**
- Démarrer vos services web a l'aide de la commande **systemctl start service**
- Vérifier le status de vos serveurs web à l'aide de la commande **systemctl status Apache2**

Serveur1 :

```
root@SRV1-Thomas6:~# systemctl status apache2
● apache2.service - The Apache HTTP Server
   Loaded: loaded (/lib/systemd/system/apache2.service; enabled; preset: enabled)
   Active: active (running) since Fri 2025-03-07 17:01:06 CET; 4 days ago
     Docs: https://httpd.apache.org/docs/2.4/
   Process: 431 ExecStart=/usr/sbin/apachectl start (code=exited, status=0/SUCCESS)
   Process: 395 ExecReload=/usr/sbin/apachectl graceful (code=exited, status=0/SUCCESS)
   Main PID: 485 (apache2)
     Tasks: 55 (limit: 2305)
         Memory: 17.6M
         CPU: 38.004s
   CGroup: /system.slice/apache2.service
           └─ 485 /usr/sbin/apache2 -k start
             └─ 4892 /usr/sbin/apache2 -k start
               └─ 4893 /usr/sbin/apache2 -k start

mars 09 00:00:27 SRV1-Thomas6 systemd[1]: Reloaded apache2.service - The Apache HTTP Server.
mars 10 00:00:20 SRV1-Thomas6 systemd[1]: Reloading apache2.service - The Apache HTTP Server.
mars 10 00:00:21 SRV1-Thomas6 apache2[13023]: AH00558: apache2: Could not reliably determine the server's fully qualified domain name, using 127.0.1.1. Set the 'ServerName' directive globally to suppress this message
mars 10 00:00:21 SRV1-Thomas6 systemd[1]: Reloaded apache2.service - The Apache HTTP Server.
mars 11 00:00:15 SRV1-Thomas6 systemd[1]: Reloading apache2.service - The Apache HTTP Server.
mars 11 00:00:15 SRV1-Thomas6 apache2[14021]: AH00558: apache2: Could not reliably determine the server's fully qualified domain name, using 127.0.1.1. Set the 'ServerName' directive globally to suppress this message
mars 11 00:00:15 SRV1-Thomas6 systemd[1]: Reloaded apache2.service - The Apache HTTP Server.
mars 12 00:00:08 SRV1-Thomas6 systemd[1]: Reloading apache2.service - The Apache HTTP Server.
mars 12 00:00:09 SRV1-Thomas6 apache2[15009]: AH00558: apache2: Could not reliably determine the server's fully qualified domain name, using 127.0.1.1. Set the 'ServerName' directive globally to suppress this message
mars 12 00:00:09 SRV1-Thomas6 systemd[1]: Reloaded apache2.service - The Apache HTTP Server.
lines 1-25/25 (END)
```

