Cluster Management

1. Creating a cluster

1.1. Install "pcs" and "fence-agents-all" on each cluster node:

```
# yum install -y pcs fence-agents-all
```

1.2. Enable and start the "pcsd" service on each cluster node:

```
# systemctl enable pcsd
# systemctl start pcsd
```

1.3. If needed, enable cluster communication through the firewall on each cluster node:

```
# firewall-cmd --permanent --add-service=high-availability
# firewall-cmd --reload
```

1.4. Change the password of the "hacluster" user to "H4clust3r" on each cluster node:

```
# echo "H4clust3r" | passwd --stdin hacluster
```

1.5. Authenticate all cluster nodes on any cluster node:

```
[root@node1 ~]# pcs cluster auth node1 node2 node3
Username: hacluster
Password: H4clust3r
node1: Authorized
node2: Authorized
node3: Authorized
```

1.6. Configure and start a three-node cluster named "cluster01" on any cluster node:

```
[root@node1 ~]# pcs cluster setup --start --name cluster01 node1 node2 node3
```

1.7. Enable automatic startup of the cluster on all configured cluster nodes on any cluster node:

```
[root@node1 ~]# pcs cluster enable --all
```

1.8. Verify that the cluster is running and all configured cluster nodes have joined the cluster on any cluster node:

```
[root@node1 ~]# pcs status
Cluster name: cluster01
WARNING: no stonith devices and stonith-enabled is not false
Last updated: Mon Sep 15 05:41:18 2014
Last change: Mon Sep 15 05:41:03 2014 via crmd on node1
Stack: corosync
Current DC: node1 (1) - partition with quorum
Version: 1.1.10-29.el7-368c726
3 Nodes configured
0 Resources configured

Online: [ node1 node2 node3 ]

Full list of resources:

PCSD Status:
node1: Online
node2: Online
node3: Online

Daemon Status:
corosync: active/enabled
pacemaker: active/enabled
pcsd: active/enabled
```

1.9. Add the fence devices for the virtual machines "node1", "node2" and "node3" to the cluster on any cluster node:

```
[root@node1 ~]# pcs stonith create fence_node1 fence_ipmilan
pcmk_host_list="node1" ipaddr="172.24.6.1" login="node1" passwd="N0d31"
lanplus=1 cipher=3
[root@node1 ~]# pcs stonith create fence_node2 fence_ipmilan
pcmk_host_list="node2" ipaddr="172.24.6.1" login="node2" passwd="N0d32"
lanplus=1 cipher=3
[root@node1 ~]# pcs stonith create fence_node3 fence_ipmilan
pcmk_host_list="node3" ipaddr="172.24.6.1" login="node3" passwd="N0d33"
lanplus=1 cipher=3
```

1.10. Verify that the fence devices have been added correctly to the cluster on any cluster node:

```
[root@node1 ~]# pcs stonith show
fence_node1 (stonith:fence_ipmilan): Started
fence_node2 (stonith:fence_ipmilan): Started
fence_node3 (stonith:fence_ipmilan): Started
```

2. Adding a node to the cluster

2.1. Install "pcs" and "fence-agents-all" on the new cluster node:

```
[root@node4 ~]# yum install -y pcs fence-agents-all
```

2.2. Enable and start the "pcsd" service on the new cluster node:

```
[root@node4 ~]# systemctl enable pcsd; systemctl start pcsd
```

2.3. If needed, allow cluster communications to pass through the firewall on the new cluster node:

```
[root@node4 ~]# firewall-cmd --permanent --add-service=high-availability;
firewall-cmd --reload
```

2.4. Change the password of the "hacluster" user to "H4clust3r" on the new cluster node:

```
[root@node4 ~]# echo "H4clust3r" | passwd --stdin hacluster
```

2.5. Authenticate the new node on the existing cluster on any cluster node:

```
[root@node1 ~]# pcs cluster auth -u hacluster -p H4clust3r node4
```

