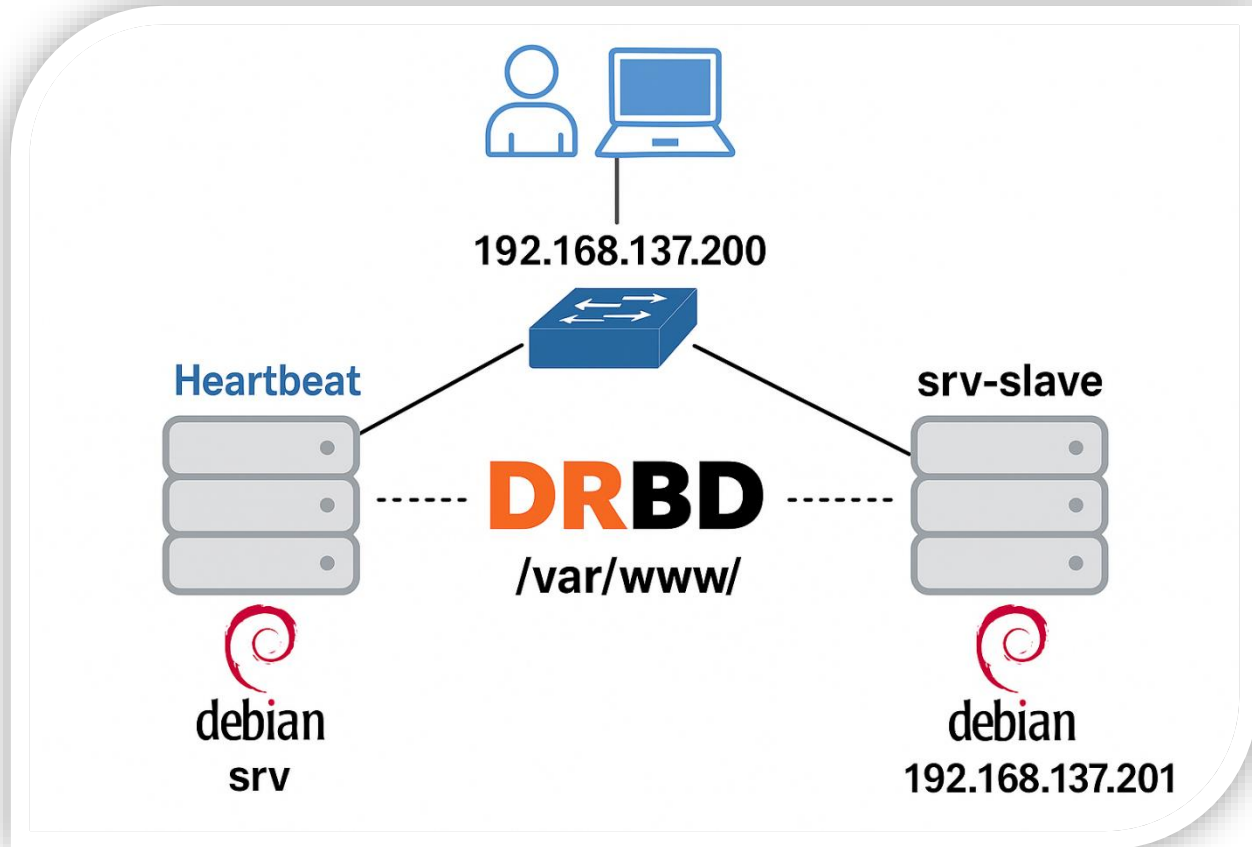
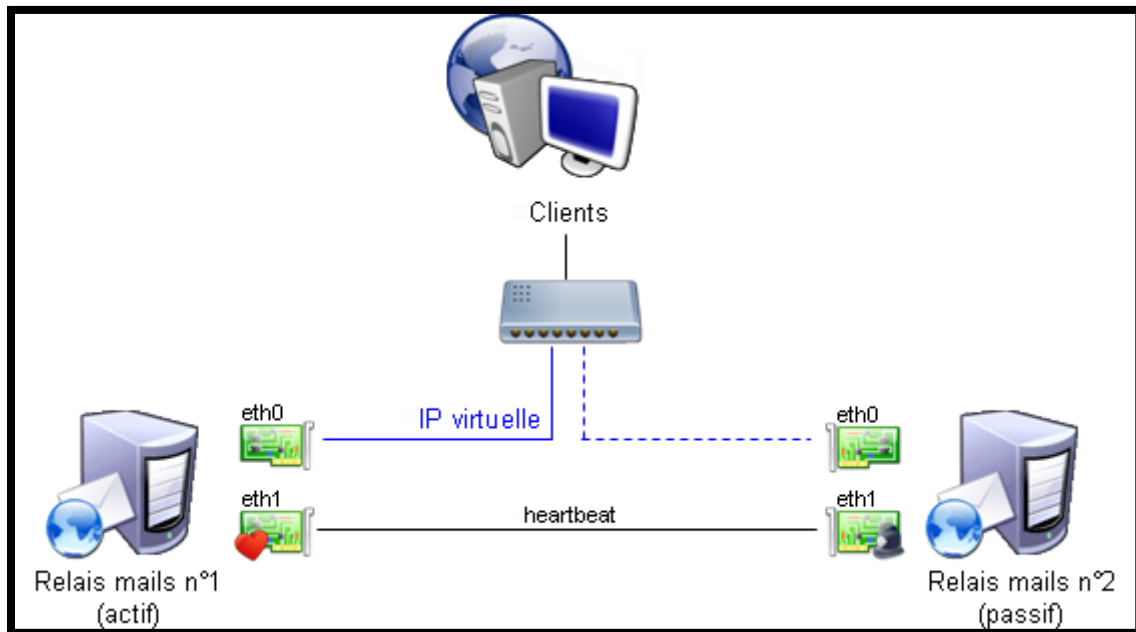


TP : Mise en place de la Haute Disponibilité logicielle avec Heartbeat



- Réalisé par : Chaimae Bouassab
- Professeur : Mr. GHADI Abderrahim
- Filière : Master Sécurité IT & Big Data
- Année universitaire : 2024-2025

- 1. Objectif du TP :** Le présent travail pratique a pour objectif la mise en œuvre d'un cluster de haute disponibilité en utilisant le logiciel Heartbeat sur deux machines Debian. Ce cluster permet de garantir la continuité de service en cas de panne d'un des nœuds. Le service web (Apache) sera automatiquement migré vers le serveur secondaire grâce à une IP virtuelle.