Serveur2 :

```
root@SRV2-Thomas6:~# systemctl status apache2
● apache2.service - The Apache HTTP Server
   Loaded: loaded (/lib/systemd/system/apache2.service; enabled; preset: enabled)
   Active: active (running) since Fri 2025-03-07 17:06:27 CET; 4 days ago
     Docs: https://httpd.apache.org/docs/2.4/
   Process: 431 ExecStart=/usr/sbin/apachectl start (code=exited, status=0/SUCCESS)
   Process: 430 ExecReload=/usr/sbin/apachectl graceful (code=exited, status=0/SUCCESS)
   Main PID: 479 (apache2)
     Tasks: 55 (limit: 2305)
         Memory: 17.6M
         CPU: 37.306s
   CGroup: /system.slice/apache2.service
           └─ 479 /usr/sbin/apache2 -k start
             └─ 4584 /usr/sbin/apache2 -k start
               └─ 4585 /usr/sbin/apache2 -k start

mars 09 00:00:16 SRV2-Thomas6 systemd[1]: Reloaded apache2.service - The Apache HTTP Server.
mars 10 00:00:10 SRV2-Thomas6 systemd[1]: Reloading apache2.service - The Apache HTTP Server.
mars 10 00:00:10 SRV2-Thomas6 apache2[12700]: AH00558: apache2: Could not reliably determine the server's fully qualified domain name, using 127.0.1.1. Set the 'ServerName' directive globally to suppress this message
mars 10 00:00:10 SRV2-Thomas6 systemd[1]: Reloaded apache2.service - The Apache HTTP Server.
mars 11 00:00:04 SRV2-Thomas6 systemd[1]: Reloading apache2.service - The Apache HTTP Server.
mars 11 00:00:04 SRV2-Thomas6 apache2[10517]: AH00558: apache2: Could not reliably determine the server's fully qualified domain name, using 127.0.1.1. Set the 'ServerName' directive globally to suppress this message
mars 11 00:00:04 SRV2-Thomas6 systemd[1]: Reloaded apache2.service - The Apache HTTP Server.
mars 12 00:00:16 SRV2-Thomas6 systemd[1]: Reloading apache2.service - The Apache HTTP Server.
mars 12 00:00:16 SRV2-Thomas6 apache2[14593]: AH00558: apache2: Could not reliably determine the server's fully qualified domain name, using 127.0.1.1. Set the 'ServerName' directive globally to suppress this message
mars 12 00:00:16 SRV2-Thomas6 systemd[1]: Reloaded apache2.service - The Apache HTTP Server.
lines 1-25/25 (END)
```

# Configuration de l'infrastructure WEB

---

- Maintenant vous allez donc devoir modifier l'adresse IP de vos deux machines serveurs pour y fixer des IP statiques.
- Aller dans le fichier ***nano /etc/network/interfaces***

## **Fichier de configuration réseaux des deux serveurs :**

**Serveur 1 :**

```
# The primary network interface
allow-hotplug ens18
iface ens18 inet static

address 192.168.206.51
netmask 255.255.255.0
```

**Serveur2 :**

```
# The primary network interface
allow-hotplug ens18
iface ens18 inet static

address 192.168.206.52
netmask 255.255.255.0
```

# Configuration de l'infrastructure WEB

---

Renommer vos deux machines pour faciliter la différence entre les deux serveurs

Aller dans le fichier ***nano /etc/hostname***

Renommer vos deux machines comme ceci :

```
SRV1-ThomasG SRV2-ThomasG
```

Relancé vos deux serveurs pour faire prendre en compte ces nouvelles modifications

Maintenant raccorder une machine Windows cliente sur le même réseau pour qu'elle puisse communiquer avec nos deux machines serveurs (voir [TP-Chapitre 3 : Mise en œuvre d'un réseau](#))

Test pour vérifier la communication des machines dans notre réseau :

