



Super quick steps to set up a Pacemaker cluster to Centos 6

20.10.23

Pegasi Knowledge

<https://ghost.pegasi.fi/wiki/>

Table of Contents

<i>Super quick steps to set up a Pacemaker cluster to Centos 6</i>	3
Install software	3
Prepare virtual machine fencing	3
configure your pacemaker	4
Fire it up	4
Start putting the thing to work	5

Super quick steps to set up a Pacemaker cluster to Centos 6

This is a quickie to show you how to start do Pacemaker ensured high available Apache with a service IP and KVM virtual host fencing.

Install software

Install these to every node and virtual host as well

```
yum install fence-virt fence-virttd fence-virttd-multicast fence-virttd-libvirt
```

Install these to pacemaker nodes / guests

```
yum install pacemaker cman pcs ccs resource-agents
```

Prepare virtual machine fencing

Configure at virtual host, use virbr0, /etc/cluster/fence_xvm.key

```
fence_virttd -c
```

Make key

```
dd if=/dev/random bs=512 count=1 of=/etc/cluster/fence_xvm.key
```

Configuration should be like this

```
backends {
    libvirt {
        uri = "qemu:///system";
    }
}
listeners {
    multicast {
        key_file = "/etc/cluster/fence_xvm.key";
        interface = "virbr0";
        port = "1229";
        address = "225.0.0.12";
        family = "ipv4";
    }
}
fence_virttd {
```

```
backend = "libvirt";  
listener = "multicast";  
module_path = "/usr/lib64/fence-virt";  
}
```

Start fencing daemon, insert to your rc.local to get it up after boot

```
fence_virt
```

Test it, you should see your virtual guests

```
fence_xvm -o list
```

Copy the /etc/cluster/fence_xvm.key to all guests

Test in GUESTS. if timeouts check if you are using other interface at the virtual host, br0 etc

```
fence_xvm -o listen
```

Configure the fencing to your pacemaker cluster later with command

```
crm configure primitive st-virt stonith:fence_xvm
```

configure your pacemaker

Do this in one node and copy the /etc/cluster to all nodes

```
ccs -f /etc/cluster/cluster.conf --addfencedev pcmk agent=fence_pcmk  
ccs -f /etc/cluster/cluster.conf --addmethod pcmk-redirect host1  
ccs -f /etc/cluster/cluster.conf --addmethod pcmk-redirect host2  
ccs -f /etc/cluster/cluster.conf --addfenceinst pcmk host1 pcmk-redirect  
port=host1  
ccs -f /etc/cluster/cluster.conf --addfenceinst pcmk host2 pcmk-redirect  
port=host2
```

Do this in every node

```
cp /etc/corosync/corosync.conf.example /etc/corosync/corosync.conf  
echo "CMAN_QUORUM_TIMEOUT=0">> /etc/sysconfig/cman
```

Fire it up

```
chkconfig cman on
```

```
chkconfig pacemaker on
```

```
service cman start  
service pacemaker start
```

```
pcs status  
pcs config
```

Start putting the thing to work

Set basic properties

```
pcs property set stonith-enabled=true  
pcs property set no-quorum-policy=ignore  
pcs resource defaults migration-threshold=1
```

Set up Apache with HA IP

Be sure that your Apache provides <http://localhost/server-status>

```
pcs resource create WebSite ocf:heartbeat:apache  
configfile=/etc/httpd/conf/httpd.conf  
statusurl="http://localhost/server-status" op monitor interval=1min  
pcs resource create ClusterIP ocf:heartbeat:IPaddr2 ip=1.2.3.4 nic="eth0"  
cidr_netmask=27 op monitor interval=30s  
pcs constraint colocation add WebSite ClusterIP INFINITY  
pcs constraint location WebSite prefers host1=50
```

Finally put the nodes to shoot each other when they've get a problem

```
crm configure primitive st-virt stonith:fence_xvm
```