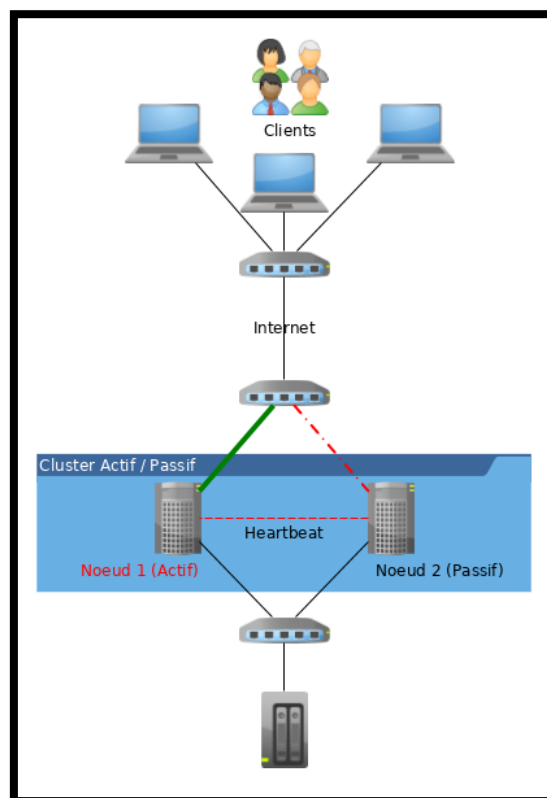
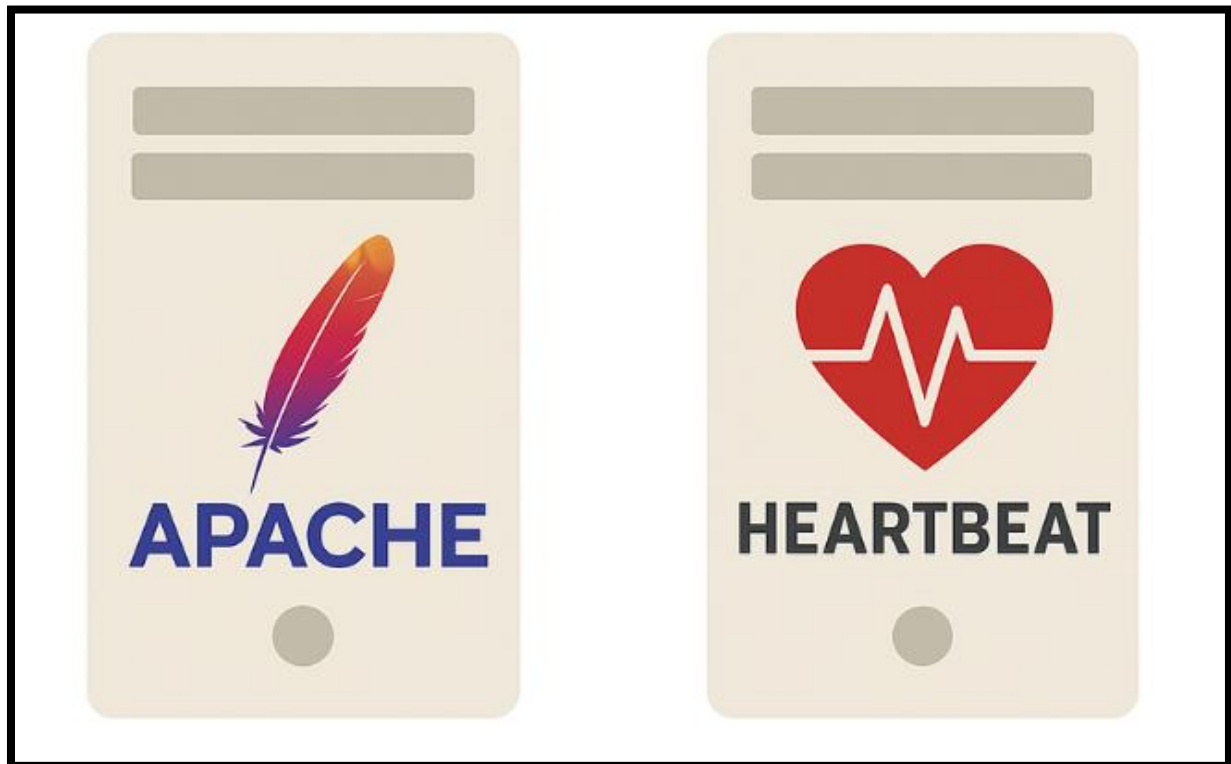
2. Configuration du matériel et réseau :

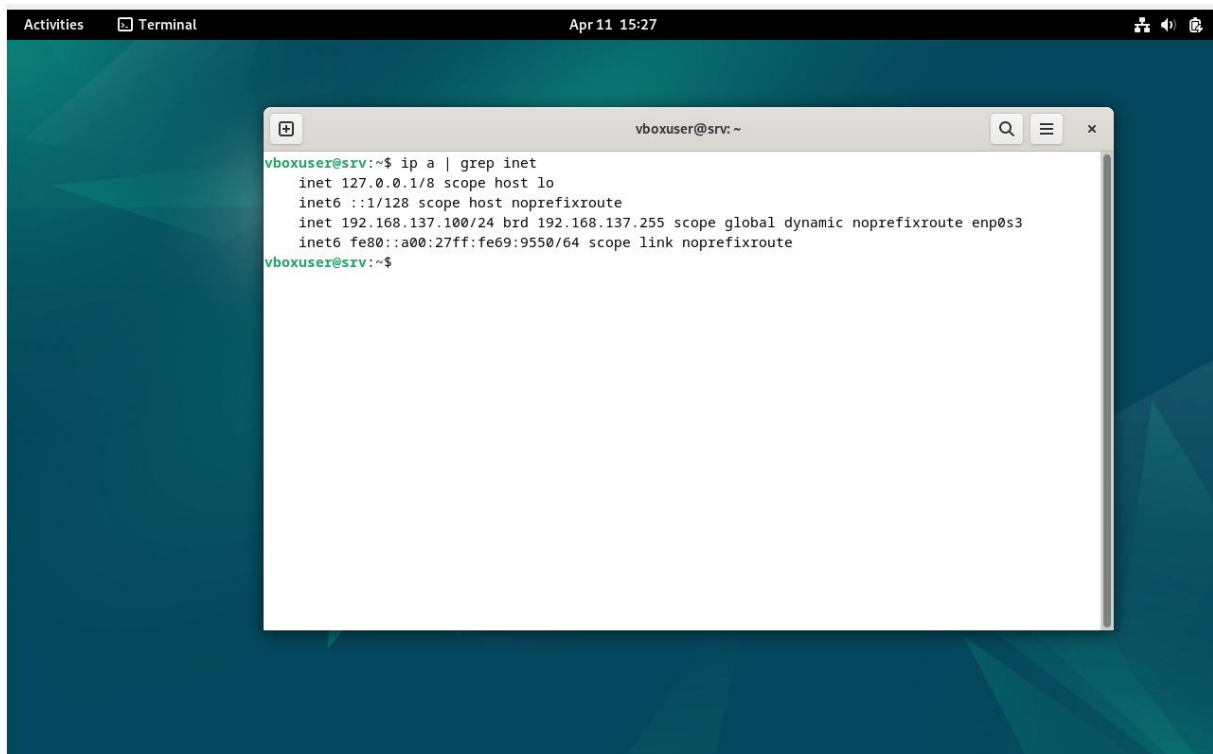
- Deux machines virtuelles sous Debian.
- Interfaces en mode « Bridged » pour permettre l'accès à l'IP virtuelle depuis le réseau local.
- IP de srv : 192.168.137.100
- IP de srv-slave : 192.168.137.63
- IP virtuelle : 192.168.137.200
- Interface réseau utilisée : enp0s3
- **3. Installation des paquets :** Sur les deux machines, installation d'Apache et Heartbeat :

```
sudo apt update
```

```
sudo apt install apache2 heartbeat -y
```

L'objectif est d'installer le serveur web à surveiller (Apache) et l'outil de gestion de la haute disponibilité (Heartbeat).