# Configuration de l'infrastructure WEB

- Ping Serveur 1 vers Serveur 2 et Client :
- Ping Serveur 2 vers Serveur 1 et Client :
- Ping Client vers Serveur 1 et Serveur 2 :

```
root@SRV1-ThomasG:~# ping 192.168.206.52
PING 192.168.206.52 (192.168.206.52) 56(84) bytes of data.
64 bytes from 192.168.206.52: icmp_seq=1 ttl=64 time=2.11 ms
64 bytes from 192.168.206.52: icmp_seq=2 ttl=64 time=1.13 ms
64 bytes from 192.168.206.52: icmp_seq=3 ttl=64 time=1.11 ms
64 bytes from 192.168.206.52: icmp_seq=4 ttl=64 time=1.09 ms
64 bytes from 192.168.206.52: icmp_seq=5 ttl=64 time=0.988 ms
64 bytes from 192.168.206.52: icmp_seq=6 ttl=64 time=1.11 ms
```

```
root@SRV1-ThomasG:~# ping 192.168.206.60
PING 192.168.206.60 (192.168.206.60) 56(84) bytes of data.
64 bytes from 192.168.206.60: icmp_seq=1 ttl=128 time=1.11 ms
64 bytes from 192.168.206.60: icmp_seq=2 ttl=128 time=1.24 ms
64 bytes from 192.168.206.60: icmp_seq=3 ttl=128 time=1.23 ms
64 bytes from 192.168.206.60: icmp_seq=4 ttl=128 time=1.49 ms
64 bytes from 192.168.206.60: icmp_seq=5 ttl=128 time=1.25 ms
```

```
root@SRV2-ThomasG:~# ping 192.168.206.51
PING 192.168.206.51 (192.168.206.51) 56(84) bytes of data.
64 bytes from 192.168.206.51: icmp_seq=1 ttl=64 time=1.28 ms
64 bytes from 192.168.206.51: icmp_seq=2 ttl=64 time=1.14 ms
64 bytes from 192.168.206.51: icmp_seq=3 ttl=64 time=1.21 ms
64 bytes from 192.168.206.51: icmp_seq=4 ttl=64 time=1.13 ms
64 bytes from 192.168.206.51: icmp_seq=5 ttl=64 time=0.965 ms
```

```
root@SRV2-ThomasG:~# ping 192.168.206.60
PING 192.168.206.60 (192.168.206.60) 56(84) bytes of data.
64 bytes from 192.168.206.60: icmp_seq=1 ttl=128 time=1.74 ms
64 bytes from 192.168.206.60: icmp_seq=2 ttl=128 time=0.973 ms
64 bytes from 192.168.206.60: icmp_seq=3 ttl=128 time=0.777 ms
64 bytes from 192.168.206.60: icmp_seq=4 ttl=128 time=1.17 ms
64 bytes from 192.168.206.60: icmp_seq=5 ttl=128 time=1.31 ms
64 bytes from 192.168.206.60: icmp_seq=6 ttl=128 time=1.61 ms
64 bytes from 192.168.206.60: icmp_seq=7 ttl=128 time=1.41 ms
64 bytes from 192.168.206.60: icmp_seq=8 ttl=128 time=1.46 ms
```

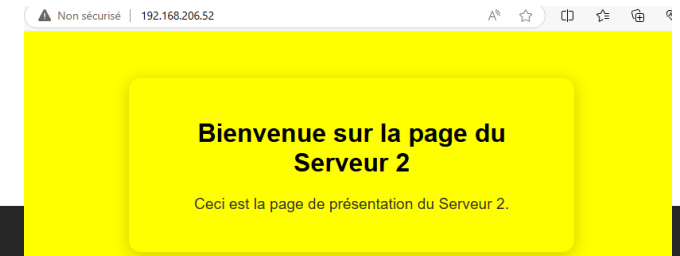
```
C:\Users\sio>ping 192.168.206.51
Envoi d'une requête 'Ping' 192.168.206.51 avec 32 octets de données :
Réponse de 192.168.206.51 : octets=32 temps=1 ms TTL=64
Réponse de 192.168.206.51 : octets=32 temps=1 ms TTL=64
Réponse de 192.168.206.51 : octets=32 temps=1 ms TTL=64
Réponse de 192.168.206.51 : octets=32 temps=1 ms TTL=64

Statistiques Ping pour 192.168.206.51:
    Paquets : envoyés = 4, reçus = 4, perdus = 0 (perte 0%),
    Durée approximative des boucles en millisecondes :
        Minimum = 1ms, Maximum = 1ms, Moyenne = 1ms
```

```
C:\Users\sio>ping 192.168.206.52
Envoi d'une requête 'Ping' 192.168.206.52 avec 32 octets de données :
Réponse de 192.168.206.52 : octets=32 temps=1 ms TTL=64
Réponse de 192.168.206.52 : octets=32 temps=1 ms TTL=64
Réponse de 192.168.206.52 : octets=32 temps=1 ms TTL=64
Réponse de 192.168.206.52 : octets=32 temps=1 ms TTL=64

Statistiques Ping pour 192.168.206.52:
    Paquets : envoyés = 4, reçus = 4, perdus = 0 (perte 0%),
    Durée approximative des boucles en millisecondes :
        Minimum = 1ms, Maximum = 1ms, Moyenne = 1ms
```

-Vérifier maintenant que nos deux serveurs sont opérationnelles en entrant leurs IP dans le navigateur web du client :



# Configuration de heartbeat

---

Maintenant que la première étape est terminée nous allons pouvoir mettre en place notre service heartbeat.

