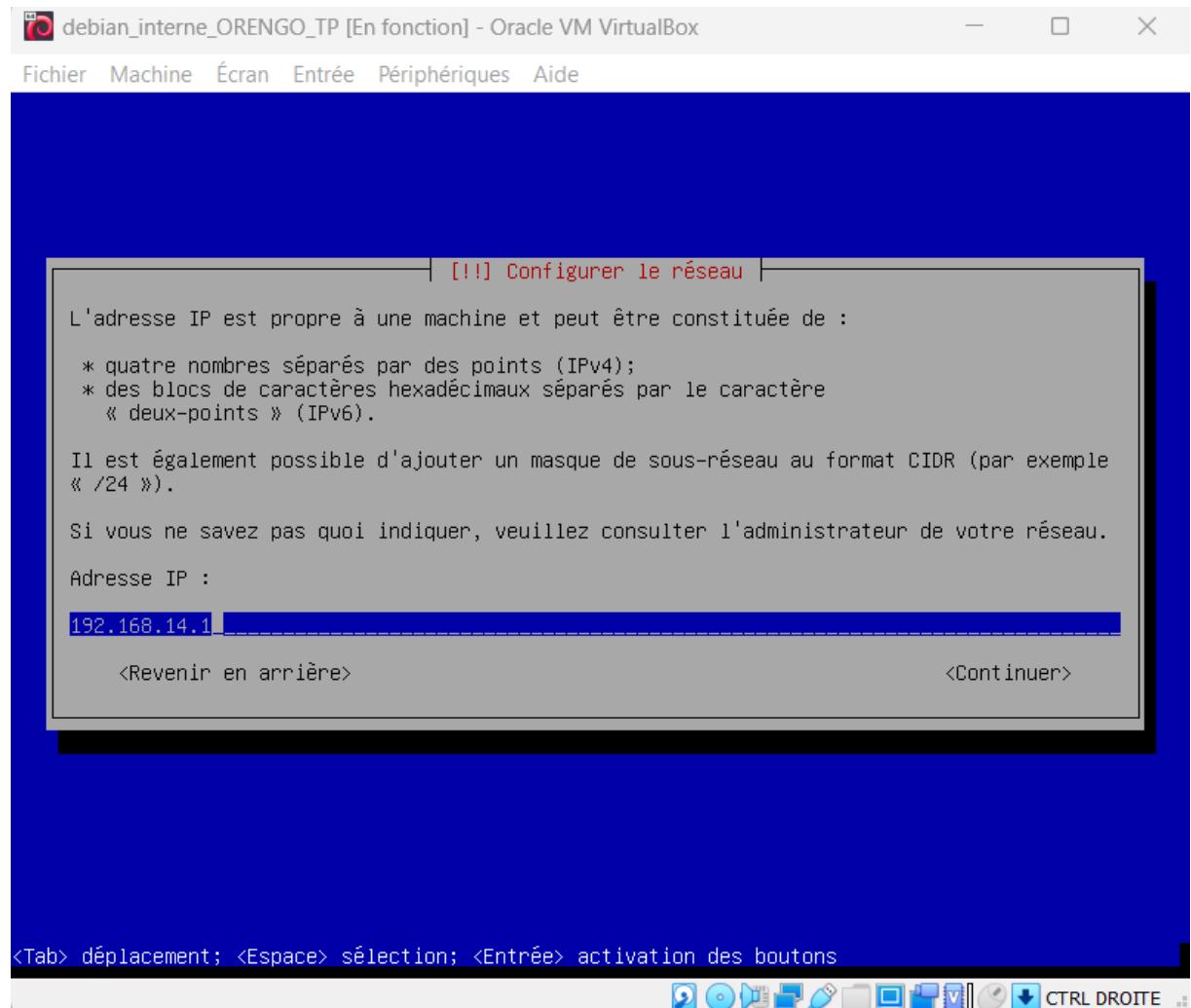
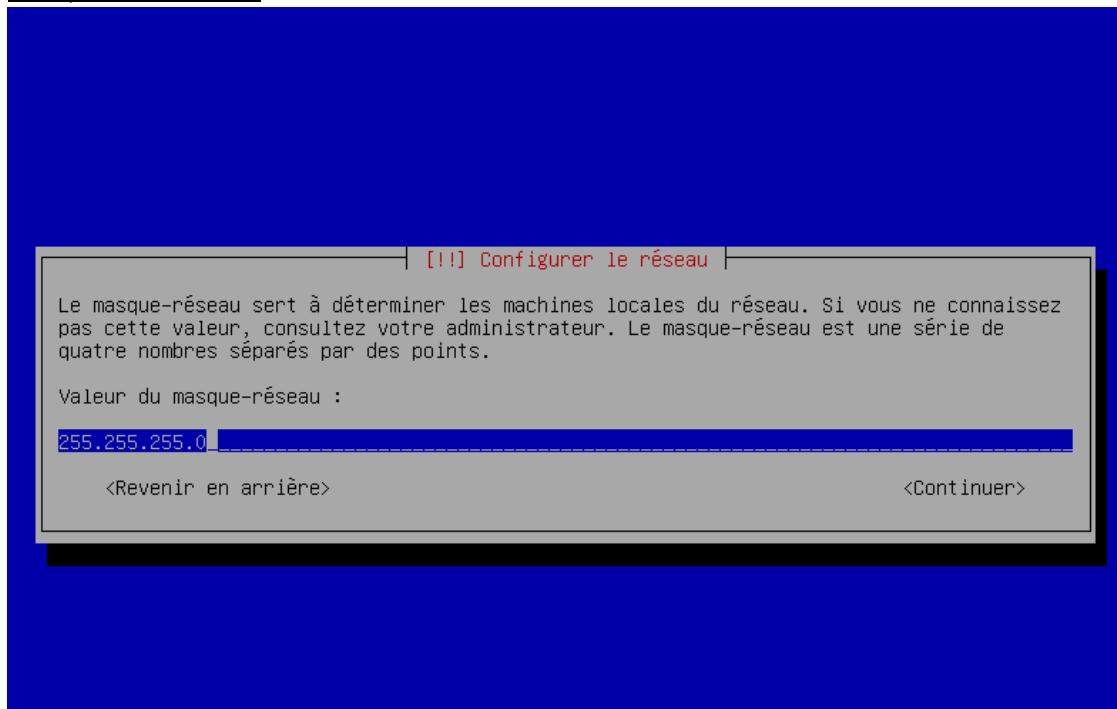


