

# Cluster failover sous linux

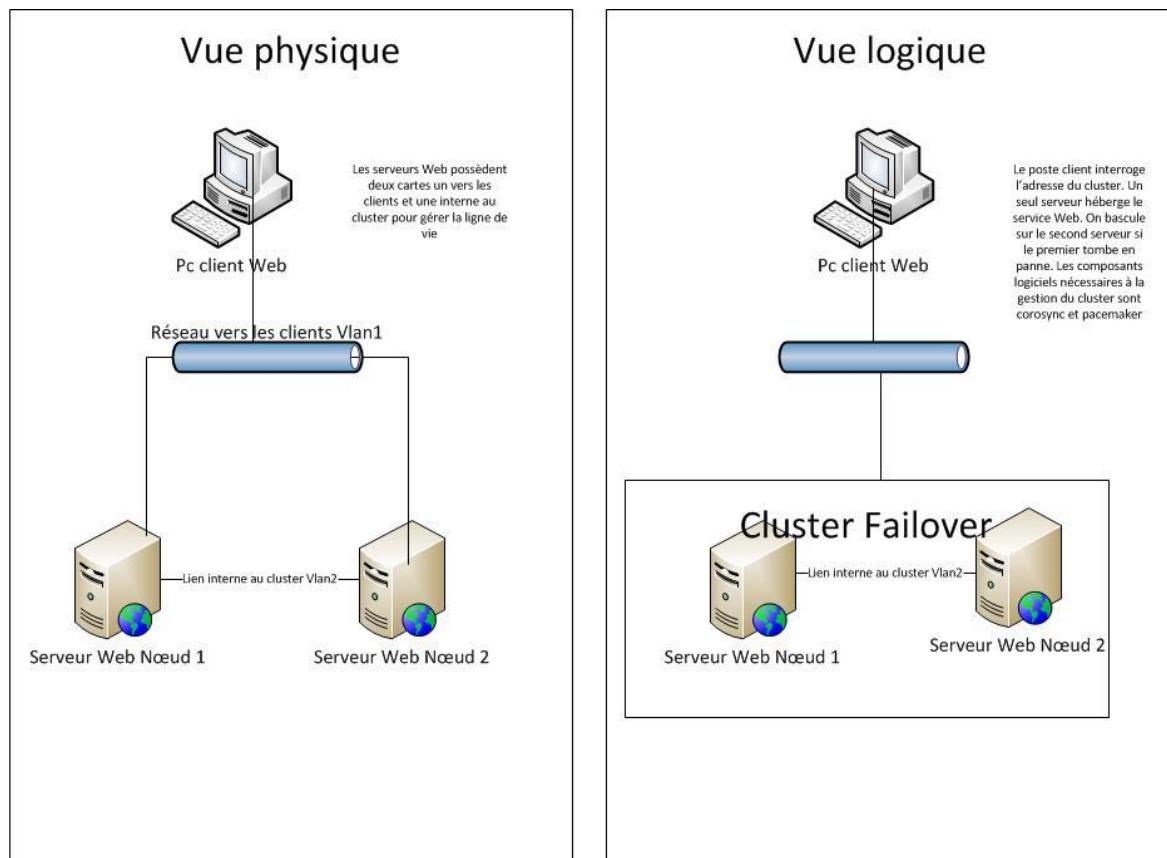
## I LA MISSION

Votre organisation a pour nouvel objectif de déployer en intranet un site statique. Ce nouveau site nécessite l'installation d'un serveur Web. Votre entreprise a choisi le SE linux et le serveur web Apache2. Vous êtes chargé de l'installation de ce serveur web. Votre préoccupation principale est d'assurer la continuité de service.

Après quelques recherches vous comprenez que vous pouvez installer deux serveurs web en cluster dit "failover". Les logiciels retenus pour la mise en place de ce cluster failover sont corosync et pacemaker.