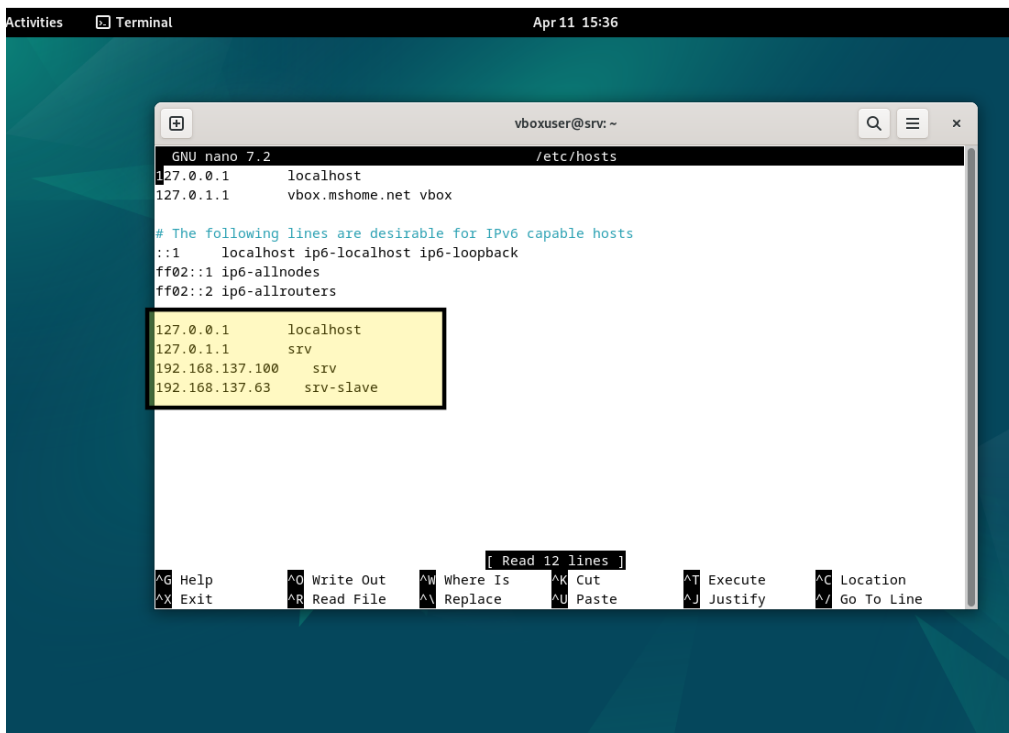


A terminal window titled 'vboxuser@srv: ~' showing the output of the command 'ip a | grep inet'. The output lists four network interfaces: 'lo' (127.0.0.1), 'eth0' (192.168.137.100), 'eth1' (192.168.137.255), and 'eth2' (fe80::a00:27ff:fe69:9550/64). The terminal is running on a system with a teal background.

```
vboxuser@srv:~$ ip a | grep inet
inet 127.0.0.1/8 scope host lo
inet6 ::1/128 scope host noprefixroute
inet 192.168.137.100/24 brd 192.168.137.255 scope global dynamic noprefixroute enp0s3
inet6 fe80::a00:27ff:fe69:9550/64 scope link noprefixroute
vboxuser@srv:~$
```

4. Configuration des hôtes (/etc/hosts) : Permet la résolution de noms entre les deux machines, indispensable pour le bon fonctionnement de Heartbeat.

Sur srv :



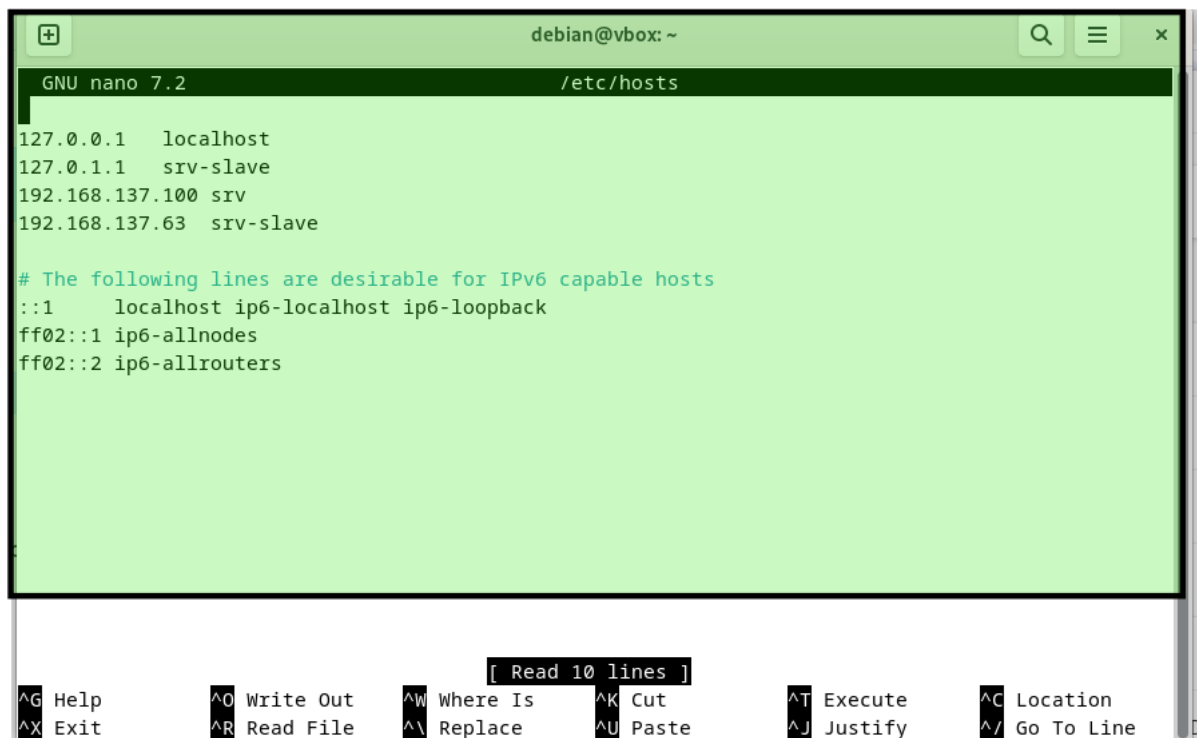
A terminal window titled 'vboxuser@srv: ~' showing the contents of the /etc/hosts file in the nano editor. The file contains entries for localhost, srv, and srv-slave. A yellow box highlights the last four lines of the file. The terminal is running on a system with a teal background.

```
GNU nano 7.2 /etc/hosts
127.0.0.1 localhost
127.0.1.1 vbox.mshome.net vbox

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

127.0.0.1 localhost
127.0.1.1 srv
192.168.137.100 srv
192.168.137.63 srv-slave
```

Sur srv-slave :



The screenshot shows a terminal window titled 'debian@vbox: ~'. Inside, the GNU nano 7.2 editor is open, editing the file /etc/hosts. The file content is as follows:

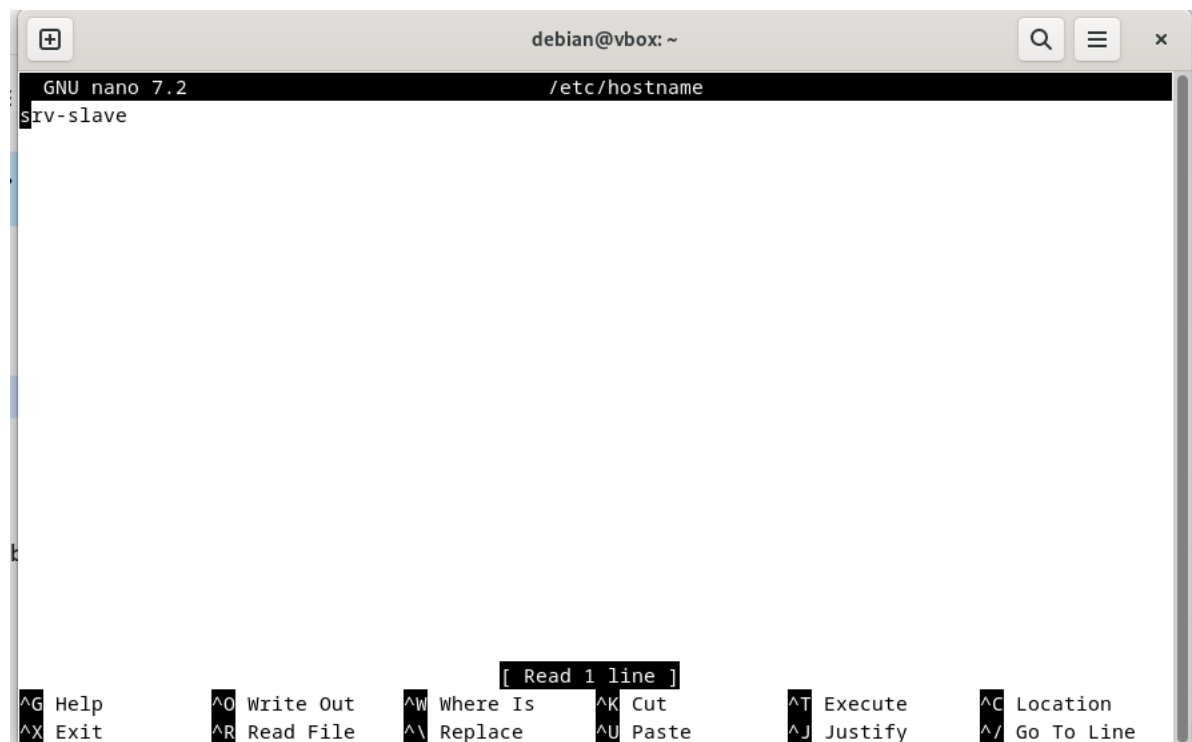
```
127.0.0.1    localhost
127.0.1.1    srv-slave
192.168.137.100 srv
192.168.137.63  srv-slave

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

At the bottom of the terminal, there is a status bar with various keyboard shortcuts: **[Read 10 lines]**, **^G Help**, **^O Write Out**, **^W Where Is**, **^K Cut**, **^T Execute**, **^C Location**, **^X Exit**, **^R Read File**, **^_ Replace**, **^U Paste**, **^J Justify**, and **^_ Go To Line**.

5. Configuration des noms d'hôtes (/etc/hostname) : Utilisé par Heartbeat pour identifier chaque nœud du cluster.

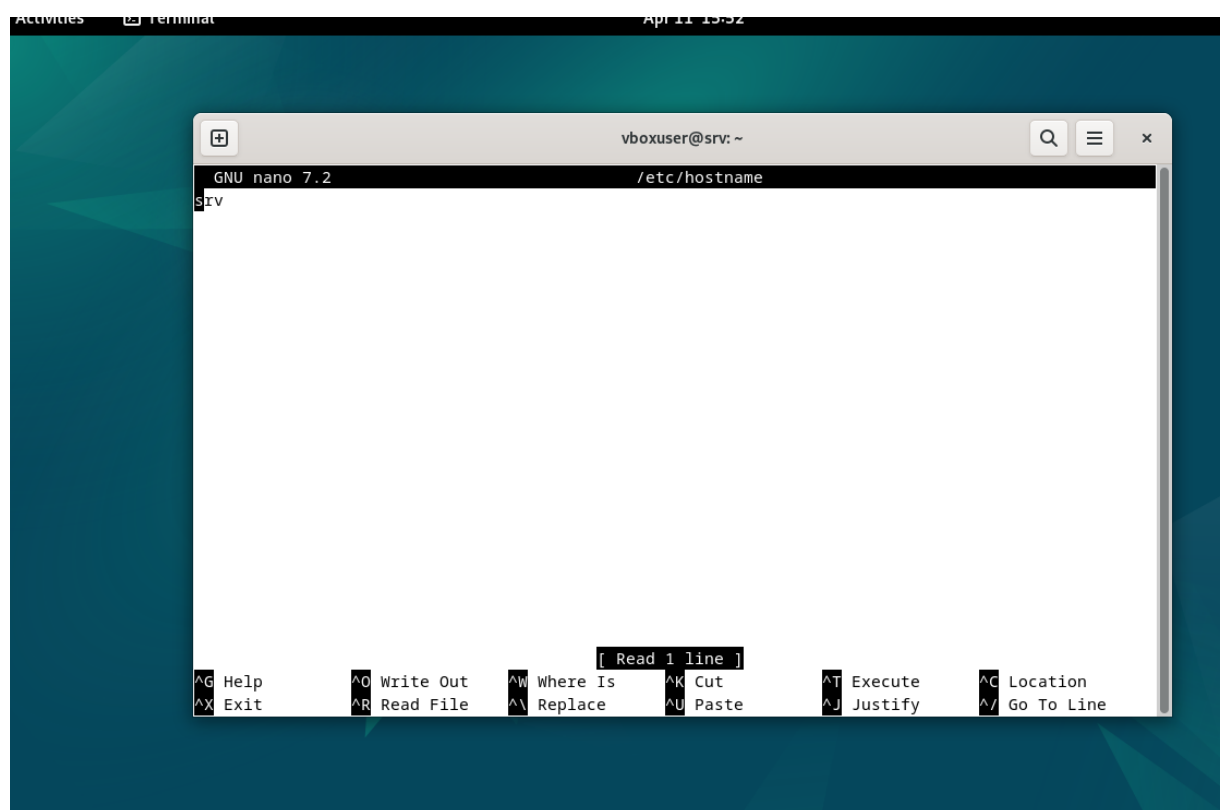
- srv contient : srv
- srv-slave contient : srv-slave
- Sur srv-slave :



A terminal window titled 'debian@vbox: ~' showing the GNU nano 7.2 editor editing the file /etc/hostname. The file content is 'srv-slave'. The bottom status bar shows '[Read 1 line]' and a list of keyboard shortcuts: ^G Help, ^O Write Out, ^W Where Is, ^K Cut, ^T Execute, ^C Location, ^X Exit, ^R Read File, ^\ Replace, ^U Paste, ^J Justify, and ^_ Go To Line.

```
GNU nano 7.2 /etc/hostname
srv-slave

[ Read 1 line ]
^G Help      ^O Write Out ^W Where Is  ^K Cut       ^T Execute   ^C Location
^X Exit      ^R Read File ^\ Replace   ^U Paste     ^J Justify   ^_ Go To Line
```

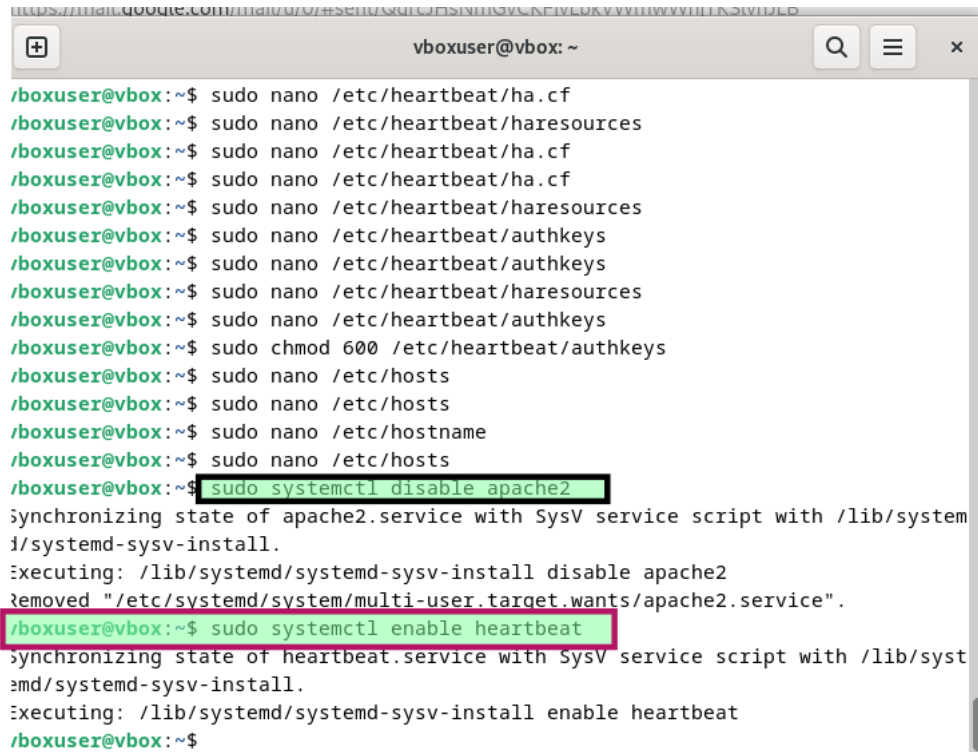


A terminal window titled 'vboxuser@srv: ~' showing the GNU nano 7.2 editor editing the file /etc/hostname. The file content is 'srv'. The bottom status bar shows '[Read 1 line]' and a list of keyboard shortcuts: ^G Help, ^O Write Out, ^W Where Is, ^K Cut, ^T Execute, ^C Location, ^X Exit, ^R Read File, ^\ Replace, ^U Paste, ^J Justify, and ^_ Go To Line.