TP 2 serveur Web + machine avec bureau

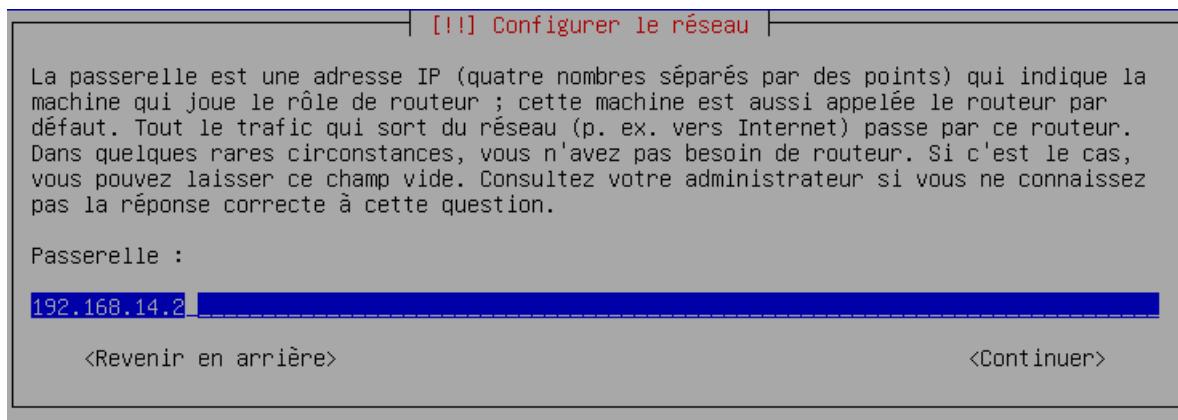
Paramétrage IP de la machine (serveur)



