

## SISR3 - Haute disponibilité d'un serveur Web

### 1 Documentation

#### 1.1 Première partie - Configuration du serveur maitre

##### 1. Qu'est-ce que la haute disponibilité ?

La haute disponibilité (ou High Availability ou HA) permet d'assurer et de garantir le bon fonctionnement des services ou applications proposées et ce 7j/7 et 24h/24.

##### 2. Pourquoi la mettre en place ?

Cela consiste donc à mettre en place toutes les actions et dispositions techniques pour qu'une infrastructure informatique soit toujours disponible en appliquant certains principes tels que la réplication des données, la sauvegarde, la répartition de la charge, la redondance, etc. pour limiter l'indisponibilité d'un SI.

##### 3. Qu'est-ce que "SLA" ?

Le service-level agreement (SLA) ou « entente de niveau de service » est un document qui définit la qualité de service, prestation prescrite entre un fournisseur de service et un client. Autrement dit, il s'agit de clauses basées sur un contrat définissant les objectifs précis attendus et le niveau de service que souhaite obtenir un client de la part du prestataire et fixe les responsabilités.

##### 4. En quoi cela peut concerner la haute disponibilité ?

Cela peut concerner la haute disponibilité car dans les deux cas il y a des responsabilités, il y a des règles et il doit assurer et garantir le bon fonctionnement des services.

##### 5. Qu'est-ce que la "réplication" ?

En informatique, la réplication est un processus de partage d'informations pour assurer la cohérence de données entre plusieurs sources de données redondantes, pour améliorer la fiabilité, la tolérance aux pannes, ou la disponibilité. On parle de réplication de données si les mêmes données sont dupliquées sur plusieurs périphériques.

##### 6. Qu'est-ce qu'un "cluster" ?

Le terme de cluster (grappe en français) fait référence à une batterie d'ordinateurs, tous interconnectés, dans le but de partager des ressources informatiques.

En d'autres termes, un cluster peut être :

-soit un ensemble d'ordinateurs connectés en réseaux et capable de traiter de concert un même travail. Par exemple le cumul de puissances des machines du cluster peut afficher un résultat inimaginable pour un particulier désirant faire du rendu d'images de synthèse à ses temps perdu.

-soit un ensemble d'ordinateurs connectés en réseaux dans le but de répartir des tâches sur chaque machine du clusters. Par exemple un cluster de tâches systèmes allouera à chaque machine du cluster un ensemble de processus à traiter suivant sa charge et sa puissance de calcul.

Les principales problématiques auxquelles répondent les clusters sont liées à la quête de puissance et à la diminution du temps d'exécution de tâches. De plus, le cluster forme un système informatique plus fiable car il représente un ensemble où le stockage des données est redondant sur chaque ordinateur. Ainsi, le cluster (quelqu'il soit) offre une meilleure qualité de service (QOS) aux utilisateurs.

Enfin, le cluster reste un système facilement administrable et modifiable. En effet, toutes les machines sont redondantes au niveau système et donc il suffira de créer un "master" système pouvant s'installer sur toute les machines. De plus, il suffira à un administrateur d'intégrer de nouvelles machines dans l'ensemble pour augmenter la puissance de son système.

##### 7. Qu'est-ce que Corosync ?

Corosync Cluster Engine est projet de logiciel Open source dérivé de l'OpenAIS2 project et sous licence Licence BSD modifiée. La mission de Corosync est de développer et de maintenir un cluster open source défini par la communauté.

## 1.2 Deuxième partie - Configuration des ressources

### 1. Qu'est-ce qu'une IP virtuelle ?

Une adresse IP virtuelle (virtual IP address, VIP ou VIPA) est une adresse IP non connectée à un ordinateur ou une carte réseau (NIC) spécifiques. Les paquets entrants sont envoyés à l'adresse IP virtuelle, mais en réalité ils circulent tous via des interfaces réseau réelles

### 2. Qu'est-ce qu'ip failover ?

Les IP failover vous permettent de bénéficier d'adresses IP basculables d'un serveur à un autre en quelques minutes, sur une simple action depuis la console de gestion Dedibox.

Elles permettent aussi d'avoir plusieurs IP sur le serveur et donc, par exemple, dans Plesk d'avoir une IP pour chaque site web.