## II LES LOGICIELS POUR CLUSTER FAILOVER SOUS LINUX

Corosync/pacemaker sont des outils permettant d'implémenter un cluster de type tolérance de panne (failover). Dans le principe on a 2 serveurs (ou plus) reliés à la fois au réseau local (vlan1) et par une connexion privée (vlan2). Les 2 serveurs ne forment qu'une seule machine mais seul un des deux fournit le service. Ce sont Corosync et pacemaker qui gèrent le basculement grâce à la ligne de vie (lien entre les machines du cluster) lorsqu'un serveur tombe, l'autre nœud prend le relais et cela reste invisible pour le client qui interroge une adresse de cluster et non pas l'adresse des nœuds du cluster.



### III PRÉ-REQUIS

Il vous faut deux machines virtuelles linux Lubuntu avec deux cartes réseaux. Une dans un premier vlan de la ferme (vlan1) et une dans un second vlan de la ferme (vlan2). Vous travaillez par groupe de 2 et chacun effectue les opérations sur sa machine.

- Pensez à changer le nom de vos machines.
- Seules les cartes réseau sur le vlan (vlan1) permettant de joindre les clients du cluster possèdent une adresse de passerelle. Le vlan accessible depuis les clients Seven du labo SISR doit être routé par le routeur de la section.
- Les secondes cartes sur le second vlan (vlan2) permettront les communications internes au cluster. Ce vlan n'a pas besoin d'être routé par le routeur de la section.
- Configurez vos interfaces et vérifiez la connectivité entre vos deux machines par l'intermédiaire de vos deux cartes réseau.
- Mettez à jour vos fichiers `/etc/hosts` pour que la résolution de noms fonctionne pour les deux machines de votre cluster.
- Installez Apache2 sur chaque machine et modifiez la page d'accueil web pour que l'on sache quel est le serveur qui la publie.
- Testez l'accès à vos pages web respectives.
- Faites un snapshot de vos machines pour pouvoir revenir à cet état ensuite.

### IV INSTALLATION ET CONFIGURATION DE COROSYNC

Corosync est une dépendance du paquet pacemaker qui est lui-même nécessaire. Il suffit donc d'installer pacemaker.

#### **apt-get install pacemaker**

Chaque nœud du cluster va s'authentifier auprès de l'autre nœud à l'aide d'une clé publique. Générez la clé d'authentification **sur un des deux nœuds** :

#### **corosync-keygen**

La clé est générée dans `/etc/corosync/authkey`. Copiez cette clé sur l'autre machine. Si vous avez un serveur ssh sur chaque machine (`apt-get install openssh-server`) vous pouvez utiliser scp (secure copy) pour la copier dans le même répertoire de l'autre machine (attention à la résolution de noms).

```
sudo scp /etc/corosync/authkey etudiant@Lubuntu2:/home/delmp/
```

Puis sur Lubuntu2

```
sudo cp /home/etudiant/authkey /etc/corosync/authkey
```

ou utilisez le client graphique gftp pour faire cette copie en précisant bien que vous utilisez le protocole SSH2.

Vérifiez ensuite les droits sur le fichier copié.

Maintenant il faut éditer le fichier `/etc/corosync/corosync.conf` pour renseigner l'adresse de réseau accessible depuis les clients (vlan1). Dans l'exemple ci-dessous cette adresse est 172.31.0.0.

```

interface {
    # The following values need to be set based on your environment
    ringnumber: 0
    bindnetaddr: 172.31.0.0
    mcastaddr: 226.94.1.1
    mcastport: 5405
}

```

Éditez ensuite le fichier `/etc/default/corosync` pour mettre le paramètre « START » à la valeur « yes ».

Démarrez le démon Corosync, puis démarrez le service pacemaker.