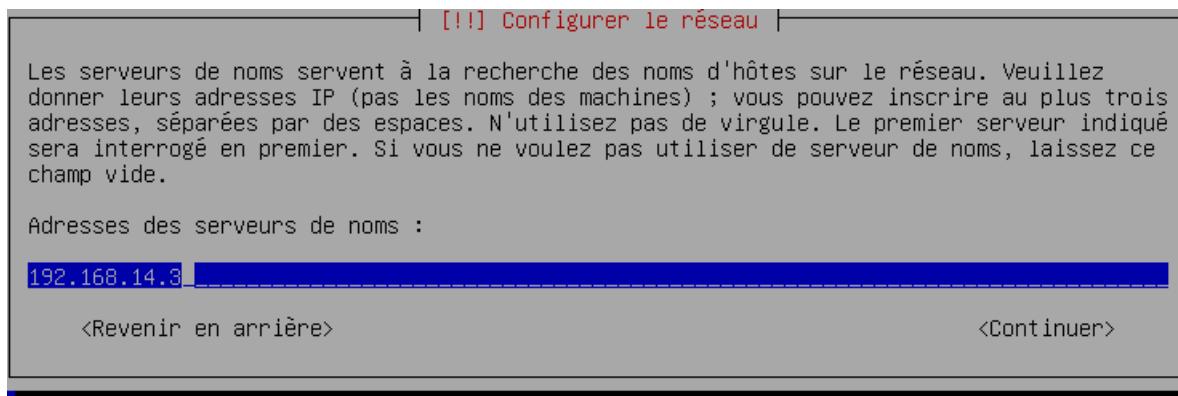
Masque sous -réseau



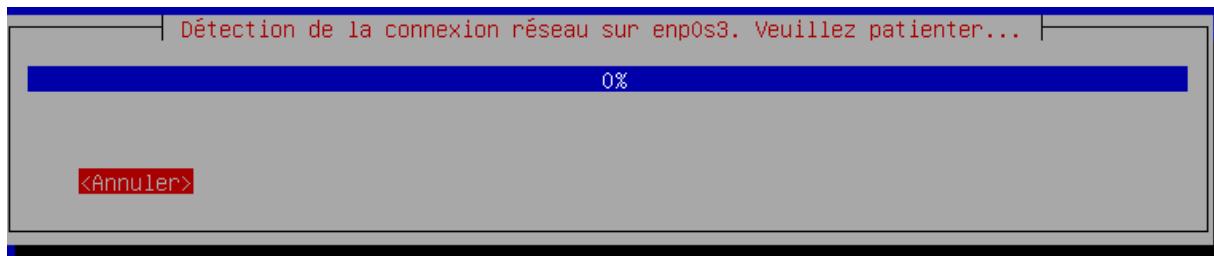
Passerelle



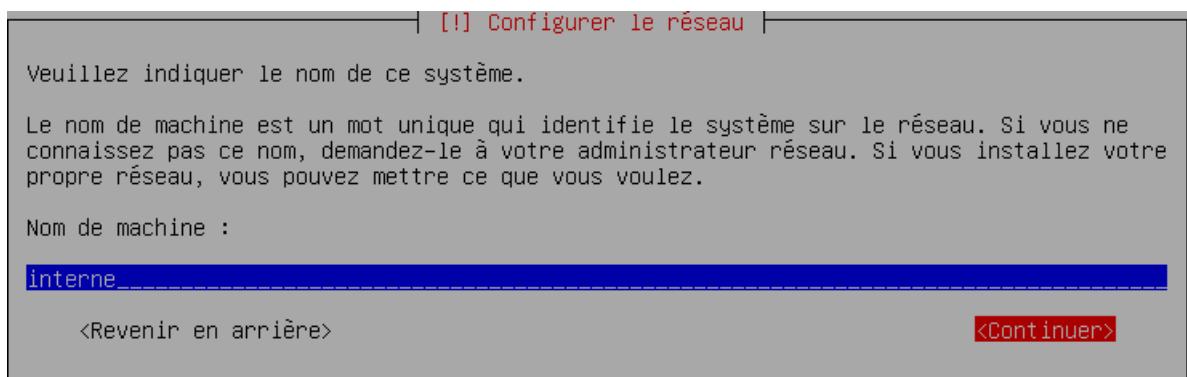
DNS



La config doit détecter la connexion réseau via la carte réseau de la machine.