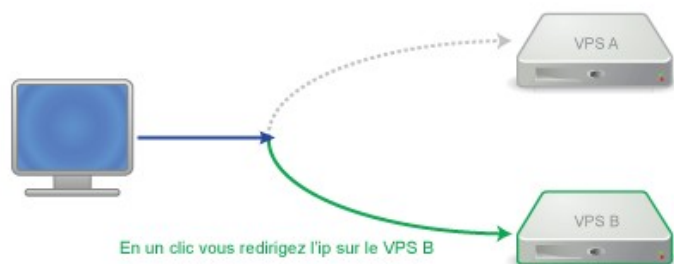
Ceci vous permet de pallier rapidement une panne de votre serveur, en basculant l'adresse IP failover vers un autre serveur de votre compte sans devoir mettre à jour et reconfigurer l'adresse IP de vos enregistrements DNS, sites internet et serveurs / clients de messagerie.

### 3. Expliquez le fonctionnement d'ip failover dans le contexte présent.

Avec vos serveur dédiés ou VPS vous pouvez disposer d'IP failover en fonction de l'offre.

A tout moment vous pouvez router une adresse vers n'importe quel serveur, dédié ou VPS. Grâce à notre console de gestion la bascule ce fait en un seul clic.

De cette façon vos services sont toujours disponibles et sans modification de DNS.



### 4. Qu'est-ce que Pacemaker ?

Pacemaker est un gestionnaire de cluster haute disponibilité. Il est chargé de démarrer, arrêter et superviser les ressources du cluster.

### 5. Qu'est-ce qu'une "ressource" au sens de Pacemaker ?

Pacemaker est un outil de gestion de ressources (une VIP, un service) pour vos clusters.

### 6. Qu'est-ce qu'un script OCF ?

### 7. Qu'est-ce qu'un CIB ?

Certains paramètres sont définis par les propriétés CIB (c'est-à-dire les attributs de la balise cib) plutôt que par le reste de la configuration du cluster dans la section de configuration.

La raison est simplement une question d'analyse. Ces options sont utilisées par la base de données de configuration qui, de par sa conception, ignore généralement le contenu qu'elle contient. La décision fut donc prise de les placer dans un endroit facile à trouver.

## 2 Travail préparatoire

Afin de mettre en place une solution de haute disponibilité, vous devez disposer des services suivants sur votre réseau :

- Un serveur DNS pour résoudre les noms d'hôtes.
- Un serveur LAP avec un vhost configuré, accessible via son adresse IP et son nom d'hôte.
- Un serveur de base de données fonctionnel.

## 3 Installation et configuration de la solution technique

### 3.1 Première partie - Configuration du serveur maitre

## 1. Installez sur votre serveur LAP Corosync et Pacemaker (1).

### (1) Ainsi que crmsh s'il ne fait pas partie des dépendances

Pour installer Corosync et Pacemaker ainsi que crmsh, il faut taper la commande sur les deux serveurs :

```
root@debian:/# apt-get install pacemaker corosync crmsh
```

## 2. Configurez Corosync.

Générer la clé de configuration pour la communication de corosync entre les deux serveurs en tapant cette commande :

```
root@debian:~# corosync-keygen
Corosync Cluster Engine Authentication key generator.
Gathering 1024 bits for key from /dev/random.
Press keys on your keyboard to generate entropy.
Press keys on your keyboard to generate entropy (bits = 488).
caribouPress keys on your keyboard to generate entropy (bits = 536).

Press keys on your keyboard to generate entropy (bits = 584).
Press keys on your keyboard to generate entropy (bits = 632).
Press keys on your keyboard to generate entropy (bits = 680).
Press keys on your keyboard to generate entropy (bits = 728).

dsPress keys on your keyboard to generate entropy (bits = 776).
fqdtgertgreqngPress keys on your keyboard to generate entropy (bits = 824).
trghththyertbgeryPress keys on your keyboard to generate entropy (bits = 872).
t(ytrgdftryutuhjdPress keys on your keyboard to generate entropy (bits = 920).
gthzrazerjffrjferPress keys on your keyboard to generate entropy (bits = 968).
verjgeiogjfiogeriojPress keys on your keyboard to generate entropy (bits = 1016).
vkr,eitjvtiiiWriting corosync key to /etc/corosync/authkey.
```