## Configuration :

Remettez vos deux machines en DHCP pour qu'elle est un accès internet pour pouvoir télécharger le service (Voir [TP-Chapitre 3 : Mise en œuvre d'un réseau](#) )

Mettez à jour vos paquets avec la commande `apt update` et ensuite installez le service heartbeat à l'aide de la commande **`apt-get install heartbeat`**

Une fois installé nous allons pouvoir mettre notre cluster en place

Tout d'abord il vous faudra modifier votre fichier hosts afin de faciliter la résolution de DNS lorsque l'on ping nos serveurs

# Configuration de heartbeat

Aller dans le fichier *nano /etc/hosts* et déclaré vos noms de machines avec leur IP associés

```
127.0.0.1    localhost
127.0.1.1    SRV2-ThomasG
192.168.206.51  SRV1-ThomasG
```

```
127.0.0.1    localhost
127.0.1.1    SRV1-ThomasG
192.168.206.52  SRV2-ThomasG
```

Pour la machine Windows aller dans le chemin *C:\Windows\System32\drivers\etc\hosts*

```
# localhost name resolution is handled within DNS itself.
#       127.0.0.1       localhost
#       ::1            localhost
192.168.206.51 SRV1-ThomasG
192.168.206.52 SRV2-ThomasG
```

Tester les pings avec le DNS maintenant, les machines devraient répondre

Serveur 1 :

```
root@SRV1-ThomasG:~# ping SRV2-ThomasG
PING SRV2-ThomasG (192.168.206.52) 56(84) bytes of data:
64 bytes from SRV2-ThomasG (192.168.206.52): icmp_seq=1 ttl=64 time=1.16 ms
64 bytes from SRV2-ThomasG (192.168.206.52): icmp_seq=2 ttl=64 time=1.22 ms
64 bytes from SRV2-ThomasG (192.168.206.52): icmp_seq=3 ttl=64 time=1.00 ms
64 bytes from SRV2-ThomasG (192.168.206.52): icmp_seq=4 ttl=64 time=1.10 ms
64 bytes from SRV2-ThomasG (192.168.206.52): icmp_seq=5 ttl=64 time=0.687 ms
64 bytes from SRV2-ThomasG (192.168.206.52): icmp_seq=6 ttl=64 time=1.86 ms
64 bytes from SRV2-ThomasG (192.168.206.52): icmp_seq=7 ttl=64 time=1.27 ms
```

Serveur 2 :

```
root@SRV2-ThomasG:~# ping SRV1-ThomasG
PING SRV1-ThomasG (192.168.206.51) 56(84) bytes of data:
64 bytes from SRV1-ThomasG (192.168.206.51): icmp_seq=1 ttl=64 time=1.03 ms
64 bytes from SRV1-ThomasG (192.168.206.51): icmp_seq=2 ttl=64 time=0.989 ms
64 bytes from SRV1-ThomasG (192.168.206.51): icmp_seq=3 ttl=64 time=1.07 ms
64 bytes from SRV1-ThomasG (192.168.206.51): icmp_seq=4 ttl=64 time=1.06 ms
64 bytes from SRV1-ThomasG (192.168.206.51): icmp_seq=5 ttl=64 time=1.13 ms
```