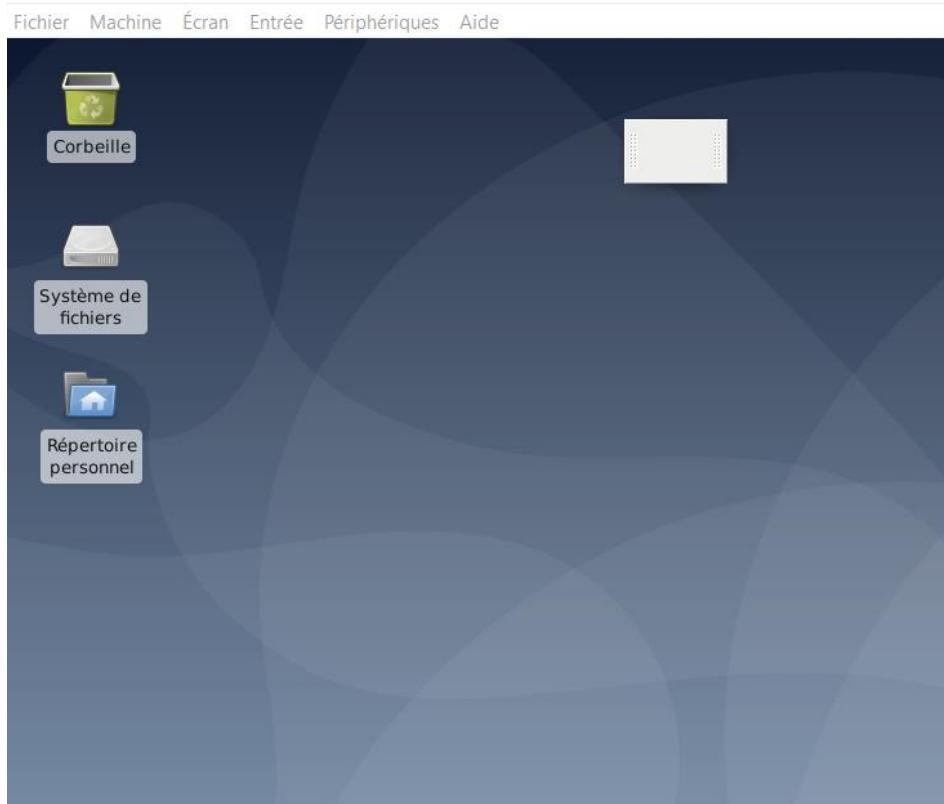


On appelle la machine « interne »

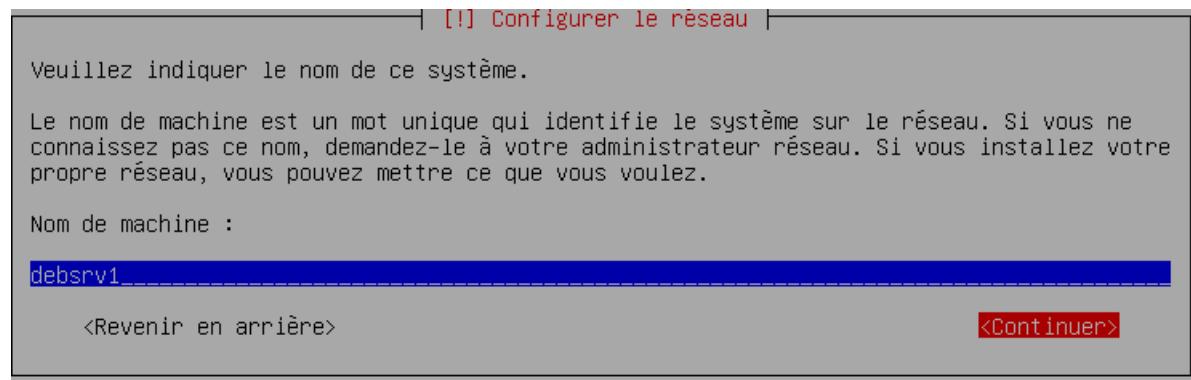


On fini l'installle puis on met bien le mode GRUB.