Après avoir fait cela, installer le ssh sur les serveurs et transférer la clé sur le deuxième serveur en tapant cette commande :

```
root@debian:/etc/corosync# scp /etc/corosync/authkey root@172.17.13.12:/etc/corosync/authkey
root@172.17.13.12's password:
authkey                               100% 128   155.6KB/s   00:00
root@debian:/etc/corosync#
```

Ensuite vous copiez le fichier de configuration corosync.conf en corosync.back afin d'avoir une copie du fichier en cas de problème en tapant cette commande sur les deux serveurs :

```
root@debian:/etc/corosync# cp corosync.conf corosync.back_
```

Tapez la commande `nano /etc/corosync/corosync.conf` et modifier le fichier comme ci-dessous

```
GNU nano 2.7.4 Fichier : corosync.conf
logging {
  debug: off
  to_syslog: yes
}
nodelist{
  node{
    name: wan-master
    nodeid: 1
    quorum_votes: 1
    ring0_addr: 172.17.13.11
  }
  node{
    name: wan-slave
    nodeid: 2
    quorum_votes: 1
    ring0_addr: 172.17.13.12
  }
}
quorum {
  provider: corosync_votequorum
}
totem{
  cluster_name: cluster-ha
  config_version: 3
  ip_version: ipv4
  secauth: on
  version: 2
  interface{
    bindnetaddr: 172.17.13.11
    ringnumber: 0
  }
}
```

Puis copier coller ce fichier de conf sur l'autre serveur en tapant cette commande :

```
root@debian:/etc/corosync# scp /etc/corosync/corosync.conf root@172.17.13.12:/etc/corosync/_
```

### 3. Démarrez Corosync.

Pour démarrer Corosync et Pacemaker en tapant ces commande :

```
root@debian:/etc/corosync# systemctl start corosync
```

```
root@debian:/etc/corosync# systemctl start pacemaker
```

Pour vérifier l'état du cluster taper la comme ci-dessous :

```
root@debian:/etc/corosync# crm status
Stack: corosync
Current DC: wan-master (version 1.1.16-94ff4df) - partition with quorum
Last updated: Thu Nov 21 11:27:04 2019
Last change: Thu Nov 21 11:25:08 2019 by hacluster via crmd on wan-master

3 nodes configured
0 resources configured

Node debian: UNCLEAN (offline)
Online: [ wan-master wan-slave ]

No resources
```

#### 4. Clonez le serveur LAP maître et configurez le serveur esclave (IP).

Si vous avez fait les memes commandes que ci-dessus vous avez donc clonez le serveur LAP maître donc il vous reste juste a configurer son ip en tapant cette commande :

```
nano /etc/network/interfaces
```

Puis modifier le fichier comme ci-dessous :

```
GNU nano 2.7.4 Fichier : 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
allow-hotplug ens32
iface ens32 inet static
address 172.17.13.12
netmask 255.255.255.0
gateway 172.17.13.254
```

#### 5. Désactivez Stonith et le Quorum

Taper ces deux commandes afin de désactivez Stonith et le Quorum et faire la même chose sur le deuxième serveurs

```
root@debian:/etc/corosync# crm configure property stonith-enablelf=false_
```

```
root@debian:/etc/corosync# crm configure property no-quorum-policy=ignore
```

### 3.2 Seconde partie - Configuration des ressources

#### 3.2.1 IP failover

##### 1. Créez la ressource permettant l'ip failover (²)

La ressource à été crée en même temps que j'ai fait la commande ci-dessous

(²) Dans la mesure du possible, il est conseillé d'utiliser les scripts OCF

##### 2. Créez un moniteur pour cette ressource avec :

— test de vie de la ressource de 10 secondes

— timeout de démarrage de 30 secondes

— timeout d'arrêt de 40 secondes.