```
GNU nano 7.2 /etc/hostname
srv

[ Read 1 line ]
^G Help      ^O Write Out ^W Where Is  ^K Cut       ^T Execute   ^C Location
^X Exit      ^R Read File ^\ Replace   ^U Paste     ^J Justify   ^_ Go To Line
```

6. Désactivation du démarrage automatique d'Apache : Pour éviter que le service Apache ne se lance sur les deux machines en même temps :



```

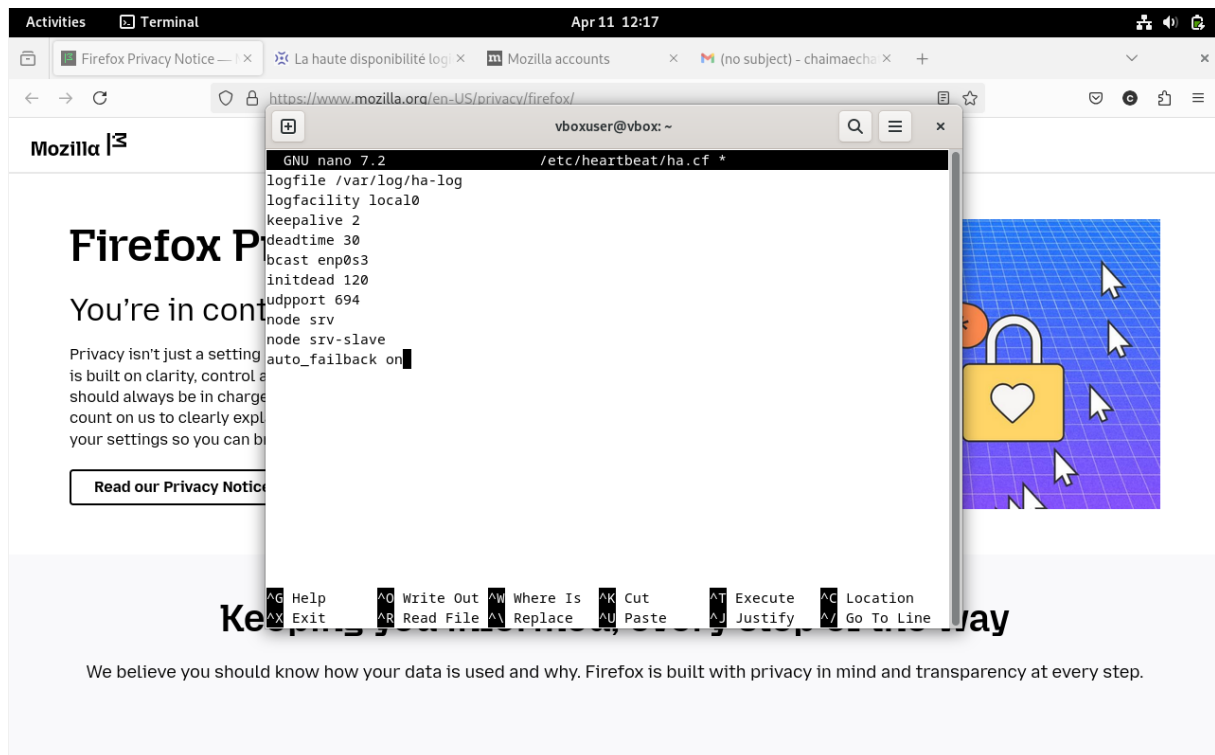
vboxuser@vbox: ~
/boxuser@vbox:~$ sudo nano /etc/heartbeat/ha.cf
/boxuser@vbox:~$ sudo nano /etc/heartbeat/haresources
/boxuser@vbox:~$ sudo nano /etc/heartbeat/ha.cf
/boxuser@vbox:~$ sudo nano /etc/heartbeat/haresources
/boxuser@vbox:~$ sudo nano /etc/heartbeat/authkeys
/boxuser@vbox:~$ sudo nano /etc/heartbeat/authkeys
/boxuser@vbox:~$ sudo nano /etc/heartbeat/haresources
/boxuser@vbox:~$ sudo nano /etc/heartbeat/authkeys
/boxuser@vbox:~$ sudo chmod 600 /etc/heartbeat/authkeys
/boxuser@vbox:~$ sudo nano /etc/hosts
/boxuser@vbox:~$ sudo nano /etc/hosts
/boxuser@vbox:~$ sudo nano /etc/hostname
/boxuser@vbox:~$ sudo nano /etc/hosts
/boxuser@vbox:~$ sudo systemctl disable apache2
synchronizing state of apache2.service with SysV service script with /lib/systemd/systemd-sysv-install.
Executing: /lib/systemd/systemd-sysv-install disable apache2
Removed "/etc/systemd/system/multi-user.target.wants/apache2.service".
/boxuser@vbox:~$ sudo systemctl enable heartbeat
synchronizing state of heartbeat.service with SysV service script with /lib/systemd/systemd-sysv-install.
Executing: /lib/systemd/systemd-sysv-install enable heartbeat
/boxuser@vbox:~$
```

sudo systemctl disable apache2

7. Création des fichiers de configuration Heartbeat :

Fichier /etc/heartbeat/ha.cf (sur les deux machines) :

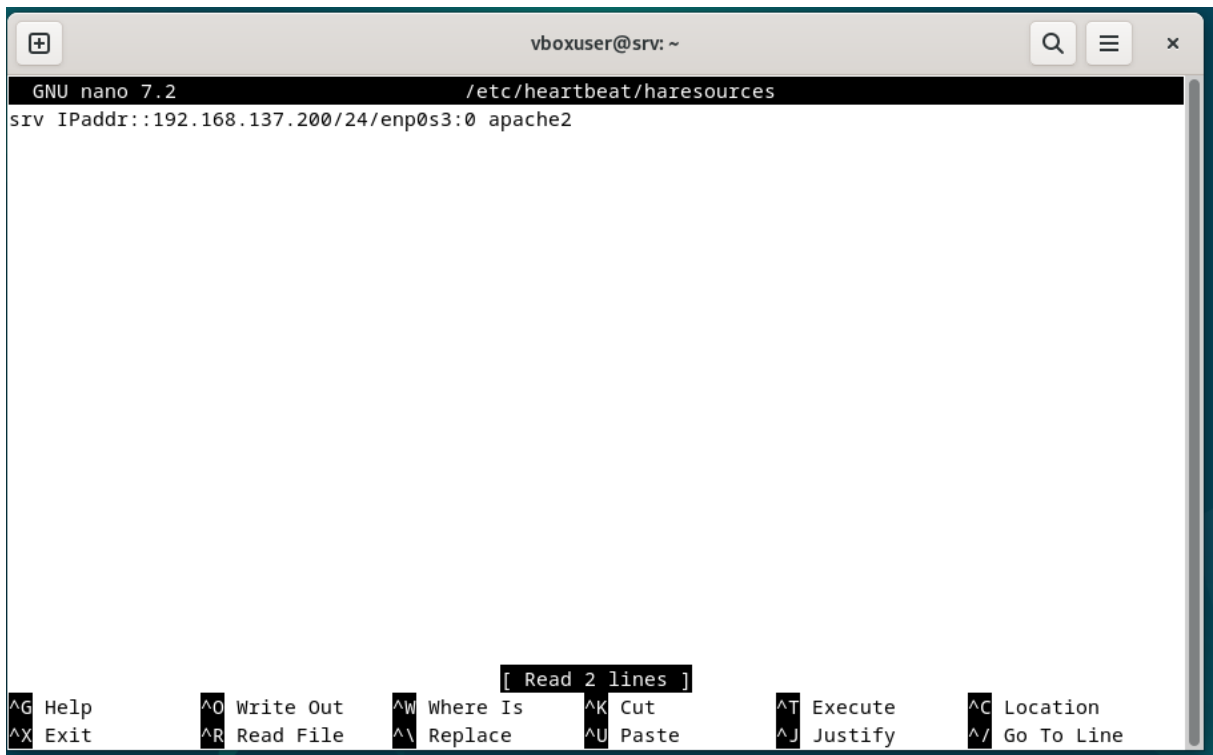
Nous mettons donc ce contenu dans le fichier sur les deux serveurs :



Ce fichier décrit les paramètres de communication et les nœuds impliqués dans le cluster.

Fichier /etc/heartbeat/haresources (sur les deux machines) :

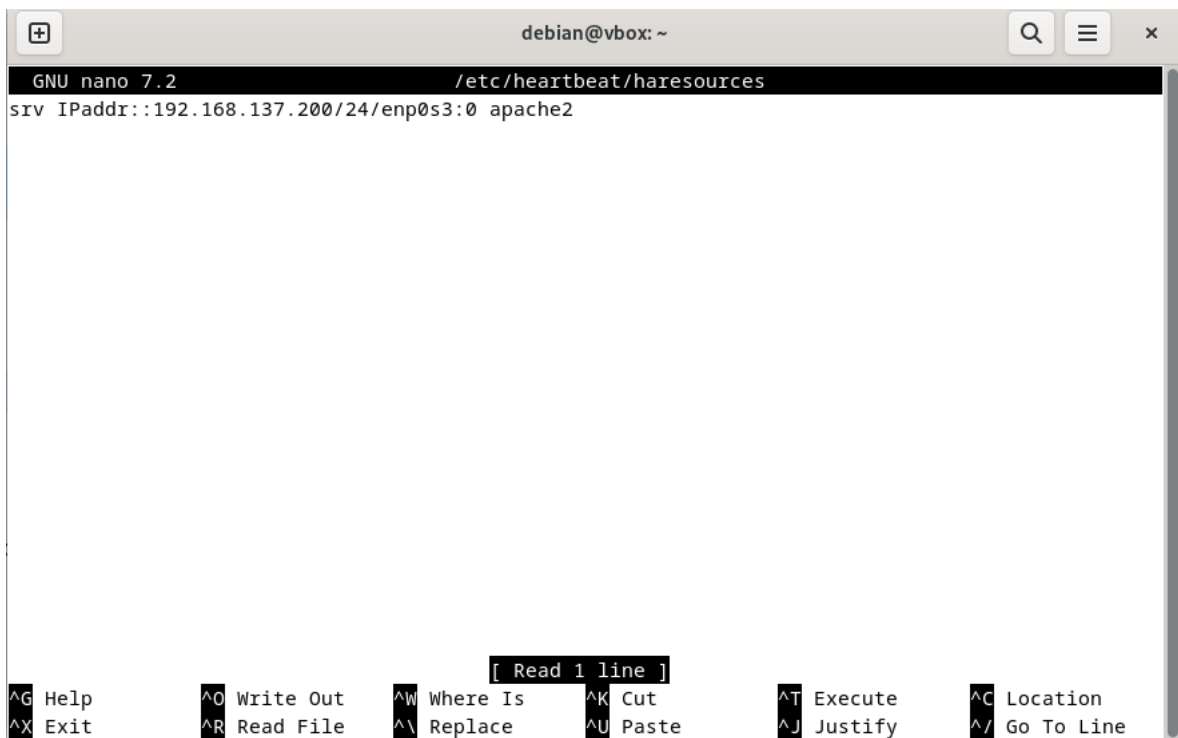
Sur srv