**service corosync start**

**service pacemaker start**

À ce stade les nœuds sont bien connectés entre eux mais le cluster n'est pas encore entièrement configuré. Vous pouvez le vérifier par la commande :

**crm\_mon -one-shot -V ou crm\_mon -l**

```

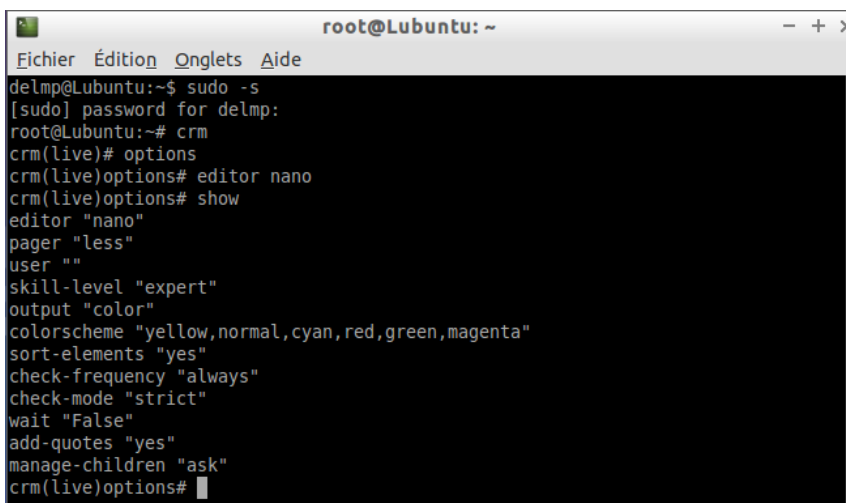
root@Lubuntu:~# crm_mon -l
Last updated: Wed Oct 21 22:28:45 2015
Last change: Wed Oct 21 22:28:34 2015 via crmd on Lubuntu1
Stack: corosync
Current DC: Lubuntu1 (740229136) - partition with quorum
Version: 1.1.10-42f2063
2 Nodes configured
0 Resources configured

Online: [ Lubuntu Lubuntu1 ]

```

## V PARAMÉTRAGE DU CLUSTER (À FAIRE SUR UN SEUL NŒUD)

Commencez par changer votre éditeur



```

root@Lubuntu: ~
Fichier Édition Onglets Aide
delmp@Lubuntu:~$ sudo -s
[sudo] password for delmp:
root@Lubuntu:~# crm
crm(live)# options
crm(live)options# editor nano
crm(live)options# show
editor "nano"
pager "less"
user ""
skill-level "expert"
output "color"
colorscheme "yellow,normal,cyan,red,green,magenta"
sort-elements "yes"
check-frequency "always"
check-mode "strict"
wait "False"
add-quotes "yes"
manage-children "ask"
crm(live)options#

```

Sortez par « end ». Il faut maintenant attribuer une adresse IP virtuelle au cluster par la commande crm (Cluster Resource Manager) en créant une nouvelle configuration (config) pour le cluster. Commencez par taper la commande « configure » :

```
crm(live)options# end
crm(live)# cib new config
INFO: config shadow CIB created
crm(config)#
```

Assignez une adresse IP virtuelle au cluster.

```
crm(config)# configure
crm(config)configure# primitive failoverip ocf:heartbeat:IPaddr params ip=172.31
.0.100 cidr netmask=16 nic=eth0 op monitor interval="10s"
crm(config)configure#
```

Descriptions des arguments :

- primitive, argument pour ajouter une primitive. Mais une primitive c'est quoi ? Un paramètre renseignant plusieurs valeurs indiquant au cluster quels scripts utiliser pour la ressource, où le trouver et à quel standard il correspond.
- failoverip est le nom de la primitive
- ocf, classe de la ressource
- heartbeat, fournisseur de la ressource
- IPaddr, RA (Resource Agent) gérant les adresses IPv4 virtuelles
- params, déclaration des paramètres
- ip=172.31.0.100, IP du cluster failover (vu des clients)
- op, les options
- monitor, action à effectuer, ici le monitoring de la ligne de vie
- interval=10s, on définit l'intervalle auquel on effectue l'action de monitoring.