Tapez la commande ci-dessous :

```
root@debian:/etc/corosync# crm configure primitive failoverip ocf:heartbeat:IPaddr2 params ip="172.17.13.12" cidr_netmask="24" op monitor interval="10s" timeout="30s" meta failure-timeout="40s"
```

### 3. Vérifiez que la ressource est activée sur le nœud primaire et que la préférence a été donnée à ce nœud.

Pour vérifier que la ressource est activée tapez la commande ci-dessous :

```
root@debian:/etc/corosync# crm status
Stack: corosync
Current DC: wan-slave (version 1.1.16-94ff4df) - partition with quorum
Last updated: Thu Nov 28 12:42:01 2019
Last change: Thu Nov 28 12:40:46 2019 by root via cibadmin on wan-master

3 nodes configured
1 resource configured

Online: [ wan-master wan-slave ]
OFFLINE: [ debian ]

Full list of resources:

failoverip      (ocf::heartbeat:IPaddr2):      Started wan-master
```

#### 3.2.2 Service web

— Créez la ressource "serviceWeb".

```
crm(live)configure# primitive serviceWeb ocf::heartbeat:apache params configfile="/etc/apache2/apache
2.conf" op monitor interval="30s" timeout="20s" op start interval="0" timeout="40s" op stop interval
="0" timeout="60s"
crm(live)configure# commit
crm(live)configure# bye
bye
```

Vérifier avec la commande de ci-dessous si le service web a bien été créé

```
root@debian:/etc/corosync# crm status
Stack: corosync
Current DC: wan-slave (version 1.1.16-94ff4df) - partition with quorum
Last updated: Tue Dec 10 10:16:38 2019
Last change: Tue Dec 10 10:15:25 2019 by root via cibadmin on wan-master

3 nodes configured
2 resources configured

Online: [ wan-master wan-slave ]
OFFLINE: [ debian ]

Full list of resources:

failoverip      (ocf::heartbeat:IPaddr2):      Started wan-slave
serviceWeb      (ocf::heartbeat:apache):       Started wan-master
```

— Groupez les ressources de manière à ce qu'elles soient actives sur le même nœud.

```
root@debian:/etc/corosync# crm configure
crm(live)configure# group HA failoverip serviceWeb
crm(live)configure# commit
crm(live)configure# quit
bye
```



Puis vérifier si les ressources sont bien grouper comme ci-dessous :

```
root@debian:/etc/corosync# crm status
Stack: corosync
Current DC: wan-slave (version 1.1.16-94ff4df) - partition with quorum
Last updated: Tue Dec 10 10:26:05 2019
Last change: Tue Dec 10 10:25:06 2019 by root via cibadmin on wan-master

3 nodes configured
2 resources configured

Online: [ wan-master wan-slave ]
OFFLINE: [ debian ]

Full list of resources:

Resource Group: HÁ
  failoverip (ocf::heartbeat:IPaddr2):      Started wan-slave
  serviceWeb (ocf::heartbeat:apache):      Started wan-slave
```

— Mettez en place un protocole de test pour votre solution.

#### 4 Tests

##### 4.1 Première partie - Configuration du maitre

###### 1. Le service web est accessible depuis son adresse IP ?

Oui, si jamais sa ne fonctionne pas vérifier le status de apache



###### 2. Le service web est accessible depuis son nom d'hôte ?

Oui



###### 3. Le service Corosync démarre et est opérationnel ?

Oui

```
root@debian:/var/www/html# service corosync status
● corosync.service - Corosync Cluster Engine
   Loaded: loaded (/lib/systemd/system/corosync.service; enabled; vendor preset: enabled)
   Active: active (running) since Thu 2019-11-28 12:24:54 CET; 1 weeks 4 days ago
     Docs: man:corosync
           man:corosync.conf
           man:corosync_overview
  Main PID: 400 (corosync)
    Tasks: 2 (limit: 4915)
   CGroup: /system.slice/corosync.service
           └─400 /usr/sbin/corosync -f
```

#### 4. Les deux serveurs sont pleinement opérationnels (Service web et Corosync) ?

Oui



```
root@debian:/etc/corosync# service corosync status
• corosync.service - Corosync Cluster Engine
  Loaded: loaded (/lib/systemd/system/corosync.service; enabled; vendor preset: enabled)
  Active: active (running) since Thu 2019-11-21 11:25:12 CET; 2 weeks 4 days ago
  Docs: man:corosync
        man:corosync.conf
        man:corosync_overview
  Main PID: 358 (corosync)
  Tasks: 2 (limit: 4915)
  CGroup: /system.slice/corosync.service
          └─358 /usr/sbin/corosync -f
```