2.6. Add the new node to the cluster on any cluster node:

```
[root@node1 ~]# pcs cluster node add node4
nodea1: Corosync updated
node2: Corosync updated
node3: Corosync updated
node4: Succeeded
```

2.7. Authenticate "node4" with the other nodes on the new cluster node:

```
[root@node4 ~]# pcs cluster auth -u hacluster -p H4clust3r
node1: Authorized
node2: Authorized
node3: Authorized
node4: Already authorized
```

2.8. Add a fence device for the new node to the cluster on any cluster node:

```
[root@node1 ~]# pcs stonith create fence_node4 fence_ipmilan
pcmk_host_list="node4" ipaddr="172.24.6.1" login="node4" passwd="N0d34"
```

```
lanplus=1 cipher=3
```

2.9. Verify that the fence device has been added correctly to the cluster on any cluster node:

```
[root@node1 ~]# pcs stonith show
fence_node1 (stonith:fence_ipmilan): Started
fence_node2 (stonith:fence_ipmilan): Started
fence_node3 (stonith:fence_ipmilan): Started
fence_node4 (stonith:fence_ipmilan): Started
```

2.10. Enable automatic startup of the cluster on the new cluster node:

```
[root@node4 ~]# pcs cluster enable
```

2.11. Start the cluster on the new cluster node:

```
[root@node4 ~]# pcs cluster start
Starting Cluster...
```

2.12. Verify that the cluster is running and all configured cluster nodes have joined the cluster on any cluster node:

```
[root@node1 ~]# pcs status
Cluster name: cluster01
Last updated: Tue Sep 24 05:41:18 2014
Last change: Tue Sep 24 05:41:03 2014 via crmd on node1
Stack: corosync
Current DC: node1 (1) - partition with quorum
Version: 1.1.10-29.el7-368c726
4 Nodes configured
4 Resources configured

Online: [ node1 node2 node3 node4 ]

Full list of resources:

fence_node1 (stonith:fence_ipmilan): Started node1
fence_node2 (stonith:fence_ipmilan): Started node2
fence_node3 (stonith:fence_ipmilan): Started node3
fence_node4 (stonith:fence_ipmilan): Started node4

PCSD Status:
node1: Online
node2: Online
node3: Online
node4: Online
```

```
Daemon Status:
corosync: active/enabled
pacemaker: active/enabled
pcsd: active/enabled
```

---

3. Removing a node from the cluster

3.1. Remove the cluster node from the cluster on any cluster node:

```
[root@node1 ~]# pcs cluster node remove node4
```

3.2. Remove the fence device from the cluster on any cluster node:

```
[root@node1 ~]# pcs stonith delete fence_node4
```

---

4. Configuring cluster quorum options

4.1. Stop the cluster on all nodes on any cluster node:

```
[root@node1 ~]# pcs cluster stop --all
```

4.2. Add the "auto_tie_breaker" option in the "quorum" section of /etc/corosync/corosync.conf on any cluster node:

```
[root@node1 ~]# vi /etc/corosync/corosync.conf
quorum {
provider: corosync_votequorum
auto_tie_breaker: 1
auto_tie_breaker_node: lowest
}
```

4.3. Synchronize /etc/corosync/corosync.conf from the current node to all other nodes in the cluster:

```
[root@node1 ~]# pcs cluster sync
```

4.4. Start the cluster on all nodes on any cluster node:

```
[root@node1 ~]# pcs cluster start --all
```

---

5. Creating iSCSI clients (initiators)

5.1. Install "iscsi-initiator-utils" on each cluster node:

```
# yum install -y iscsi-initiator-utils
```

5.2. Create a unique IQN for the iSCSI initiator by modifying the "InitiatorName" setting in /etc/iscsi/initiatorname.iscsi on each cluster node:

```
# vi /etc/iscsi/initiatorname.iscsi
InitiatorName=iqn.2020-02.com.domain:nodeX
```