On impose ensuite la contrainte de fonctionnement suivante :

- la ressource tourne sur le serveur 1 (dans ce document Ubuntu) dans le cas normal.

Il faut arrêter le service stonith dont le cluster n'a pas besoin. [STONITH](#) est une fonction de sécurité des données lors du basculement, il est garant de l'intégrité des données à chaque bascule. Ici nous n'en avons pas besoin, puisque nous avons installé le même site sur les deux machines et que ce site est statique. Quant au quorum, on n'en a pas besoin ici.

```
crm(config)configure# property stonith-enabled=false
crm(config)configure# property no-quorum-policy=ignore
crm(config)configure#
```

A ce stade on a une adresse de cluster mais on n'a pas encore imposé le nœud préféré. Et on voudrait dans le cas normal que la ressource tourne sur la machine Ubuntu, notre nœud primaire :

```
crm(config)configure# location preferred-Lubuntu failover_ip 100: Lubuntu
crm(config)configure#
```

Vérifiez et validez les commandes :

```
crm(config)configure# verify
crm(config)configure#
```

Vous pouvez maintenant vérifier le fichier généré en l'éditant :

```
crm(config)configure# edit
```

```
node $id="740229135" Lubuntu
node $id="740229136" Lubuntu1
primitive failoverip ocf:heartbeat:IPaddr \
    params ip="172.31.0.100" cidr_netmask="16" nic="eth0" \
    op monitor interval="10s"
location preferred-Lubuntu failoverip 100: Lubuntu
property $id="cib-bootstrap-options" \
    dc-version="1.1.10-42f2063" \
    cluster-infrastructure="corosync" \
    stonith-enabled="false" \
    no-quorum-policy="ignore"
```

Pour sortir, si « vi », faire ESC ":wq".

Enfin sortez de l'outil de configuration sans oublier d'appliquer les changements :

```
crm(config)configure# end
There are changes pending. Do you want to commit them? yes
crm(config)#
```

Sortez par « quit ». Vérifiez qu'apache est lancé (apachectl start sinon). Relancez les deux services corosync et pacemaker sur les deux machines.

```
root      9713   2028   0 15:17 ?        00:00:15 /usr/sbin/corosync
root      9726   2028   0 15:17 ?        00:00:00 pacemakerd
haclust+  9728   9726   0 15:17 ?        00:00:01 /usr/lib/pacemaker/cib
root      9729   9726   0 15:17 ?        00:00:01 /usr/lib/pacemaker/stonithd
haclust+  9730   9726   0 15:17 ?        00:00:00 /usr/lib/pacemaker/attrd
haclust+  9731   9726   0 15:17 ?        00:00:01 /usr/lib/pacemaker/crmd
delmp    10892   2028   0 15:22 ?        00:00:04 lxterminal
delmp    10893  10892   0 15:22 ?        00:00:00 gnome-pty-helper
delmp    10894  10892   0 15:22 pts/1    00:00:00 /bin/bash
root     10914  10894   0 15:22 pts/1    00:00:00 sudo -s
root     10931  10914   0 15:22 pts/1    00:00:00 /bin/bash
root     18986     2   0 16:02 ?        00:00:00 [kworker/u16:0]
root     20035     2   0 16:07 ?        00:00:00 [kworker/u16:2]
root     20866   2028   0 16:12 ?        00:00:00 /usr/sbin/apache2 -k start
www-data 20867  20866   0 16:12 ?        00:00:00 /usr/sbin/apache2 -k start
www-data 20868  20866   0 16:12 ?        00:00:00 /usr/sbin/apache2 -k start
root     20957  10931   0 16:12 pts/1    00:00:00 ps -edf
```

Et voilà ! Le cluster est activé et fonctionnel ! Remarquez que la ressource surveillée est bien sur le nœud choisi comme nœud préféré.

```
root@Lubuntu:~# crm_mon -l
Last updated: Wed Oct 21 22:52:52 2015
Last change: Wed Oct 21 22:52:04 2015 via cibadmin on Lubuntu
Stack: corosync
Current DC: Lubuntu (740229135) - partition with quorum
Version: 1.1.10-42f2063
2 Nodes configured
1 Resources configured

Online: [ Lubuntu Lubuntu1 ]

failoverip      (ocf::heartbeat:IPaddr):      Started Lubuntu
```

