

PETIT Antoine

Date de début : 02/01/2025

Date de fin : 02/01/2025

Haute Disponibilité avec Corosync & Pacemaker



Sommaire

1	CONTEXTE	3
2	OBJECTIFS	3
3	PRÉREQUIS	3
4	RÉALISATION	4
4.1	PRÉCONFIGURATION MACHINE MASTER	4
4.2	PRÉCONFIGURATION MACHINE SLAVE	4
4.3	CONFIGURATION COROSYNC MASTER	5
4.4	CONFIGURATION COROSYNC SLAVE	6
4.5	CONFIGURATION PACEMAKER MASTER	7

1 CONTEXTE

Le projet a pour objectif de mettre en place une solution de **serveur haute disponibilité (HA)** pour garantir la continuité de service et la tolérance aux pannes des applications critiques. La solution sera basée sur **Corosync** et **Pacemaker**, qui permettront la gestion des ressources et des services afin d'assurer la disponibilité des serveurs en cas de défaillance d'un nœud.

2 OBJECTIFS

Mettre en place un cluster haute disponibilité permettant de maintenir les services critiques en fonctionnement même en cas de panne d'un serveur.

3 PRÉREQUIS

Deux machines Debian 12 (Bookworm), chacune équipée de deux cartes réseau : une connectée à Internet et l'autre permettant l'interconnexion entre les deux machines.

Exemple :

IP Master	
INT 1	192.168.2.X
INT 2	192.168.56.2

IP Slave	
INT 1	192.168.2.X
INT 2	192.168.56.3

IP Haute Disponibilité	
INT 1	192.168.2.201

4 RÉALISATION

4.1 *PRÉCONFIGURATION MACHINE MASTER*

```
sudo apt update && sudo apt upgrade -y
mv /etc/hosts /etc/hosts.old
touch /etc/hosts
echo 192.168.56.2 master >> /etc/hosts
echo 192.168.56.3 slaive >> /etc/hosts
sudo apt install corosync pacemaker crmsh -y
sudo apt install apache2 -y
systemctl enable apache2 && systemctl start apache2
echo "This is the default page for master" | sudo tee /var/www/html/index.html
```

4.2 *PRÉCONFIGURATION MACHINE SLAVE*

```
sudo apt update && sudo apt upgrade -y
mv /etc/hosts /etc/hosts.old
touch /etc/hosts
echo 192.168.56.2 master >> /etc/hosts
echo 192.168.56.3 slaive >> /etc/hosts
sudo apt install corosync pacemaker crmsh -y
sudo apt install apache2 -y
systemctl enable apache2 && systemctl start apache2
echo "This is the default page for slaive" | sudo tee /var/www/html/index.html
```

```
corosync-keygen
scp /etc/corosync/authkey vagrant@slave:/home/vagrant/
mv /etc/corosync/corosync.conf /etc/corosync/corosync.back
nano /etc/corosync/corosync.conf
```

dans nano :

```
logging {
  debug: off
  to_syslog: yes
}

odelist {
  node {
    name: master
    nodeid: 1
    quorum_votes: 1
    ring0_addr: 192.168.56.2
  }

  node {
    name: slave
    nodeid: 2
    quorum_votes: 1
    ring0_addr: 192.168.56.3
  }
}
```

```
quorum {  
    provider: corosync_votequorum  
    two_node: 1  
}
```

```
totem {  
    cluster_name: cluster-ha  
    config_version: 3  
    ip_version: ipv4  
    secauth: on  
    version: 2  
    interface {  
        bindnetaddr: 192.168.56.0  
        ringnumber: 0  
    }  
}
```

/exit nano

```
scp /etc/corosync/corosync.conf vagrant@slave:/home/vagrant/  
systemctl start corosync && systemctl restart corosync  
systemctl start pacemaker && systemctl restart pacemaker
```

4.4 *CONFIGURATION COROSYNC SLAVE*

```
mv /home/vagrant/authkey /etc/corosync/authkey  
mv /etc/corosync/corosync.conf /etc/corosync/corosync.back  
mv /home/vagrant/corosync.conf /etc/corosync/corosync.conf  
systemctl start corosync && systemctl restart corosync  
systemctl start pacemaker && systemctl restart pacemaker
```

4.5 CONFIGURATION PACEMAKER MASTER

```
crm configure property stonith-enabled=false
crm configure property no-quorum-policy=ignore
systemctl start corosync && systemctl restart corosync
systemctl start pacemaker && systemctl restart pacemaker
crm configure primitive failover_ip ocf:heartbeat:IPaddr2 params ip=192.168.2.201
cidr_netmask=24 nic=eth1 op monitor interval=10s
crm configure location preferred-master failover_ip 100: master
```