Machine Windows cliente :

```
C:\Users\sio>ping SRV1-ThomasG

Envoi d'une requête 'ping' sur SRV1-ThomasG [192.168.206.51] avec 32 octets de données :
Réponse de 192.168.206.51 : octets=32 temps=1 ms TTL=64
Réponse de 192.168.206.51 : octets=32 temps=1 ms TTL=64
Réponse de 192.168.206.51 : octets=32 temps=1 ms TTL=64
Réponse de 192.168.206.51 : octets=32 temps=1 ms TTL=64

Statistiques Ping pour 192.168.206.51:
    Paquets : envoyés = 4, reçus = 4, perdus = 0 (perte 0%),
    Durée approximative des boucles en millisecondes :
        Minimum = 1ms, Maximum = 1ms, Moyenne = 1ms
```

```
C:\Users\sio>ping SRV2-ThomasG

Envoi d'une requête 'ping' sur SRV2-ThomasG [192.168.206.52] avec 32 octets de données :
Réponse de 192.168.206.52 : octets=32 temps=1 ms TTL=64
Réponse de 192.168.206.52 : octets=32 temps=1 ms TTL=64
Réponse de 192.168.206.52 : octets=32 temps=1 ms TTL=64
Réponse de 192.168.206.52 : octets=32 temps=1 ms TTL=64

Statistiques Ping pour 192.168.206.52:
    Paquets : envoyés = 4, reçus = 4, perdus = 0 (perte 0%),
    Durée approximative des boucles en millisecondes :
        Minimum = 1ms, Maximum = 1ms, Moyenne = 1ms
```

## Configuration de heartbeat

---

Nous allons maintenant mettre en place notre cluster, pour que le cluster fonctionne correctement nous allons devoir maintenant mettre en place 3 fichiers.

- **Un fichier ha.cf** qui nous permettra de définir le comportement du cluster heartbeat et à spécifier les paramètres nécessaires pour assurer la haute disponibilité.
- **Un fichier authkeys** qui nous permettra de sécuriser la communication entre les nœuds du cluster. Il permet d'authentifier les messages échangés afin d'éviter qu'un serveur non autorisé ne puisse s'infiltrer dans le cluster.
- **Et un fichier haresources** qui nous permettra de définir quels services et quelles ressources seront gérés par le cluster et sur quel nœud ils doivent être actifs en cas de basculement.

Ces fichiers devront être identiques sur les deux serveurs une copie des fichiers par SSH est fortement recommandé.

## Configuration de heartbeat

---

Pour le fichier **ha.cf** aller dans **cd /etc/ha.d** créer un fichier à l'aide la commande **touch ha.cf** et ensuite entrer dans le fichier à l'aide de la commande **nano ha.cf**

Mettez y dedans :

```
GNU nano 7.2 ha.cf
logfile /var/log/ha-log  Enregistre les événements de Heartbeat dans le fichier /var/log/ha-log
logfacility local0      Définit la catégorie de logs utilisée par le système
keepalive 2            Envoie un signal (heartbeat) toutes les 2 secondes
deadtime 30           Si un nœud ne répond pas après 30 secondes, il est déclaré inactif
initdead 120          Au démarrage du cluster, un nœud peut être déclaré mort au bout de 100
                      secondes
bcast ens18          Utilise le mode broadcast sur l'interface
                      réseau
udpport 694           Utilise le port UDP 694 pour la communication entre les
                      nœuds.
auto_failback on      Active le retour automatique du nœud principal si celui-ci redevient
                      disponible.
node SRV1-ThomasG     Déclare les serveurs qui font partie du
node SRV2-ThomasG     cluster.
```

## Configuration de heartbeat

---

Pour le fichier **authkeys** aller dans **cd /etc/ha.d** créer un fichier à l'aide la commande **touch authkeys** et ensuite entrer dans le fichier à l'aide de la commande **nano authkeys**

Mettez y dedans :

```
GNU nano 7.2 authkeys
auth 2 Définit la méthode d'authentification utilisée.
2 sha1 test-ha Définit la clé d'authentification pour l'option 2
```

Mettez y les droits **chmod 600 /etc/ha.d/authkeys**

Il est important de mettre les droits sur le fichier pour pas qu'un utilisateur non autorisé accède au fichier, et lise la clé pour pouvoir ensuite ensuite falsifier des messages heartbeat qui pourrait donc perturber le cluster.