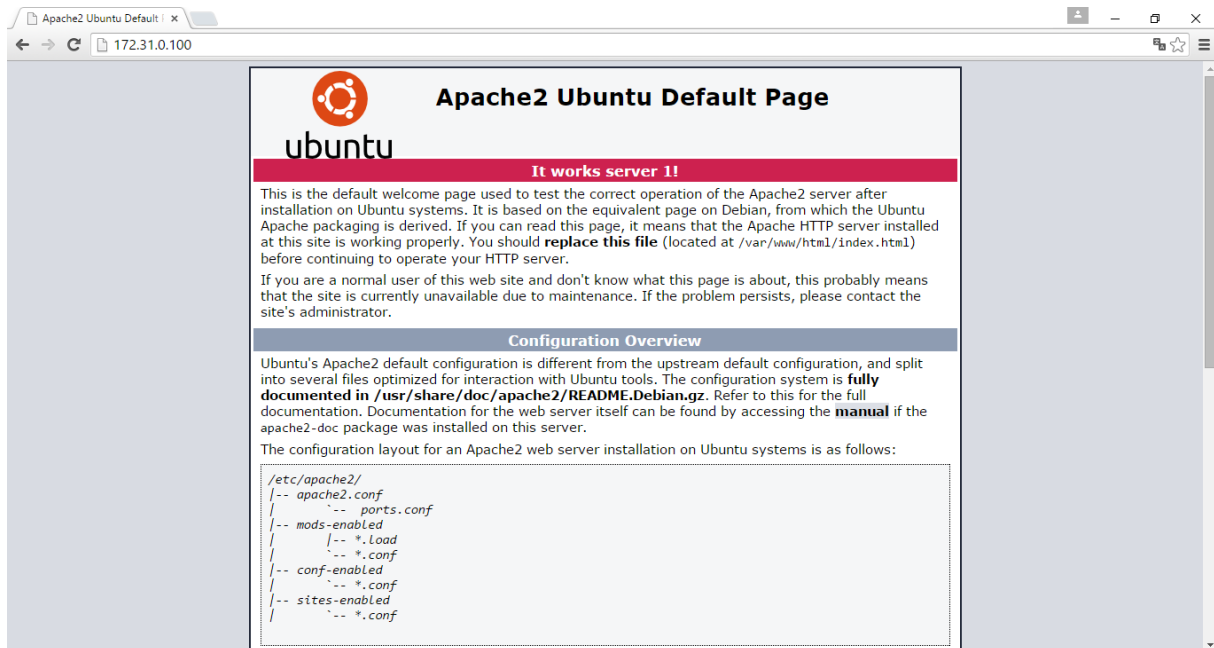
5

Marie-pascale Delamare

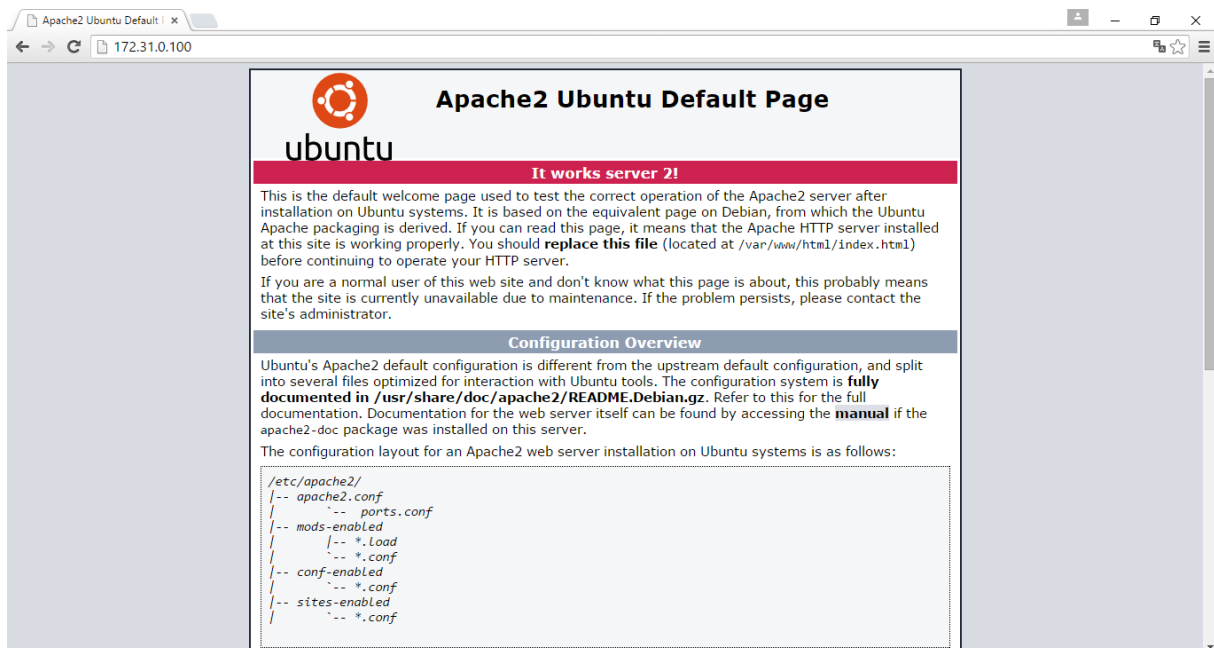
D'après <http://www.sebastien-han.fr/blog/2011/07/04/introduction-au-cluster-sous-linux> et <http://blogduyax.madyanne.fr/haute-disponibilite-avec-corosync-et-pacemaker.html> <http://www.reseaucerta.org/haute-disponibilite-C3%A9-dun-service-web-dynamique>

## VI TESTEZ LE CLUSTER

Depuis un client Seven, accédez à votre site web par l'adresse du cluster (ici 172.31.0.100) et notez le nom de la machine qui héberge actuellement le service.



Désactivez le réseau sur cette machine et vérifiez le basculement. Normalement les ressources devraient basculer sur le nœud restant.



Vous pouvez aussi vérifier cela avec la commande

`crm_mon --one-shot -V` ou `crm_mon -l`. La variable `Started` devrait changer.



```

root@Lubuntu1:~# crm_mon -l
Last updated: Wed Oct 21 22:55:46 2015
Last change: Wed Oct 21 22:52:04 2015 via cibadmin on Lubuntu
Stack: corosync
Current DC: Lubuntu1 (740229136) - partition WITHOUT quorum
Version: 1.1.10-42f2063
2 Nodes configured
1 Resources configured

Online: [ Lubuntu1 ]
OFFLINE: [ Lubuntu ]

failoverip (ocf::heartbeat:IPaddr): Started Lubuntu1

```

Relancez le réseau sur la machine préférée et vérifiez que la ressource est revenue sur le nœud préféré après avoir redémarré les services corosync et pacemaker.

```

root@Lubuntu:~# crm_mon -l
Last updated: Wed Oct 21 22:57:53 2015
Last change: Wed Oct 21 22:52:04 2015 via cibadmin on Lubuntu
Stack: corosync
Current DC: Lubuntu1 (740229136) - partition with quorum
Version: 1.1.10-42f2063
2 Nodes configured
1 Resources configured

Online: [ Lubuntu Lubuntu1 ]

failoverip (ocf::heartbeat:IPaddr): Started Lubuntu

```

## VII BASCULER SI LE SERVICE APACHE EST DÉFICIENT SUR LE NŒUD COURANT

Là je vous laisse faire, voir la seconde URL donnée en bas de page et l'annexe située en fin de document.