```
GNU nano 7.2 /etc/heartbeat/haresources
srv IPaddr::192.168.137.200/24/enp0s3:0 apache2

[ Read 2 lines ]
^G Help      ^O Write Out  ^W Where Is   ^K Cut        ^T Execute    ^C Location
^X Exit      ^R Read File  ^\ Replace    ^U Paste      ^J Justify    ^_ Go To Line
```

Sur srv-slave :



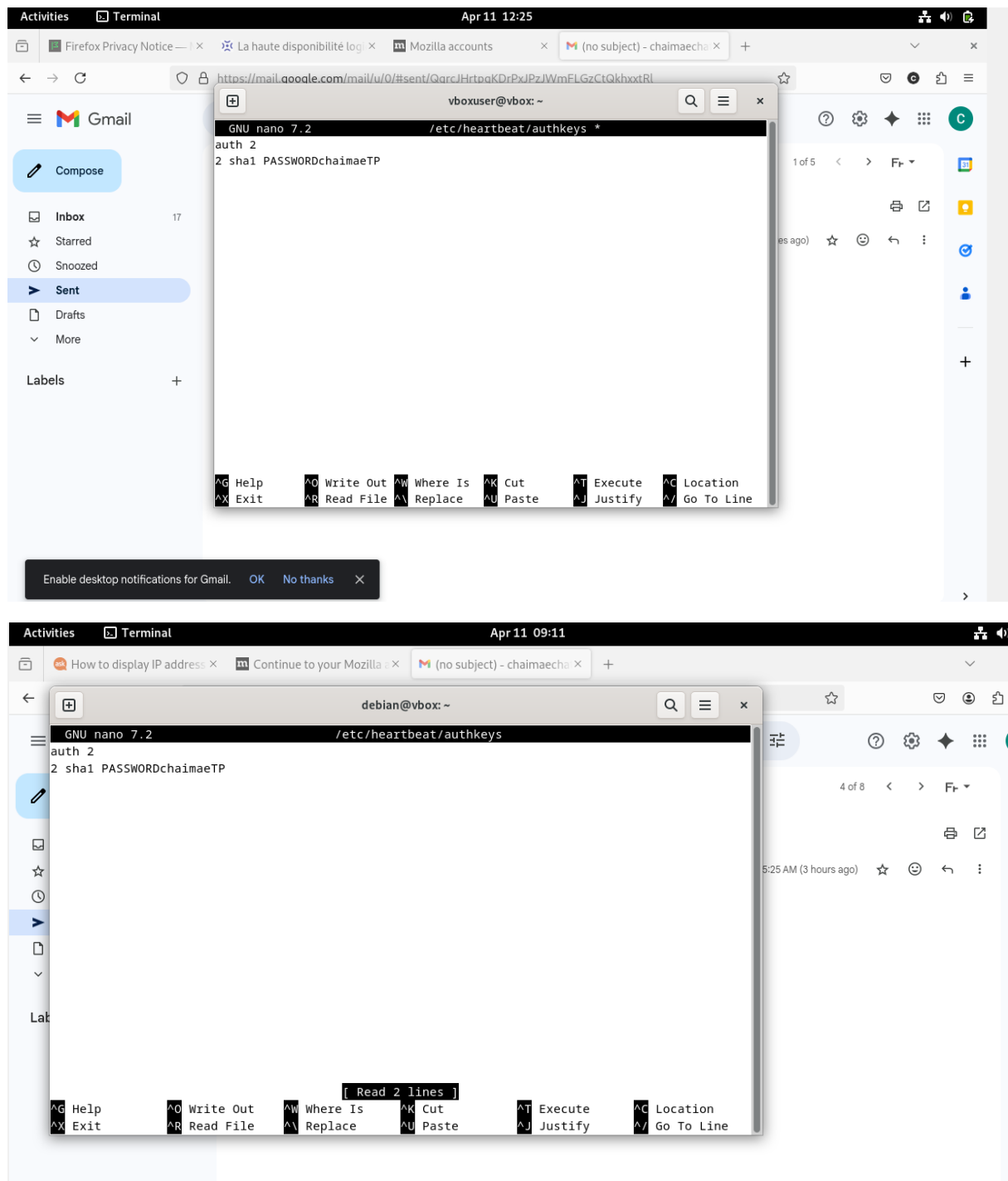
```
GNU nano 7.2 /etc/heartbeat/haresources
srv IPaddr::192.168.137.200/24/enp0s3:0 apache2