#### 5. Vérifiez la configuration de Corosync.

On vérifie via la commande ci-dessous pour les deux serveurs :

```
root@debian:/etc/corosync# corosync-cfgtool -s
Printing ring status.
Local node ID 1
RING ID 0
      id      = 172.17.13.11
      status  = ring 0 active with no faults
```

```
root@debian:/etc/corosync# corosync-cfgtool -s
Printing ring status.
Local node ID 2
RING ID 0
      id      = 172.17.13.12
      status  = ring 0 active with no faults
```

#### 6. Après avoir désactivé Stonith et le Quorum vérifiez une dernière fois la configuration.

Le résultat est le même que ci-dessus

### 4.2 Deuxième partie - Configuration des ressources

#### 4.2.1 IP virtuelle

##### 1. Sur quel nœud la ressource a-t-elle été lancée ? Justifiez et vérifiez.

Sur le premier serveur

```
root@debian:/etc/corosync# crm status
Stack: corosync
Current DC: wan-slave (version 1.1.16-94ff4df) - partition with quorum
Last updated: Tue Dec 10 11:06:37 2019
Last change: Tue Dec 10 10:25:06 2019 by root via cibadmin on wan-master

3 nodes configured
2 resources configured

Online: [ wan-master wan-slave ]
Offline: [ debian ]

Full list of resources:

Resource Group: HA
  failoverip (ocf::heartbeat:IPaddr2):      Started wan-slave
  serviceWeb (ocf::heartbeat:apache):      Started wan-slave
```



2. Le service web est-il accessible à partir de l'adresse virtuelle ? Par son nom d'hôte ?

Oui il est accessible par son adresse virtuelle



3. Testez l'ip failover de deux manières (mise du nœud en stand-by et extinction du serveur). Y a-t-il une différence ?

Taper la commande ci-dessous dans le même répertoire :

```
crm(live)node# standby wan-slave
```

puis faites un crm status et vous devez obtenir comme ci-dessous :

```
root@debian:/# crm status
Stack: corosync
Current DC: wan-master (version 1.1.16-94ff4df) - partition with quorum
Last updated: Wed Dec 11 12:51:54 2019
Last change: Wed Dec 11 12:51:50 2019 by root via crm_attribute on wan-master

3 nodes configured
2 resources configured

Node wan-master: standby
Online: [ wan-slave ]
OFFLINE: [ debian ]

Full list of resources:

Resource Group: HA
  failoverip (ocf::heartbeat:IPaddr2):      Started wan-slave
  serviceWeb (ocf::heartbeat:apache):      Started wan-slave
```

Puis faites le test sur votre machine cliente :



### 4.2.2 Service web

1. Sur quel nœud la ressource a-t-elle été lancée ? Justifiez et vérifiez.

Voir 4.2.1

2. Vérifiez que le service HTTP soit bien démarré sur le serveur spécifié et non activé sur l'autre.

Sur le premier serveur apache est activé et sur le deuxième serveur apache est bien désactiver

```
root@debian:/etc/apache2/sites-available# service apache2 status
• apache2.service - The Apache HTTP Server
  Loaded: loaded (/lib/systemd/system/apache2.service; enabled; vendor preset: enabled)
  Active: inactive (dead) since Tue 2019-12-10 11:31:54 CET; 9min ago
  Process: 9357 ExecStop=/usr/sbin/apachectl stop (code=exited, status=0/SUCCESS)
  Process: 25403 ExecReload=/usr/sbin/apachectl graceful (code=exited, status=0/SUCCESS)
  Process: 9352 ExecStart=/usr/sbin/apachectl start (code=exited, status=0/SUCCESS)
  Main PID: 511 (code=exited, status=0/SUCCESS)
```

3. Testez votre solution grâce à votre protocole de test (³).

(³). Comment tuer votre service apache ?

<b><u>Test</u></b>	<b><u>Résultat attendu</u></b>	<b><u>Conforme</u></b>
Ip virtuelle	Accès au sites	✓
Noeud	Actif / ressources	✓
Bascule 1 > 2	Pas de perte de connexion	✓
Bascule 2 > 1	Pas de perte de connexion	✓