## VIII POUR ALLER PLUS LOIN

### VIII.1 AJOUTER UN NŒUD AU CLUSTER

Rien de plus simple, il suffit simplement d'appliquer le même paramétrage qu'aux deux nœuds précédents. En résumé :

- Installer pacemaker.
- Via SCP on copie le fichier /etc/corosync/corosync.conf sur le nouveau nœud.
- Toujours via SCP on copie la "authkey" sur le nouveau nœud.
- Lancer le démon corosync et le démon pacemaker.

ou

- cloner une de vos machines.
- changer les paramètres qui s'imposent.

## VIII.2 CHANGER L'IP VIRTUELLE DU CLUSTER

Petite astuce pour changer l'IP du cluster :

```
1 root@Lubuntu:~# crm
2 configure
3 edit
4 commit
5 show
6 verify
7 end
8 quit
```

Après avoir rentré la commande « edit » vous pourrez éditer la configuration du fichier du cluster, chercher la ligne correspondante à l'adresse IP virtuelle du cluster et renseigner la nouvelle IP. Après avoir appliqué via « commit », faites un petit test de ping pour voir que tout fonctionne et que la ressource est maintenant accessible sur la nouvelle adresse.

## VIII.3 METTRE DES NŒUDS EN STANDBY

Pour des raisons de maintenances, mise à jour ou autre on peut avoir besoin de mettre en standby un nœud du cluster, pour cela :

```
1 root@Lubuntu:~# crm
2 crm(live)# node
3 crm(live)node# standby <votre_noeud>
4 crm(live)node# quit
5 bye
```

Vous devriez observer le basculement de la ressource sur l'autre nœud du cluster. Pour remettre celui-ci en ligne :

```
1 root@Lubuntu:~# crm
2 crm(live)# node
3 crm(live)node# online <votre_noeud>
4 crm(live)node# bye
5 bye
```

## VIII.4 ARRÊTER ET SUPPRIMER UNE RESSOURCE DU CLUSTER

Vous vous êtes trompé et voulez recommencer ?

```
1 root@Lubuntu:~# crm resource stop ma_resource
2 root@Lubuntu:~# crm configure delete ma_resource
```

## VIII.5 VOIR SUR QUEL NŒUD FONCTIONNE LA RESSOURCE

Rentrez la commande suivante :

```
1 root@Lubuntu:~# crm_resource -r failoverip -W
2 resource failoverip is running on: Lubuntu
```

# Annexe : Configuration de la ressource « serviceWeb »

Pour configurer cette ressource :

- l'utilisation de la touche tabulation qui peut aider à trouver la « bonne » commande ou le « bon » argument (la double tabulation listera les possibilités) ;



- l'utilisation de la commande `help` qui liste les commandes possibles ;
- la possibilité de configurer plusieurs ressources avec leurs propriétés avant de valider le tout.

```
root@Lubuntu:~# crm configure
crm(live)configure# primitive serviceWeb ocf:heartbeat:apache params
configfile="/etc/apache2/apache2.conf" op monitor interval="60s" op start timeout="40s" op stop
timeout="40s"
crm(live)configure# commit
crm(live)configure# quit
```

**On crée une ressource nommée « serviceWeb » avec comme paramètre le chemin du fichier de configuration d'Apache2.**

Remarque : « configfile » a, pour paramètre par défaut `/etc/apache2/http.conf` (on peut le constater avec la commande : `crm ra info ocf:heartbeat:apache`), ce qui n'est pas valable dans notre configuration => il faut donc lui préciser ce paramètre. Les autres paramètres par défaut sont corrects.

**Chaque option (en règle générale, chaque action à effectuer) est ensuite précédée du mot clé « op ».**

On crée un moniteur pour cette ressource avec un test de la « vie » de la ressource toutes les 60 secondes (`op monitor interval="60s"`) et on spécifie en plus un timeout de démarrage de 40 secondes (`"op start timeout="40s"`) et un timeout d'arrêt de 40 secondes (`"op stop timeout="40s"`).