5.3. Enable and start the "iscsi" service on each cluster node:

```
# systemctl enable iscsi; systemctl start iscsi
```

5.4. Discover the configured iSCSI target(s) provided by the iSCSI target server portal(s) on each cluster node:

```
# iscsiadm -m discovery -t st -p 192.168.1.2
192.168.1.2:3390,1 iqn.2020-02.com.domain:cluster
# iscsiadm -m discovery -t st -p 192.168.1.3
192.168.1.3:3390,1 iqn.2020-02.com.domain:cluster
```

5.5. Log into the presented iSCSI targets on each cluster node:

```
# iscsiadm -m node -T iqn.2020-02.com.domain:cluster -p 192.168.1.2 -l
# iscsiadm -m node -T iqn.2020-02.com.domain:cluster -p 192.168.1.3 -l
```

5.6. Verify that the new block devices created by the iSCSI target logins are available on each cluster node:

```
# lsblk
NAME   MAJ:MIN RM SIZE RO TYPE MOUNTPOINT
sda      8:0    0  20G  0 disk
sdb      8:16   0  20G  0 disk
vda    253:0    0  10G  0 disk
└─vda1 253:1    0  10G  0 part /
```

5.7. Optionally display detailed information about the available iSCSI targets on any cluster node:

```
# iscsiadm -m session -P 3
```

6. Configuring multipathing

6.1. Install "device-mapper-multipath" on each cluster node:

```
# yum install -y device-mapper-multipath
```

6.2. Enable multipath configuration on each cluster node:

```
# mpathconf --enable
```

6.3. Edit the "blacklist" section in /etc/multipath.conf to ignore the local disks on each cluster node:

```
# vi /etc/multipath.conf
blacklist {
devnode "^vd[a-z]"
}
```

6.4. Enable and start the "multipathd" service on each cluster node:

```
# systemctl enable multipathd; systemctl start multipathd
```

6.5. Verify that the multipath device "mpatha" is available on each cluster node:

```
# multipath -ll
mpatha (360014053bd9ea2a35914e39a556051cf) dm-0 LIO-ORG ,clusterstor
size=20.0G features='0' hwhandler='0' wp=rw
|-+- policy='service-time 0' prio=1 status=active
| `- 2:0:0:0 sda 8:0  active ready running
`-+- policy='service-time 0' prio=1 status=enabled
  `- 3:0:0:0 sdb 8:16 active ready running
```

7. Creating a clustered logical volume

7.1. Install "dlm" and "lvm2-cluster" on each cluster node:

```
# yum install -y dlm lvm2-cluster
```

7.2. Set the LVM "locking_type" to "3" (enable cluster locking) on each cluster node:

```
# lvmconf --enable-cluster
```

7.3. Stop the lvmetad service to match the modified LVM configuration on each cluster node:

```
# systemctl stop lvm2-lvmetad
```

7.4. Set the global Pacemaker parameter "no_quorum_policy" to "freeze" on any cluster node:

```
[root@node1 ~]# pcs property set no-quorum-policy=freeze
```

7.5. Create a cloned "dlm" resource (required dependency for clvmd and GFS2) using the "controld" resource agent on any cluster node:

```
[root@node1 ~]# pcs resource create dlm controld op monitor interval=30s on-
fail=fence clone interleave=true ordered=true
```

7.6. Create a cloned "clvmd" resource using the "clvm" resource agent on any cluster node:

```
[root@node1 ~]# pcs resource create clvmd clvm op monitor interval=30s on-
fail=fence clone interleave=true ordered=true
```

7.7. Create resource constraints to control "dlm" and "clvmd" start up order and enforce them to run on the same node on any cluster node:

```
[root@node1 ~]# pcs constraint order start dlm-clone then clvmd-clone
[root@node1 ~]# pcs constraint colocation add clvmd-clone with dlm-clone
```

7.8. Create an LVM physical volume "/dev/mapper/mpatha" on any cluster node:

```
[root@node1 ~]# pvcreate /dev/mapper/mpatha
```

7.9. Create a clustered volume group "clustervg" on any cluster node:

```
[root@node1 ~]# vgcreate -Ay -cy clustervg /dev/mapper/mpatha
```