[ Read 1 line ]
^G Help      ^O Write Out  ^W Where Is   ^K Cut        ^T Execute    ^C Location
^X Exit      ^R Read File  ^\ Replace    ^U Paste      ^J Justify    ^_ Go To Line
```

Il indique que l'adresse IP virtuelle sera attachée au nœud `srv` par défaut, avec le service `apache2`.

Fichier `/etc/heartbeat/authkeys` (sur les deux machines) :

Nous passons maintenant au fichier `"/etc/heartbeat/authkeys"`, ce fichier va définir la clé qui permettra d'authentifier un minimum les serveurs membres du cluster entre eux

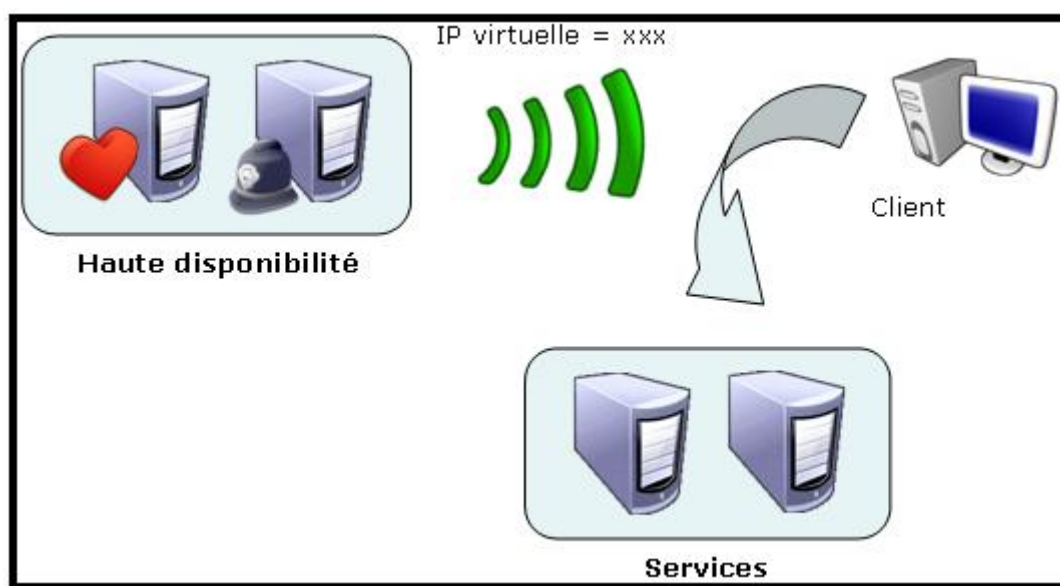
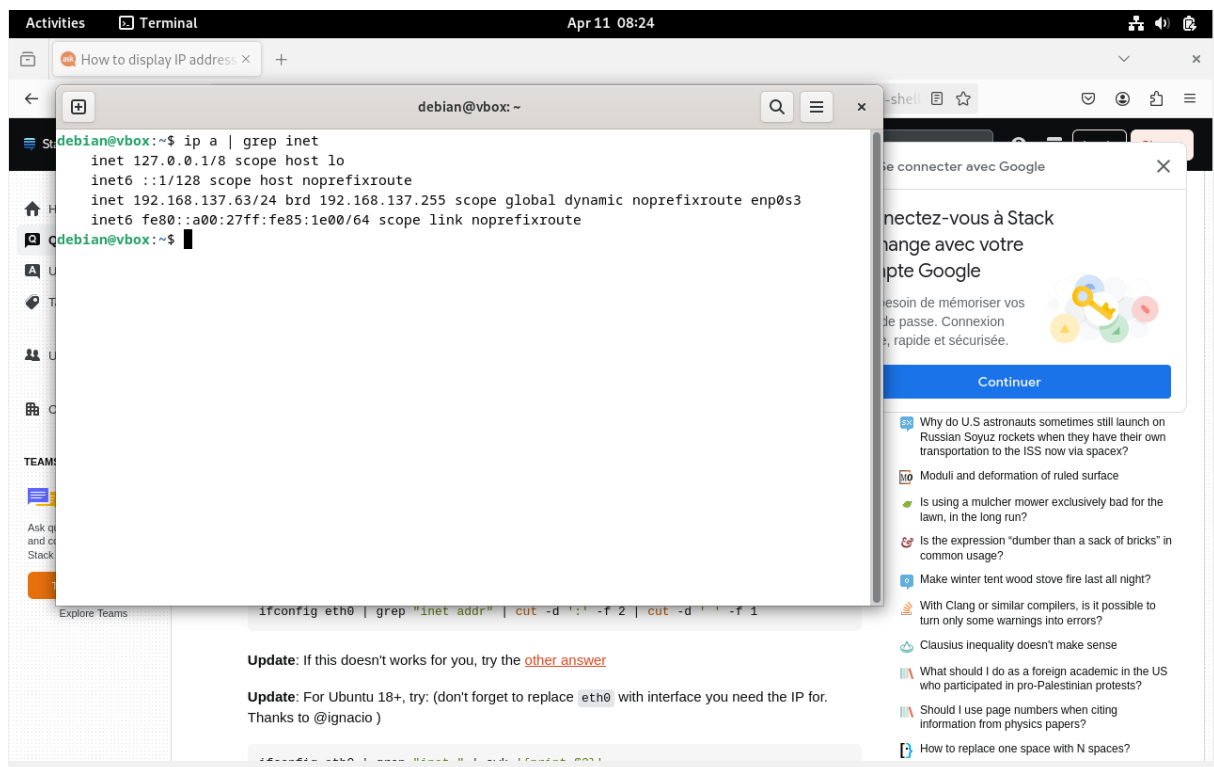


Ce fichier contient les clés d'authentification entre les nœuds.

chmod 600 /etc/heartbeat/authkeys

Par sécurité, on sécurise ce fichier en lui mettant des droits plus restreints :

```
root@srv-slave:/home/debian# sudo chmod 600 /etc/heartbeat/authkeys
root@srv-slave:/home/debian#
```



sudo chmod 600 /etc/heartbeat/authkeys pour les deux machines

```
vboxuser@srv:~$ sudo nano /etc/heartbeat/authkeys
vboxuser@srv:~$ sudo chmod 600 /etc/heartbeat/authkeys
vboxuser@srv:~$
```

```
root@vbox:/home/debian# sudo nano /etc/heartbeat/authkeys
root@vbox:/home/debian# sudo chmod 600 /etc/heartbeat/authkeys
```

8. Création de pages web différentes : Permet de vérifier visuellement quel serveur est actif.

Sur srv :

```
vboxuser@srv:~$ echo "Serveur Master" | sudo tee /var/www/html/index.html
Serveur Master
```

Sur srv-slave :

```
root@vbox:/home/debian# echo "Serveur Slave" | sudo tee /var/www/html/index.html
Serveur Slave
root@vbox:/home/debian#
```

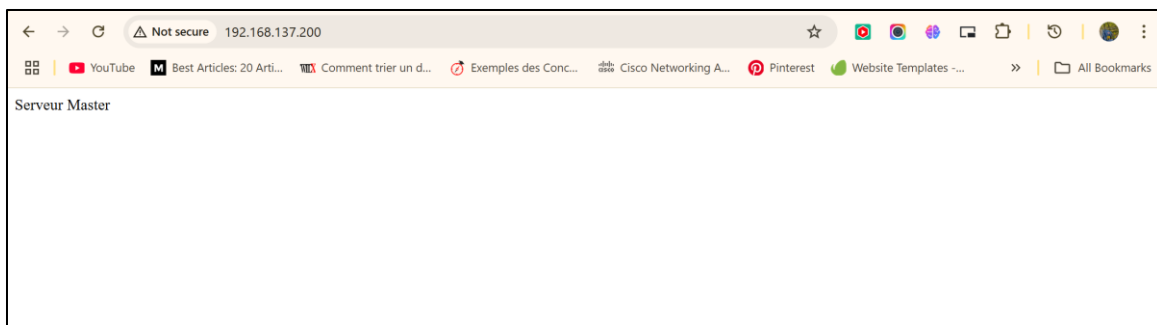
9. Lancement de Heartbeat : Sur les deux machines :

Sur srv :

```
vboxuser@srv:~$ sudo systemctl enable heartbeat
Synchronizing state of heartbeat.service with SysV service script with /lib/systemd/systemd-sysv-install.
Executing: /lib/systemd/systemd-sysv-install enable heartbeat
vboxuser@srv:~$ sudo systemctl start heartbeat
vboxuser@srv:~$
```

Sur srv-slave :

```
root@vbox:/home/debian# sudo systemctl enable heartbeat
Synchronizing state of heartbeat.service with SysV service script with /lib/systemd/systemd-sysv-install.
Executing: /lib/systemd/systemd-sysv-install enable heartbeat
root@vbox:/home/debian# sudo systemctl start heartbeat
root@vbox:/home/debian#
```