On valide la ressource par « *commit* » et on sort de la commande interactive par « *quit* » ou « *exit* »

*N'hésitez pas à utiliser la tabulation et double tabulation...*

**Remarque : avant de valider la ressource, on peut la consulter :**

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

**À la validation, un « Warning » peut apparaître :**

***WARNING: serviceWeb: specified timeout 40s for stop is smaller than the advised 60s***

Vous pouvez l'ignorer ou... en tenir compte. Les développeurs ont estimé qu'un délai de 60s pour arrêter ce service était raisonnable. Si vous pensez qu'effectivement, l'arrêt du service Web prendra plus de 40 secondes, il vaut mieux augmenter le délai.

**Sur la console `crm_mon`**, il apparaît bien la nouvelle ressource, vous pouvez le vérifier par la commande « `crm_mon -1` »:

Mais, comme on peut le voir, la ressource *serviceWeb* n'est pas démarrée sur le même nœud que celle concernant l'IP virtuelle. **Par défaut Pacemaker répartit les ressources entre les membres du Cluster.**

```

root@Lubuntu:~# crm_mon -l
Last updated: Wed Oct 21 23:04:26 2015
Last change: Wed Oct 21 23:03:05 2015 via cibadmin on Lubuntu
Stack: corosync
Current DC: Lubuntu (740229135) - partition with quorum
Version: 1.1.10-42f2063
2 Nodes configured
2 Resources configured

Online: [ Lubuntu Lubuntu1 ]

failoverip      (ocf::heartbeat:IPaddr):      Started Lubuntu
serviceweb      (ocf::heartbeat:apache):      Started Lubuntu1

```

Pour que l'adresse IP virtuelle et le service « serviceWeb » soient sur le même nœud, il existe plusieurs possibilités comme la colocation :

```

root@Lubuntu:~# crm configure
crm(live)configure# colocation failoverip-serviceWeb inf: failoverip serviceweb
crm(live)configure# commit
crm(live)configure# quit

```

On crée une colocation nommée « failoverip-serviceweb » composé des ressources « failoverip » et « serviceWeb » qui seront toujours démarrées sur le même nœud.

Attention c'est le cluster qui gère désormais le démarrage d'apache2, il faut donc le retirer du démarrage automatique de chaque machine par la commande :

```
update-rc.d -f apache2 remove
```

```

root@Lubuntu:~# crm_mon -l
Last updated: Wed Oct 21 23:09:29 2015
Last change: Wed Oct 21 23:07:13 2015 via cibadmin on Lubuntu
Stack: corosync
Current DC: Lubuntu (740229135) - partition with quorum
Version: 1.1.10-42f2063
2 Nodes configured
2 Resources configured

Online: [ Lubuntu Lubuntu1 ]

failoverip      (ocf::heartbeat:IPaddr):      Started Lubuntu
serviceweb      (ocf::heartbeat:apache):      Started Lubuntu

```

## Test de la solution

Plusieurs possibilités pour tester la solution mise en place dont :

- arrêtez le serveur ou le service (utile pour tester la migration des autres ressources) ;
- mettre un nœud en maintenance : **crm node standby** sur le nœud que l'on veut rendre inactif (**crm node online** pour le remettre actif).

Si le service est arrêté, pacemaker a été configuré pour essayer de le redémarrer dans un premier temps : c'est le premier point à tester.

Ensuite, pour tester un arrêt définitif du service de manière à ce que l'ensemble des ressources migrent sur l'autre nœud, il faut s'arranger pour que le service ne puisse pas redémarrer...

Quand le Cluster bascule sur l'autre nœud suite à un certain nombre d'erreurs sur une ressource, le nœud qui a rencontré les erreurs ne pourra plus héberger les ressources tant que toutes les erreurs n'auront pas été effacées par la commande suivante : ***crm resource cleanup <id\_resource>***