7.10. Create a 10 GB logical volume "clusterlv" on any cluster node:

```
[root@node1 ~]# lvcreate -L 10G -n clusterlv clustervg
```

---

8. Creating a GFS2 cluster file system

8.1. Install "gfs2-utils" on each cluster node:

```
# yum install -y gfs2-utils
```

8.2. Format the logical volume with a GFS2 file system with 3 journals on any cluster node:

```
[root@node1 ~]# mkfs.gfs2 -j3 -p lock_dlm -t cluster01:web
/dev/clustervg/clusterlv
```

---

9. Creating cluster resources

9.1. List all available resource agents (scripts managing cluster resources) on any cluster node:

```
[root@node1 ~]# pcs resource list
ocf:heartbeat:CTDB - CTDB Resource Agent
ocf:heartbeat:Delay - Waits for a defined timespan
ocf:heartbeat:Dummy - Example stateless resource agent
ocf:heartbeat:Filesystem - Manages filesystem mounts
...
```

9.2. Display a particular resource agent parameters on any cluster node:

```
[root@node1 ~]# pcs resource describe Filesystem
ocf:heartbeat:Filesystem - Manages filesystem mounts
...
Resource options:
device (required): The name of block device for the filesystem, or -U, -L
options for mount, or NFS mount specification.
directory (required): The mount point for the filesystem.
fstype (required): The type of filesystem to be mounted.
options: Any extra options to be given as -o options to mount. For bind
mounts,
add "bind" here and set fstype to "none". We will do the right thing for
options
such as "bind,ro".
...
```

9.3. Create a "clusterfs" cloned resource using the "Filesystem" resource agent to manage the GFS2 file system and automatically mount it on all three nodes on cluster startup on any cluster node:

```
[root@node1 ~]# pcs resource create clusterfs Filesystem
device="/dev/clustervg/clusterlv" directory="/var/www" fstype="gfs2"
options="noatime" op monitor interval=10s on-fail=fence clone
interleave=true
```

9.4. Create resource constraints to control "clvmd" and "clusterfs" start up order and enforce them to run on the same node on any cluster node:

```
[root@node1 ~]# pcs constraint order start clvmd-clone then clusterfs-clone
```

```
[root@node1 ~]# pcs constraint colocation add clusterfs-clone with clvmd-
clone
```

9.5. Create a "webip" resource using the "IPaddr2" resource agent for the IP address 172.15.1.10/24 as part of the "web_rg" resource group on any cluster node:

```
[root@node1 ~]# pcs resource create webip IPaddr2 ip=172.15.1.10
cidr_netmask=24 --group web_rg
```

9.6. Create a "webserver" resource using the "apache" resource agent with default settings as part of the "web_rg" resource group on any cluster node:

```
[root@node1 ~]# pcs resource create webserver apache --group web_rg
```

9.7. Configure the "web_rg" resource group to start after the "clusterfs" resource on any cluster node:

```
[root@node1 ~]# pcs constraint order start clusterfs-clone then web_rg
```

9.8. Configure the "web_rg" resource group to preferably start on "node1" on any cluster node:

```
[root@node1 ~]# pcs constraint location web_rg prefers node1
```

9.9. Display all configured cluster resource constraints on any cluster node:

```
[root@node1 ~]# pcs constraint show
```

9.10. Display all configured cluster resources on any cluster node:

```
[root@node1 ~]# pcs resource show
```

9.11. Display details about a configured cluster resource on any cluster node:

```
[root@node1 ~]# pcs resource show webserver
```

---

10. Removing cluster resources

10.1. Remove the "webserver" resource from the "web_rg" resource group on any cluster node:

```
[root@node1 ~]# pcs resource group remove web_rg webserver
```

10.2. Remove the "web_rg" resource group including all resources that it contains on any cluster

node:

```
[root@node1 ~]# pcs resource delete web_rg
```

From:
https://prompt.cz/en/ - **Prompt.cz**

Permanent link:
https://prompt.cz/en/cluster-management

Last update: **2021/08/05 11:26**