```
vboxuser@srv: ~  
• heartbeat.service - Heartbeat High Availability Cluster Communication and Membership  
  Loaded: loaded (/lib/systemd/system/heartbeat.service; enabled; preset: enabled)  
  Active: active (running) since Fri 2025-04-11 16:41:43 +01; 9s ago  
    Docs: man:heartbeat(8)  
          http://www.linux-ha.org/wiki/Documentation  
 Main PID: 3808 (heartbeat)  
   Tasks: 6 (limit: 2282)  
  Memory: 8.9M  
    CPU: 1.702s  
  CGroup: /system.slice/heartbeat.service  
          └─3808 "heartbeat: master control process"  
            └─3812 "heartbeat: FIFO reader"  
              └─3813 "heartbeat: write: bcast enp0s3"  
                └─3814 "heartbeat: read: bcast enp0s3"  
                  └─4036 /bin/sh /usr/lib/ocf/resource.d/heartbeat/IPaddr start  
                    └─4041 /usr/libexec/heartbeat/send_arp -i 500 -r 10 -p /run/resource-agents/send_arp>  
  
Apr 11 16:41:44 srv heartbeat[3808]: [3808]: info: Local Resource acquisition completed. (none)  
Apr 11 16:41:45 srv heartbeat[3808]: [3808]: info: srv-slave wants to go standby [foreign]  
Apr 11 16:41:46 srv heartbeat[3808]: [3808]: info: standby: acquire [foreign] resources from srv->  
Apr 11 16:41:46 srv heartbeat[3834]: [3834]: info: acquire local HA resources (standby).  
Apr 11 16:41:47 srv /usr/lib/ocf/resource.d/heartbeat/IPaddr(IPaddr_192.168.137.200)[3913]: INFO>  
Apr 11 16:41:47 srv IPaddr(IPaddr_192.168.137.200)[4024]: INFO: eval ifconfig enp0s3:0 192.168.137.200  
lines 1-23
```

10. Vérification de l'adresse IP virtuelle : La commande `ip a` sur la machine active doit montrer l'interface `enp0s3:0` avec l'IP `192.168.137.200`.

```
vboxuser@srv:~$ ip a  
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000  
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00  
    inet 127.0.0.1/8 scope host lo  
        valid_lft forever preferred_lft forever  
    inet6 ::1/128 scope host noprefixroute  
        valid_lft forever preferred_lft forever  
2: enp0s3: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default qlen 1000  
    link/ether 08:00:27:69:95:50 brd ff:ff:ff:ff:ff:ff  
    inet 192.168.137.100/24 brd 192.168.137.255 scope global dynamic noprefixroute enp0s3  
        valid_lft 600200sec preferred_lft 600200sec  
    inet 192.168.137.200/24 brd 192.168.137.255 scope global secondary enp0s3:0  
        valid_lft forever preferred_lft forever  
    inet6 fe80::a00:27ff:fe69:9550/64 scope link noprefixroute  
        valid_lft forever preferred_lft forever  
vboxuser@srv:~$
```

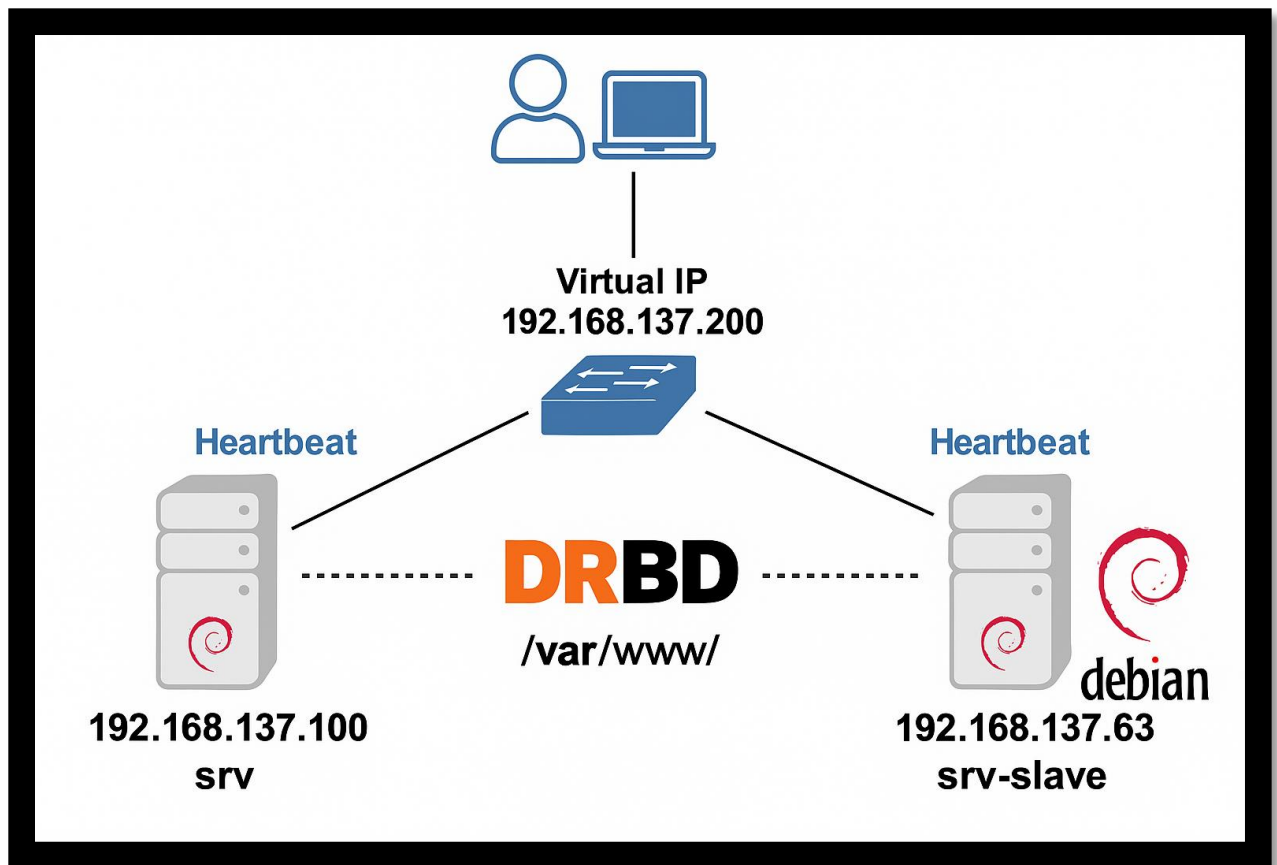
11. Test de bascule : Depuis un navigateur sur la machine hôte, on accède à `http://192.168.137.200` :

- Si le serveur `srv` est actif → affichage : "Serveur Master"
- Si on éteint `srv` → Heartbeat transfère l'IP virtuelle à `srv-slave`, affichage : "Serveur Slave"



12. Conclusion : Ce TP a permis de mettre en place une solution simple mais efficace de haute disponibilité avec Heartbeat. En cas de défaillance d'un serveur, l'autre prend automatiquement le relais, assurant ainsi la continuité du service web sans intervention humaine.

Ce système est fondamental dans des contextes critiques où l'interruption de service n'est pas tolérable (banques, hôpitaux, services en ligne, etc.).



Ce travail pratique a permis de comprendre et de mettre en œuvre un mécanisme essentiel dans les systèmes critiques : **la haute disponibilité logicielle**. Grâce à la solution Heartbeat, nous avons mis en place un cluster capable de **surveiller en temps réel l'état des nœuds** et de **transférer automatiquement le service Apache** en cas de défaillance de l'un d'eux. Ce processus garantit une **continuité de service transparente** pour l'utilisateur final.

L'expérience a également mis en évidence l'importance de la **configuration réseau**, de la **synchronisation entre les machines**, et de la **sécurisation des échanges**. Ce type de solution est couramment utilisé dans les domaines où l'interruption de service est inacceptable, comme les **systèmes bancaires**, les **hôpitaux**, les **plateformes e-commerce** ou les **services d'urgence**.

En conclusion, ce TP est un exemple concret d'**ingénierie de la résilience**, un pilier fondamental dans le domaine de l'administration des systèmes et des infrastructures informatiques modernes.