## Configuration de heartbeat

---

Pour le fichier **haresources** aller dans **cd /etc/ha.d** créer un fichier à l'aide la commande **touch authkeys** et ensuite entrer dans le fichier à l'aide de la commande **nano haresources**

Mettez y dedans :

```
GNU nano 7.2  
SRV1-ThomasG IPaddr::192.168.206.51/24/ens18
```

Explication du fichier :

SRV1-ThomasG → Indique que ce nœud principal est responsable de l'adresse IP virtuelle.

IPaddr::192.168.206.51/24/ens18 → Configure une adresse IP flottante sur l'interface ens18

# Mise en place du service

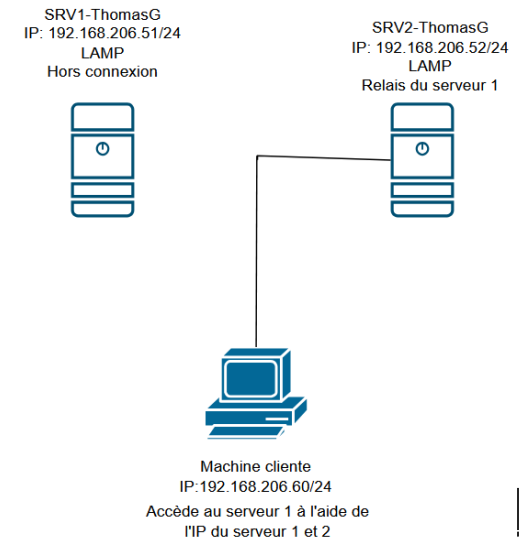
---

Maintenant démarré vos services sur vos deux serveurs en effectuant les commandes suivantes sur les deux serveurs :

- **service heartbeat stop**
- **service heartbeat start**

Ensuite sur votre serveur 2 il faut ping le serveur 1. Coupé ensuite le serveur 1 et vous allez observer que votre serveur 2 pourra toujours ping le serveur 1. **C'est normal car le serveur 2 a déjà pris le relais sans que vous vous en rendiez compte.**

```
root@SRV2-ThomasG:~# ping SRV1-ThomasG
PING SRV1-ThomasG (192.168.206.51) 56(84) bytes of data:
64 bytes from SRV1-ThomasG (192.168.206.51): icmp_seq=1 ttl=64 time=0.067 ms
64 bytes from SRV1-ThomasG (192.168.206.51): icmp_seq=2 ttl=64 time=0.090 ms
64 bytes from SRV1-ThomasG (192.168.206.51): icmp_seq=3 ttl=64 time=0.086 ms
64 bytes from SRV1-ThomasG (192.168.206.51): icmp_seq=4 ttl=64 time=0.083 ms
64 bytes from SRV1-ThomasG (192.168.206.51): icmp_seq=5 ttl=64 time=0.102 ms
64 bytes from SRV1-ThomasG (192.168.206.51): icmp_seq=6 ttl=64 time=0.096 ms
64 bytes from SRV1-ThomasG (192.168.206.51): icmp_seq=7 ttl=64 time=0.101 ms
64 bytes from SRV1-ThomasG (192.168.206.51): icmp_seq=8 ttl=64 time=0.086 ms
64 bytes from SRV1-ThomasG (192.168.206.51): icmp_seq=9 ttl=64 time=0.084 ms
```



## Mise en place du service

---

Sur votre machine cliente entrée l'adresse IP du serveur1, et vous allez vous rendre compte que le serveur 2 a belle et bien pris le relais du serveur 1

Vous pouvez faire le test inverse, et vous verrez que votre serveur 1 prend aussi le relais du serveur 2

**Résultat du serveur 1 coupé sur son adresse IP :**



**Résultat du serveur 2 coupé sur son adresse IP :**