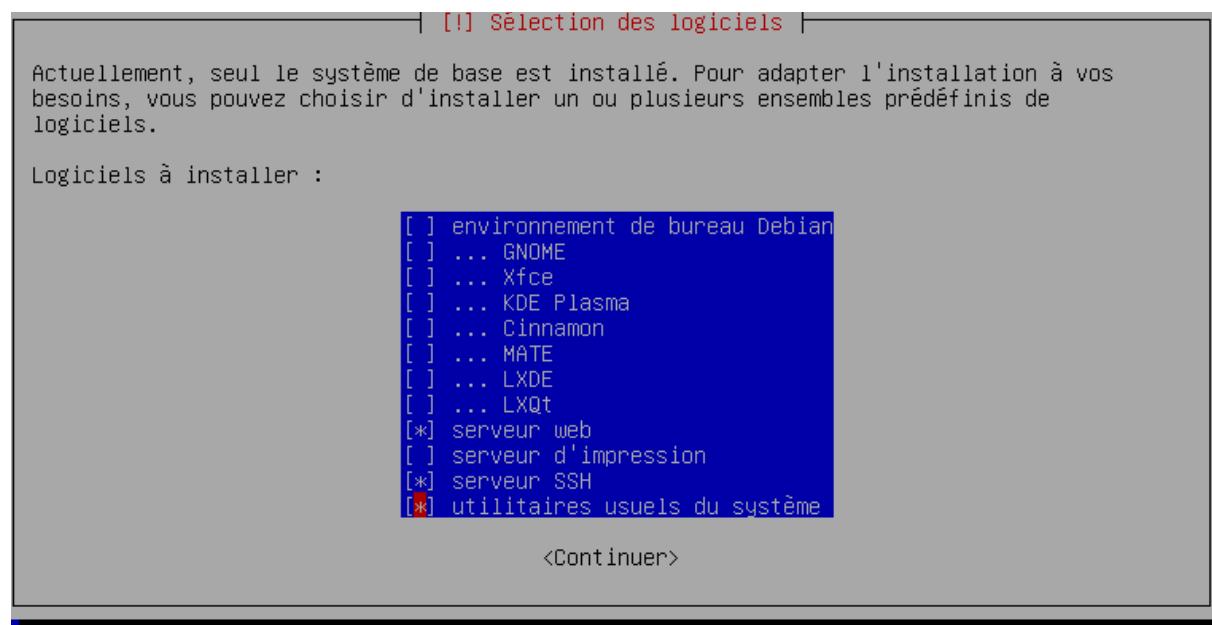
On se connecte ensuite en root avec le mdp choisi puis on accède au bureau de la première VM



Paramétrage de la seconde VM en Debian mais sans bureau. Elle sera un serveur Web. Nommé « debsrv1 ».



On paramètre notre machine en serveur Web, on lui met également le serveur SSH



On finit l'installle.

Paramétrage de l'IP fixe de ce serveur.

On tape la commande suivante : « nano /etc/network/interfaces »

```
# This file describes the network interfaces available on your system
# and how to activate them. For more information, see interfaces(5).

source /etc/network/interfaces.d/*

# The loopback network interface
auto lo
iface lo inet loopback

# The primary network interface
auto enp0s3
allow-hotplug enp0s3
iface enp0s3 inet static
address 10.0.2.212
netmask 255.255.255.0
gateway 10.0.2.2
```

« ctrl + X » on enregistre. Puis on ferme la VM pour faire une deuxième en serveur Web. Même procédure.

IP du second serveur :

```
# This file describes the network interfaces available on your system
# and how to activate them. For more information, see interfaces(5).

source /etc/network/interfaces.d/*

# The loopback network interface
auto lo
iface lo inet loopback

# The primary network interface
auto enp0s3
allow-hotplug enp0s3
iface enp0s3 inet static
address 10.0.2.215
netmask 255.255.255.0
gateway 10.0.2.2
```

Sur les 2 serveurs on installe aussi les applications suivante :

- Fail2ban
- Rkhunter
- Clamav

Configuration du cluster HA

- Modifier le fichiers hosts des deux serveurs :

```
• nano /etc/hosts
• 10.0.2.212      debsrv1
• 10.0.2.215      debsrv2
```

- Ouvrir les port UDP 5404 et 5405 en entrée et sortie :

```
iptables -I INPUT -m state --state NEW -p udp -m multiport --dports 5404,5405 -j ACCEPT
iptables -I OUTPUT -m state --state NEW -p udp -m multiport --sports 5404,5405 -j ACCEPT
```

- Générer la clé authentification pour la communication de corosync entre les deux serveurs :

```
corosync-keygen
```

- Copier le fichier authkey vers l'autre serveur :

```
scp /etc/corosync/authkey root@wan-slave:/etc/corosync/
```

- Faire une copie de la configuration de corosync :

```
mv /etc/corosync/corosync.conf /etc/corosync/corosync.back
```

- Créer la nouvelle configuration de corosync :

```
nano /etc/corosync/corosync.conf
logging {
    debug: off
    to_syslog: yes
}
nodelist {
    node {
        name: debsrv1
        nodeid: 1
        quorum_votes: 1
        ring0_addr: 10.0.2.212
    }
}
```

```
node {
    name: debsrv2
    nodeid: 2
    quorum_votes: 1
    ring0_addr: 10.0.2.215
}
}
quorum {
    provider: corosync_votequorum
}
totem {
    cluster_name: cluster-ha
    config_version: 3
    ip_version: ipv4
    secauth: on
    version: 2
    interface {
        bindnetaddr: 10.xx.xx.243
        ringnumber: 0
    }
}
```

- Copier le fichier corosync.conf vers l'autre serveur :

```
scp /etc/corosync/corosync.conf root@wan-slave:/etc/corosync/
```

- Désactiver deux fonctionnalités inutile pour notre cluster :

- **stonith** « shot the other node in the head » permet lorsqu'une machine n'est plus joignable d'être sur que cette machine soit bien hors ligne mais nécessite la gestion du fencing (gestion matériel avec par exemple *ipmi*).

```
crm configure property stonith-enabled=false
```

- **quorum** indique le nombre minimal de membres pour prendre une décision. Ce paramètre est utile pour des clusters à partir de trois machines.

```
crm configure property no-quorum-policy=ignore
```

- Démarrer les services corosync et pacemaker :

```
systemctl start corosync
systemctl start pacemaker
```

- Vérification de l'état du cluster :

```
crm status
```

```
root@debsrv1:~# crm status
Stack: corosync
Current DC: debsrv1 (version 2.0.1-9e909a5bdd) - partition with quorum
Last updated: Wed Apr  5 18:38:41 2023
Last change: Wed Apr  5 13:31:52 2023 by hacluster via crmd on debsrv1
2 nodes configured
0 resources configured

Inline: [ debsrv1 debsrv2 ]

0 resources
```

Configuration du failover d'IP

Première étape : attribution de l'adresse IP virtuelle

À minima, on crée une ressource nommée « *IPFailover* » (le nom est libre) qui va créer une adresse IP virtuelle 10.0.2.230/24 (d'autres options sont disponibles).

```
root@intralabXX:~# crm configure primitive IPFailover ocf:heartbeat:IPAddr2 params ip=10.0.2.230 cidr_netmask=24 nic=enp0s3 on monitor interval="30s" timeout="20s" iflabel=VIP
```

On fait à nouveau *crm status* et on voit que la connexion est ok.

```
root@debsrv2:~# crm status
Stack: corosync
Current DC: debsrv1 (version 2.0.1-9e909a5bdd) - partition with quorum
Last updated: Wed Apr  5 19:49:32 2023
Last change: Wed Apr  5 19:47:52 2023 by root via cibadmin on debsrv1

2 nodes configured
1 resource configured

Online: [ debsrv1 debsrv2 ]

Full list of resources:

IPFailover    (ocf::heartbeat:IPAddr2):     Started debsrv1
```

Deuxième étape : définir une préférence de nœud primaire

```
root@debsrv1:~# crm node standby
root@debsrv1:~# crm status
Stack: corosync
Current DC: debsrv1 (version 2.0.1-9e909a5bdd) - partition with quorum
Last updated: Wed Apr  5 20:04:21 2023
Last change: Wed Apr  5 20:04:16 2023 by root via crm_attribute on debsrv1

2 nodes configured
1 resource configured

Node debsrv1: standby
Online: [ debsrv2 ]

Full list of resources:

IPFailover    (ocf::heartbeat:IPAddr2):     Started debsrv2

root@debsrv1:~# crm node online
root@debsrv1:~# crm status
Stack: corosync
Current DC: debsrv1 (version 2.0.1-9e909a5bdd) - partition with quorum
Last updated: Wed Apr  5 20:04:59 2023
Last change: Wed Apr  5 20:04:55 2023 by root via crm_attribute on debsrv1

2 nodes configured
1 resource configured

Online: [ debsrv1 debsrv2 ]

Full list of resources:

IPFailover    (ocf::heartbeat:IPAddr2):     Started debsrv2

root@debsrv1:~#
```

On peut voir sur l'image ci-dessus que lorsque que je stop un serveur c'est le deuxième qui prend le relais.